Master's thesis

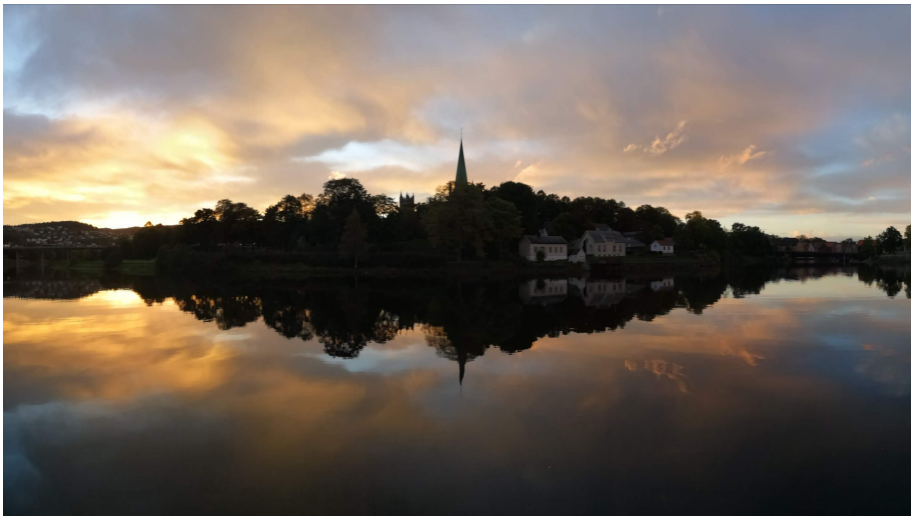
NTNU
Norwegian University of Science and Technology
Faculty of Information Technology and Electrical Engineering
Department of Mathematical Sciences

Stephanie Jebsen Fagerås

# Forecasting Red Wine Rankings at Vinmonopolet with Machine Learning

## Forecasting Red Wine Rankings

Master's thesis in Applied Physics and Mathematics
Supervisor: Erlend Aune
July 2021



Nidelva in September, personal photo

NTNU
Norwegian University of
Science and Technology

Stephanie Jebsen Fagerås

# Forecasting Red Wine Rankings at Vinmonopolet with Machine Learning

Forecasting Red Wine Rankings

Master's thesis in Applied Physics and Mathematics
Supervisor: Erlend Aune
July 2021

Norwegian University of Science and Technology
Faculty of Information Technology and Electrical Engineering
Department of Mathematical Sciences

NTNU
Norwegian University of
Science and Technology

# Abstract

In Norway, beverage sales with over 4.75 % alcohol are monopolized by Vinmonopolet and controlled by strict laws prohibiting advertisement. Vinmonopolet changes its product line every two months, launching new products and removing the least popular products from *basisutvalget*, the small selection not only available through orders but also available in stores. The products are imported by importers, whose aim for some of the products is to get them into *basisutvalget*, guaranteeing higher sales numbers. Which products that manage to claim a spot in *basisutvalget* is decided through a ranking system based on sales numbers.

In this thesis, we analyze red wine sales and attempt to forecast the ranking lists to evaluate which products risk leaving *basisutvalget* and which products might sell well enough to enter *basisutvalget*. The ranking lists are mapped from one-, two-, and three-month forecasts using Long Short-Term Memory (LSTM), Seasonal Autoregressive Integrated Moving Average (SARIMA), and persistence forecasting on sales numbers. Additional features are tested on the LSTM and SARIMA models, and various combinations of price groups are used to train the LSTM model.

None of the attempted features improved the models significantly, but training the LSTM model on all price groups improved the Mean Absolute Error (MAE) by 25 %. The final models produced an average MAE of 158, 205, and 291 for a one-month LSTM, SARIMA, and persistence forecast consecutively. The MAE increased with 105 %, 150 %, and 158 % for the same models for three-month forecasts.

Attempting to identify products whose rank shifts over or below the ranking limit, we find that the results are poor and fluctuate; these events occur too seldom to function as an accurate performance measure. The stability of these ranks imply that the most important factors influencing entry and exit of *basisutvalget* are the number of new products launched directly into *basisutvalget* and products shifting price range.

We discuss alternative methods to better utilize the forecasts for knowledge gain. Ranking the forecasts resulted in unnecessary information loss, and the performance measures we chose all had different weaknesses. We also discuss which features outside the data set that are expected to increase performance and some factors that might limit the obtainable performance.

# Sammendrag

I Norge blir salg av drikkevarer med over 4,75 % alkohol monopolisert av Vinmonopolet og kontrollert av strenge lover som forbyr reklame. Vinmonopolet bytter produktutvalg annenhver måned, der de lanserer nye produkter og fjerner de minst populære produktene fra basisutvalget, utvalget som er garantert en plass i butikkene. Disse produktene og noen av de nye ender opp i bestillingsutvalget. Produktene importeres av importører som for noen av produktene har som mål å få disse i basisutvalget, noe som garanterer høyere salgstall. Hvilke produkter som får en plass i basisutvalget avgjøres gjennom et rangeringssystem basert på salgstall.

I denne oppgaven analyserer vi salg av rødviner og prøver å lage en prognose av rangeringslistene for å evaluere hvilke produkter som risikerer å forlate basisutvalget og hvilke produkter som kan selge godt nok til å gå inn i basisutvalget. Rangeringslistene er laget av en-, to- og tremånedersprognoser ved bruk av Long Short-Term Memory (LSTM), Seasonal Autoregressive Integrated Moving Average (SARIMA) og persistence forecast på salgstall. Ytterligere kovariater er testet på LSTM- og SARIMA-modellene, og forskjellige kombinasjoner av prisgruppene ble brukt til å trene LSTM-modellen.

Ingen kovariater forbedret modellene betydelig, men å trene LSTM-modellen på alle prisgrupper forbedrer gjennomsnittlig absolutt avvik (MAE) med 25 %. De endelige modellene produserte en gjennomsnittlig MAE på 158, 205 og 291 for én måneds prognose med henholdsvis LSTM, SARIMA og persistence forecast. MAE økte med 105 %, 150 % og 158 % for de samme modellene for tremånedersprognoser.

Ved forsøk på å identifisere produkter med en rangering som flytter seg over eller under styringstallet, finner vi at resultatene er dårlige og svinger mye; disse hendelsene forekommer for sjelden til å kunne brukes til å evaluere modellen. Stabiliteten i disse rangeringene tyder på at de viktigste faktorene som påvirker inngang og utgang av basisutvalget er antallet nye produkter som lanseres direkte i basisutvalget og produktene som skifter prisklasse.

Vi diskuterer alternative metoder for å bedre kunne utnytte resultatene fra prognosene. Rangeringen av disse resulterte i unødvendig informasjonstap og evalueringsmetodene vi brukte hadde alle sine svakheter. Vi diskuterer også hvilke kovariater utenfor datasettet som forventer å kunne øke ytelsen til modellen og noen elementer som kan begrense mulig ytelse.

# Preface

This thesis marks the end of my five-year master's degree in Industrial Mathematics within the Applied Physics and Mathematics M.Sc. program at the Norwegian University of Science and Technology (NTNU). I was engaged in writing this thesis from March to July 2021 at the Department of Mathematical Sciences. The thesis is not a direct follow-up of my specialization project, "Classifying Trends in Wine Sales using LSTM Multi-class Classification on Multivariate Time Series," for the fall 2020 semester. Still, the experience with the data and data handling was invaluable upon starting this work.

I want to thank my supervisor, Erlend Aune, for all his guidance and support this last year. Our weekly brainstorming was a great motivational boost and left me with so many ideas for this project that I feel we've barely scratched the surface of the possibilities presented by this data set. I would also like to thank Grapespot for collecting the data and sharing it with me. Last of all, I would like to thank my family, friends, and everyone I've been in contact with this last year for all the support and inspiration you have come with. I was lucky to have a thesis topic that most people can relate to and are excited to talk about.

For those of you who are not only interested in the data science behind this project, but also interested in wine, I hope you won't be too disappointed to hear that reading this thesis won't turn you into a wine connoisseur. That said, you are welcome to go to Vinmonopolet's site to look up the article numbers I've used in my examples and see if your palate agrees with the average Norwegian's.

Stephanie Jebsen Fagerås
July 21, 2021
Bergen, Norway

# Contents

# Chapter 1

# Introduction

## 1.1 Wine Sales

In 2018, Norwegians over 15 years of age bought an average of 6.77 liters of pure alcohol [1]. In comparison, the Swedish bought 8.83 [2], and the Danish bought 9.29 liters of pure alcohol[3]. Despite the slightly lower sales numbers in Norway, this still amounts to large quantities of alcoholic beverages. A total of 2 677 008 000 liters of beer, 88 029 000 liters of wine, and 15 783 000 liters of liquor were sold that year, counting registered and unregistered sales. Registered sales are sales through Vinmonopolet, restaurants, bars, and shops, while unregistered sales are through duty-free shops and importation from other countries. While 93.6 % of beer sales were made through local stores, restaurants, or bars, 76.1 % of wine sales and 71.1 % of liquor sales were made through Vinmonopolet [1].

Vinmonopolet is a Norwegian state-owned retailer that sells alcoholic beverages with an alcoholic percentage above 4.75 %, which registered importers import. The company was formed in 1922, a period after the first world war when liquor was banned, and certain parties were trying to ban wine and beers with over 2.5 % alcohol. This ban forced the Norwegian alcohol consumption to an all-time registered low in 1918, 0.61 liters of pure alcohol per adult [4]. Vinmonopolet could guarantee equal access to alcohol countrywide, and the promise of steady importation of alcohol helped facilitate new trade treaties between Norway and countries exporting alcohol, especially France [5]. Though the ban was removed in 1927 and political changes have been made in the following years, Vinmonopolet has remained the primary provider of stronger alcoholic beverages in Norway since then.

Vinmonopolet sells only four different red wines produced in Norway per 2021; most wines are imported from countries with warmer climates more suitable for growing grapes. Italy, Spain, and France alone produced approximately 74 % of the red wines imported and sold at Vinmonopolet in 2018. These three countries are also the largest wine exporters; they alone exported 6.12 billion liters of wine in 2018, 56.7 % of the total amounts of wine exported in the world [6]. Chile, Australia, Argentina, the USA, South Africa, and New Zealand were among the top 10 selling countries, showing how popular red wines are produced on multiple continents. The relatively small selection of wines sold on the Norwegian market is decided by consumers' demand, fashions Vinmonopolet wishes to explore, and what deals importers make with wineries. The primary factor in deciding which wines *remain* in the selection is the sales numbers.

## 1.2 Machine Learning

Machine learning is a popular tool in present-day technological developments. Not in the typical cinematic sense, where artificial intelligence takes over the world, leaving humanity fighting technology with analog weapons, but rather more discretely, analyzing large amounts of data quicker and in many cases more accurately than humans are capable of.

For each day that passes, machine efficiency is taken more and more for granted. Not only in terms of speed, how often do you visit page two or three during a Google search these days? Not too long ago, checking these pages was standard procedure during thorough investigations. Those days human-defined algorithms and rules decided which query result would appear in your search engine. Nowadays, artificial intelligence and machine learning are not only using search history to learn how to improve its query ranking, but it is also capable of picking up the nuances in web page content, picking up key moments in videos, or returning direct statistics and responses to your query.

Where traditional analysis by humans and computers would be based on limited but long-established experience or rules, these methods could be overwhelmed by the amounts of data collected worldwide. Machine learning uses these masses of data to its advantage, sometimes surpassing traditional methods and sometimes not.

## 1.3 Motivation

Norwegian law forbids any advertisement for alcoholic drinks. Employees at Vinmonopolet are not supposed to be influenced by importers, and they attempt to convey objective rather than subjective advice to customers. Product placement in the stores is strictly sorted by country, district, and price to avoid influencing the customers. Combining this with the monopolistic market, we have a unique opportunity to analyze a sales market with reduced external factors.

Vinmonopolet has a system where the highest-ranked products by sales numbers in each product group are placed in *basisutvalget*. The products in *basisutvalget* are guaranteed a spot on the store shelves, getting a considerable advantage over other products that might only be available in certain stores and otherwise have to be ordered by the consumers. While some importers are pleased to sell their products on a small scale, others aim to get some their products into *basisutvalget*, as this increases the chances of a robust market for their product, giving a stable income/profit. To do this, they need to make an educated guess on what products the Norwegian market will embrace. If they believe a product would do well, they can apply to have it added to *testutvalget*, which functions as a trial period where the product is available in the stores, but at the risk and cost of the importer if the products are not sold.

Wines evolve with age, wineries have varying weather from year to year, and the taste changes drastically depending on what dish they are served with. For an average person, the different flavors of wine are difficult to describe and even more challenging to remember for future references; this leads to consumers using different tactics when buying wine. These include buying the same wines year after year, asking the staff for recommendations based on previous preferences, checking the latest recommendations in the paper, buying from the nearly empty shelves, choosing fancy bottles or labels, or simply aiming for a high alcoholic percentage for the lowest possible price. With such a wide range of consumers, it leads us to the motivation behind this thesis, whether these sales trends can be forecasted in such a fashion that importers can benefit from the results. With the very limited influence an

importer has on the sales performance of their investments, a significant advantage for an importer would be to know which wines they should invest in and which wines that have a large risk leaving *basisutvalget*.

## 1.4 Research Questions

For this thesis we will study our data from a data science perspective, both analyzing the data for trends and connections, and using machine learning to exploit the predictiveness of time series and search for important features. From a business standpoint, the main question we wish to answer is:

- Can red wine rankings be forecasted with enough "precision" to be a beneficial reference when making decisions on which wines to invest in and which wines to stop investing in?

From a scientific standpoint the same question can be formulated into:

- How successfully can we forecast whether products will enter or leave *basisutvalget*?

To answer this and to get a wider perspective on the results, we have multiple smaller research questions we wish to answer as well:

- Can machine learning surpass traditional methods such as forward filling and SARIMAX?
- Which features improve machine learning forecasts the most?
- Does a model trained upon multiple price groups give better forecasts than a model trained on its specific price group?
- Which price groups are easiest to forecast?
- What is a reasonable way to evaluate performance of the ranking?

## 1.5 Contributions

This thesis contributes mainly to two research fields. The first is the field of wine studies. Few studies are made on alcohol sales data in the Norwegian market, and those focus mainly on total amounts of alcohol consumption with regards to the societal issues of overconsumption of alcohol. This is to our knowledge the first study of red wine trends in the time period 2007 to 2018. Though the red wine trends themselves are not the main focus of this thesis, a large part of chapter 2 is designated to analyzing these trends to compensate for lack of relevant background data.

The second research field we will contribute to is the field of forecasting ranks. Most ranking research is associated with information retrieval, where ranking algorithms decide which results answer best the given query. Little background information is found on this topic as well, and we hope that some of the results in this thesis will contribute to further research in this field.

We will be using LSTM networks in our work, but all of the methods we use are standard models that have been thoroughly researched before, none of our work will add new knowledge to this field.

# Chapter 2

# Data

The data for this thesis is collected by Grapespot and presented in three files, called *Sales*, *Rankings*, and *Products* for our purposes. *Sales* contains consecutive sales data for each product for each month in the time period January 2007 to October 2019, with a total of 3,251,078 rows. *Rankings* contains monthly rankings of products based on sales amounts, in the time period January 2007 to September 2019, with a total of 1,435,424 rows. *Products* contains qualitative data for each product, with 183,764 rows. The most descriptive data in *Products*, such as taste, color, and smell, is scarce.

## 2.1 Ranking Lists

The most relevant data for this thesis is stored in *Rankings*. An initial study of the ranking lists shows no proper identifier for each list; thereby, there is no simple method to extract a relevant list for each month. The lists are separated by product group, and the most common products are additionally separated by price group. The top 10 product groups are shown in Table 2.1, where we identify two product groups of red wine that have a large number of products, *Rødvin < 75 g sukker* and *Rødvin < 9 g sukker*.

| Product Group | Translation | Nr. of rows |
|---|---|---|
| Rødvin <75 g sukker per liter | Red wine <75 g sugar per liter | 377266 |
| Hvitvin <15 g sukker per liter | White wine <75 g sugar per liter | 189230 |
| Rødvin <9 g sukker per liter | Red wine <9 g sugar per liter | 169685 |
| Hvitvin <9 g sukker per liter | White wine <9 g sugar per liter | 93029 |
| Øl, overgjæret | Ale | 64423 |
| Musserende vin og champagne <75 g sukker per liter | Sparkling wine and champagne <75 g sugar per liter | 49288 |
| Cognac | Cognac | 39108 |
| Rosévin | Rosé wine | 34386 |
| Skotsk Whisky | Scotch whiskey | 28022 |
| Musserende vin <12 g sukker per liter | Sparkling wine <12 g sugar per liter | 24365 |

**Table 2.1:** The ten most common product groups out of the 117 listed in *Rankings*.

Extracting these two red wine product groups, we see that the first group stopped being used in February 2016. The second group starts being used in March 2016, implying a change in product groups' categorization. An analysis of the red wine categorization changes is presented in Table 2.2, where we see how the sugar limit is changed, and new product

groups and ranking lists are made for large and small bottles. Except for the two most common product groups, the other groups all have less than 4200 rows. We choose to only focus on the two largest groups together, and will simply call this group *Rødvin*. All mentions of red wine after this section are only the products in *Rødvin*.

Product Groups in *Rankings* Changing over Time

| 01.2007-02.2016 | 03.2016-12.2017 | 01.2018-09.2019 |
|---|---|---|
| $<$ 75 g sugar | $<$ 9 g sugar | $<$ 9 g sugar |
| $\geq$ 75 g sugar | $\geq$ 9 g sugar, $<$ 45 g sugar | $\geq$ 9 g sugar, $<$ 45 g sugar |
| | $\geq$ 45 g sugar | $\geq$ 45 g sugar |
| | | $<$ 45 g sugar, $>$ 100 cl |
| | | $<$ 45 g sugar, $<$ 75 cl |
| | | $\geq$ 45 g sugar, $<$ 75 cl |

**Table 2.2:** Product groups for bottled red wine as they change over time, originally only sorted by sugar amounts per liter, then new rankings are set up for above average and below average sized bottles.

Despite lowering the upper sugar limit from 75 g to 9 g, on average there are more wines in the ranking lists per month under the new category. The liquidity of which wines go in and out of the market on a monthly basis should to some degree neutralize the effect of removing the wines with sugar levels between 9 g and 75 g per liter from our data. *Rødvin <75 g sukker per liter* is split into the price groups $[0, 60)$, $[60, 70)$, $[70, 80)$, $[80, 90)$, $[90, 100)$, $[100, 125)$, $[125, 150)$, $[150, 175)$, $[175, 200)$, $[200, 250)$, $[250, 300)$, $[300, 400)$, $[400, 500)$, and $[500, 100000)$. *Rødvin <9 g sukker per liter* is split into the price groups $[0, 100)$, $[100, 125)$, $[125, 150)$, $[150, 175)$, $[175, 200)$, $[200, 250)$, $[250, 300)$, $[300, 400)$, and $[400, 100000)$. We will only study the price groups that overlap with both product groups. All prices mentioned in this thesis are Norwegian krone.

## 2.2 Product Selection

Vinmonopolet has five main sales categories[1]:

- *basisutvalget* - products that sell well enough to establish set procurement deals that guarantee a spot on the shop shelves. Products that are added to this sales category are guaranteed a minimum of 12 months of sales.
- *partiutvalget* - products that sell well enough to establish set procurement deals that guarantee a spot on the shop shelves. Unlike *basisutvalget*, these are procured in a limited quantity, and sales only last until the final product is sold.
- *bestillingsutvalget* - products that are in stock in Norway and available by order. Some of these products may be available in certain shops, based on local preferences.
- *tilleggsutvalget* - products that are available by order but are not guaranteed in stock. The wholesaler can deny delivering orders smaller than a certain quantity. Some of these products may be available in certain shops, based on local preferences.
- *testutvalget* - importers can pay to have their product available in *testutvalget*. The products that are tested in the shops for 6 months, and if they sell well, they increase chances of becoming a part of *basisutvalget*. If they are not sold, Vinmonopolet can return them to the importer, with risk and costs laying on the importer.

[1] https://www.vinmonopolet.no/innkjopsprosess

The products that have the highest sales are placed in *basisutvalget*, and the products in *basisutvalget* sell better due to their availability in the stores, therefore getting a product into *basisutvalget* is often the goal of importers. Products in *partiutvalget* and *tilleggsutvalget* are excluded from the ranking lists, and will therefore not be further analyzed. The remaining products need to have registered sales to be included in the ranking lists. Per September 2019, the 1410 products in *Rødvin* are distributed with 74.2 % in *bestillingsutvalget*, 24.6 % in *basisutvalget*, and 1.2 % in *testutvalget*.

Six times a year, new products are launched. Summing up the products first registered in *Rødvin* in 2018, we find that 1530 products are launched in *bestillingsutvalget*, 41 products are launched in *basisutvalget*, and 10 products are launched in *testutvalget*. The number of products launched in 2018 versus the number of products available in September 2019 show how many products that are tested but never become popular, getting them removed from the market.

## 2.3 Ranking Limit

Discussing high and low rankings can lead to confusion, as a low ranking could be interpreted as a low value and good score, but could also be interpreted as a bad score and therefore a high value. For clarity's sake, we will be using the expression *ranking value*, where a low value is a good score, and a high value is a bad score.

For each separate ranking there is a value, *styringstall*, which sets a ranking limit where products with ranking values below this limit (i.e. better score) are placed in *basisutvalget*. Products with a worse score than the ranking limit are usually placed in *bestillingsutvalget*. An exception to this are products that were first placed in *basisutvalget* less than 12 months previous, they are protected and stay in *basisutvalget* even with a high ranking value. As we can see in Figure 2.1, the ranking limit is frequently changed to adapt to what is currently considered an ideal distribution of number of wines in the various price groups. The change in product group conditions in 2016 is marked, but only the lowest price group has a significant change in ranking limit.

## 2.4 Trends

**Sales**

In Figure 2.2 we see the sales trend of article 4176601, an average popular wine, compared to the sum of all red wines in the category we are analyzing. Clear yearly seasonal trends are visible in both time series. They both show the same tendencies, a spike in sales every December, a slight dip every January, and in general low sales numbers in the summer half of the year. A likely cause for these spikes are various Christmas celebrations, many of which include alcohol in the Norwegian culture. The dips in January could be repercussions of the large alcohol intake in December, where people feel that they have had enough to drink for a while or have the common New Year's resolution to start a healthier life. Low sales numbers in the summer half of the year could be caused by a preference towards white wine or beer in warmer weather.

To get an impression of how the sales are distributed between the price groups, the sum of wines sold in each price group are shown in Table 2.3. Here we observe that the cheaper the wine, the more is bought.

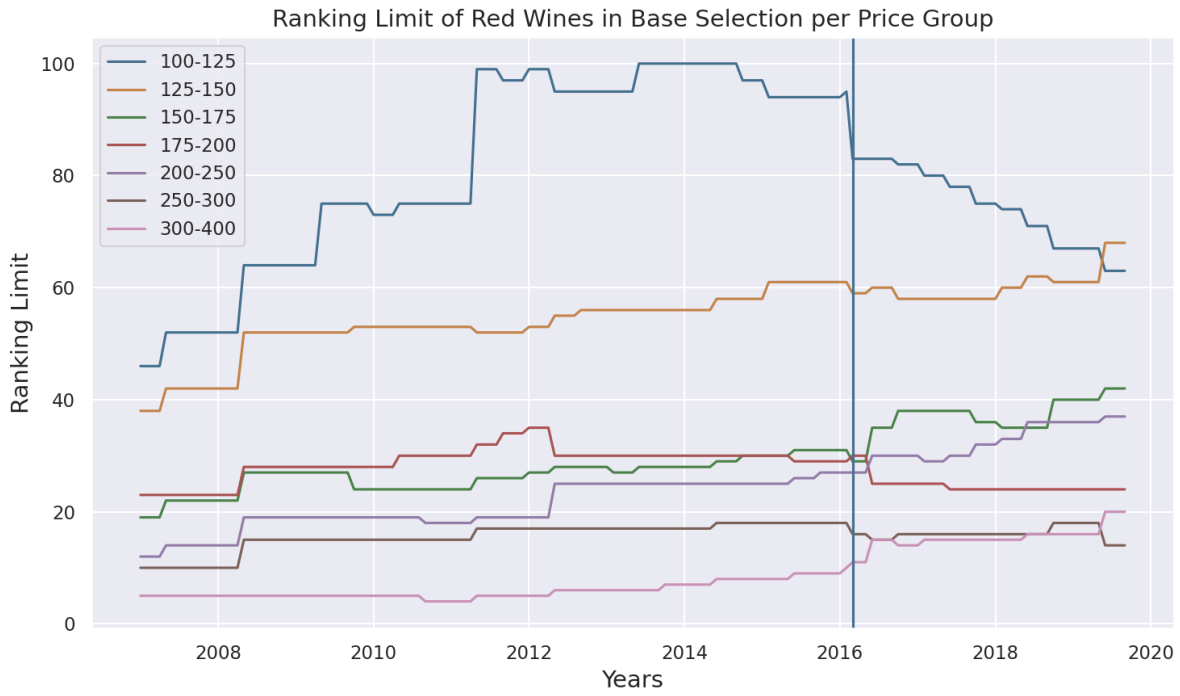



**Figure 2.1:** Plot of the ranking limit per price group that a wine has to surpass to be guaranteed a spot on the shelves. The vertical line marks the date when the main product group changed from *Rødvin < 75 g sukker* to *Rødvin < 9 g sukker*.

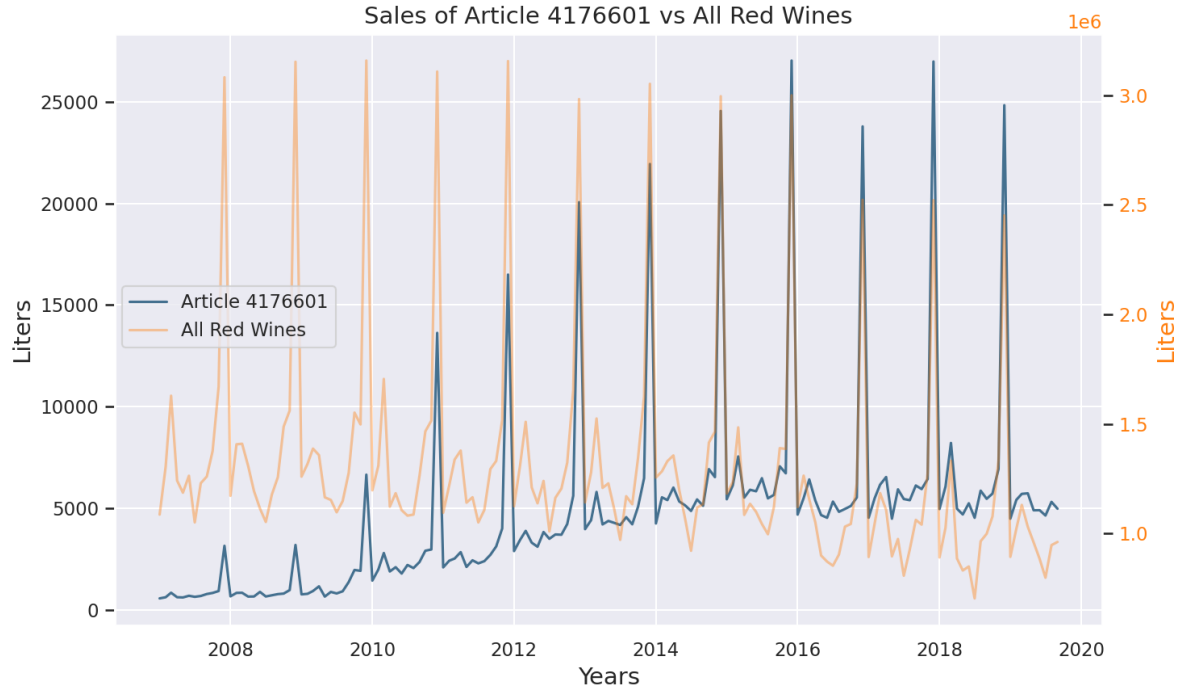



**Figure 2.2:** Clear seasonal trends are visible for both article 4176601 and the total sales of all red wines.

## Rankings

To get an impression of how the ranking value changes for individual wines, we plot the time series of the top five wines in price group 100-125 in January 2007 and September 2019

| Price group | 100 | 125 | 150 | 175 | 200 | 250 | 300 |
|---|---|---|---|---|---|---|---|
| 1000 liters sold | 63437 | 40986 | 14974 | 7706 | 5707 | 2498 | 1490 |

**Table 2.3:** Total liters of red wine sold per price group at *Vinmonopolet* in the time period January 2007 - September 2019.

in Figure 2.3 and Figure 2.4 respectively. One of the first things we notice is that the wines don't necessarily stay in their price range, in the 12 year time period many of the wines go up one or two price groups. Secondly, we observe that the ranking value tends to be slightly lowered when going up a price group. The reason for this could be that the product price is among the lower in its price group, and presumably a large portion of consumers prefer cheaper products. Another reason could be that its previous popularity is unaffected by a slight increase in price, and consumers stick to wines they know that they like. Thirdly, we notice that the lower the rank value is, the more stable the values seem to stay. This makes sense intuitively, as the more popular a wine is, the harder it is for a competing wine to surpass in sales amounts.

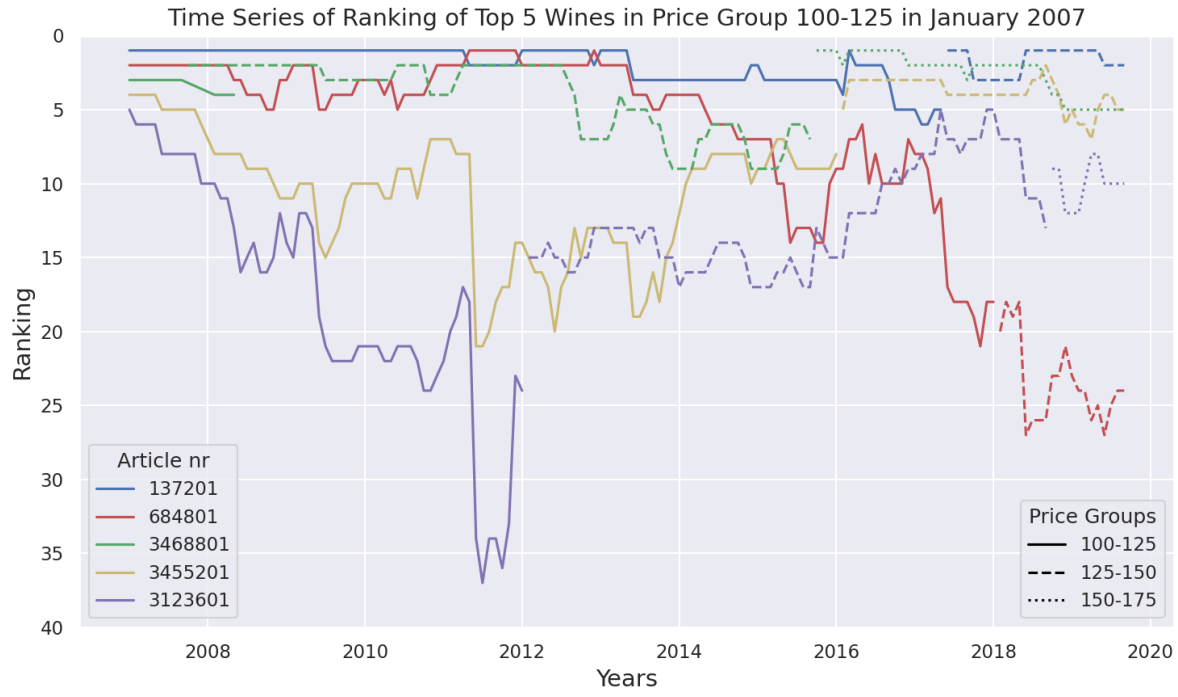



**Figure 2.3:** Time series of the ranking of top 5 wines in January 2007 in price group 100-125 until September 2019. The dashed and dotted lines represent the same wine after it shifted into a new price group, competing for lowest ranking on a new basis of wines.

### Prices

As we observed in Figures 2.3 and 2.4, and also from the changing of price groups in 2016 as discussed in section 2.1, we see that red wine prices seem to increase over time. This is confirmed by following the average price of the wines that were ranked among the top 50 per price group in 2007, as shown in Figure 2.5. The increase in these prices is faster than the increase of the consumer price index and does not show any obvious correlation

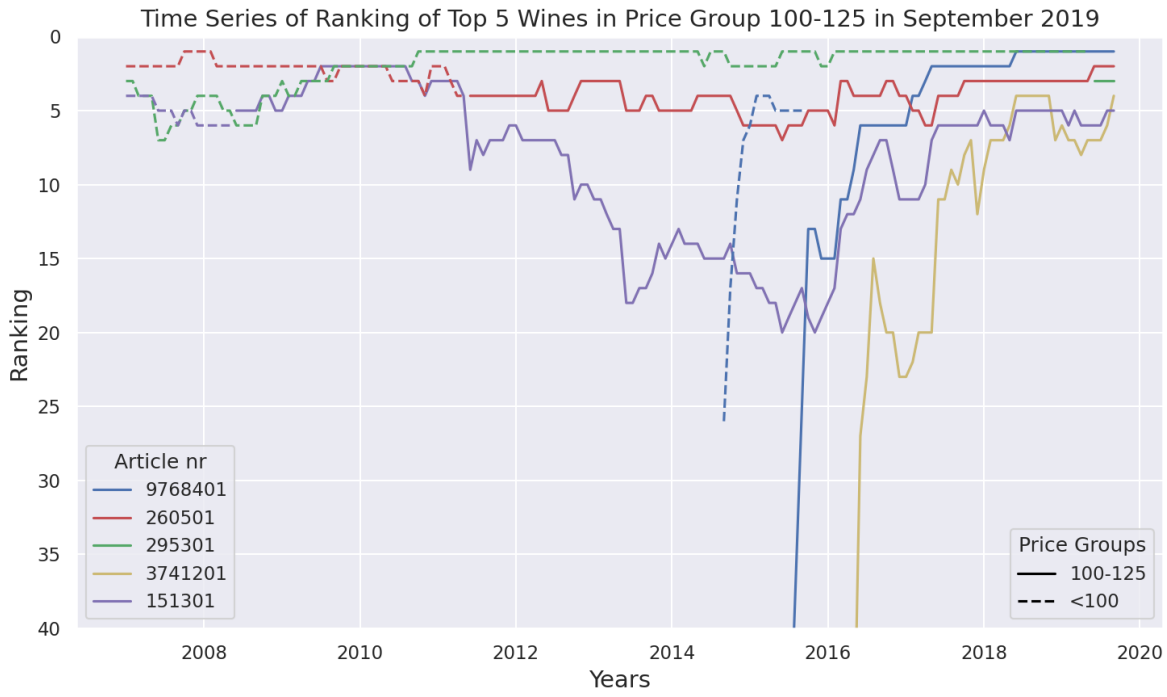



**Figure 2.4:** Time series of the ranking of top 5 wines in September 2019 in price group 100-125 dating back to January 2007. The dashed lines represent the same wine when it was in a lower price group. Before March 2016, there were multiple price groups below 100, making it possible for multiple dashed lines to have the same ranking.

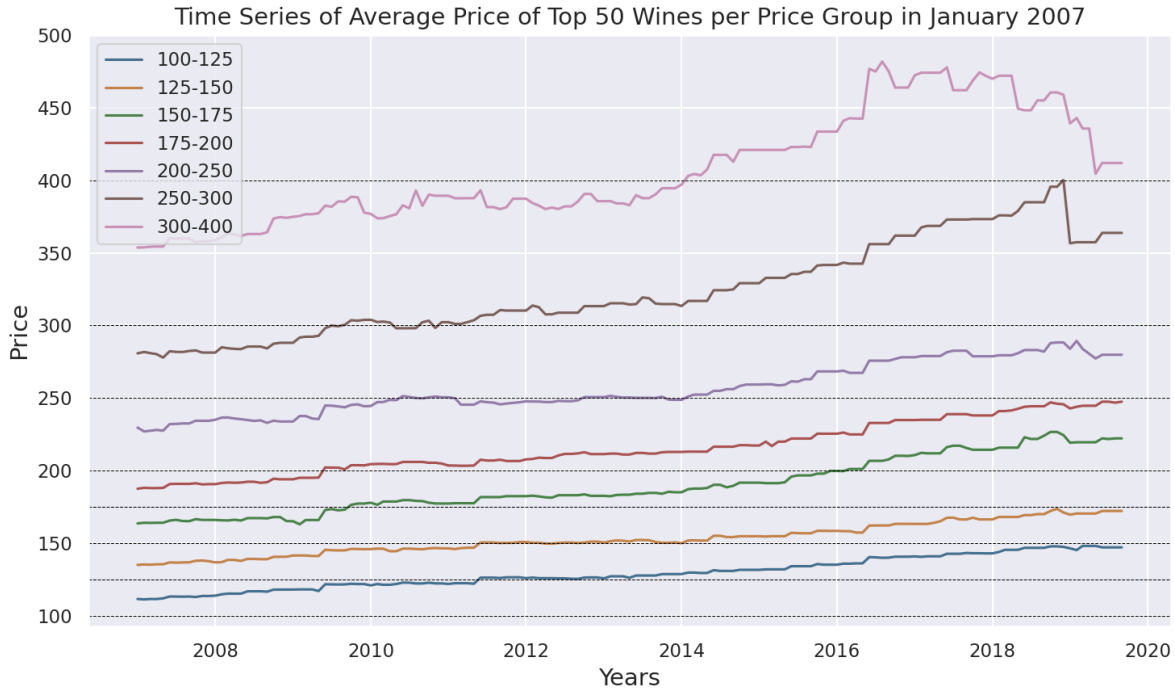



**Figure 2.5:** An average of the prices of the 50 highest ranked wines in January 2007 for all of the price groups. The price ranges are separated by the black dotted lines. Many of the wines dropped off market, especially in the two highest price groups, therefore the averages are based on less data towards the end of the time period.

to the Gross Domestic Product per capita. Despite a near linear trend visible in Figure 2.5, this does not accurately reflect the individual wine prices, which are continuously adapted to the market.

## 2.5 Calculating Rankings

Vinmonopolet's website states that the ranking lists are made every other month, based on the last six months' sales. Figures 2.3 and 2.4 show that ranking lists are updated every month, though this might be done automatically and does not mean that Vinmonopolet acts on the results monthly.

An attempt to reconstruct a ranking list using six months of sales data from *Sales* returns a similar but not identical list. The ranking lists' sales values are based on *Netto Salg* (*net sale*) in *Rankings*. Comparing this value to the calculated six-month sum for article nr 137201, we observe that the calculated value is approximately twice the size of the *net sales* value. The data collector at Grapespot reveals upon inquiry that *net sales* is calculated from the last six months of sales in the 60 largest stores in the country, not from total sales, which is what the sales data in *Sales* represents. Vinmonopolet has, per January 2021, 337 stores, split into categories 1 to 6 depending on size. Category 1 stores have approximately 200 products, while the 60 category 6 stores have at least 1700 products. Only the 60 largest stores guarantee to sell all products from *basisutvalget* and *testutvalget*, as the rest do not have enough shelf space for all products.

Figure 2.6 compares the total sales and sales from the top 60 stores for five popular products. In general, the two time series seem to follow each other closely, but article 684801

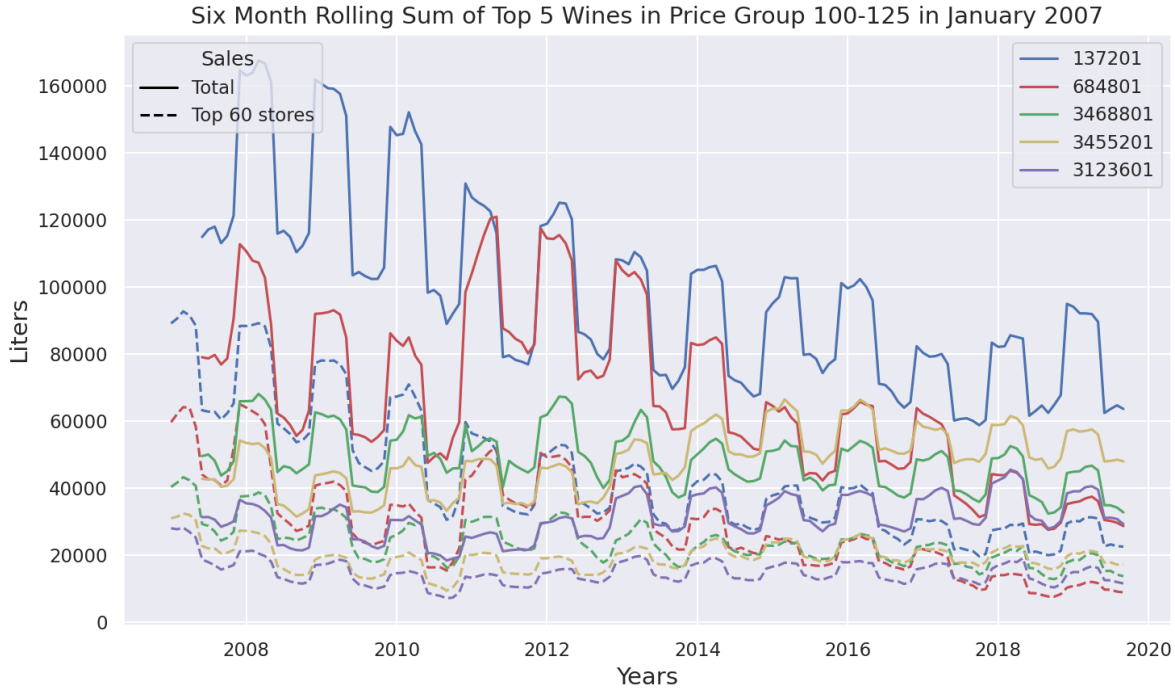



**Figure 2.6:** Six month rolling sum of the ranking of top 5 wines in January 2007 in price group 100-125 until September 2019. The solid lines are total sales and the dashed lines are sales in the top 60 stores. The ranking lists are based on the dashed lines, *net sales*.

shows an example of how the ranking basis does not correctly reflect the total sales for all

months. Figure 2.7 is a scatter plot of the percent change for these two measurement forms. The scatter plot shows a strong positive linear association with a few outliers. The average correlation is 0.958.

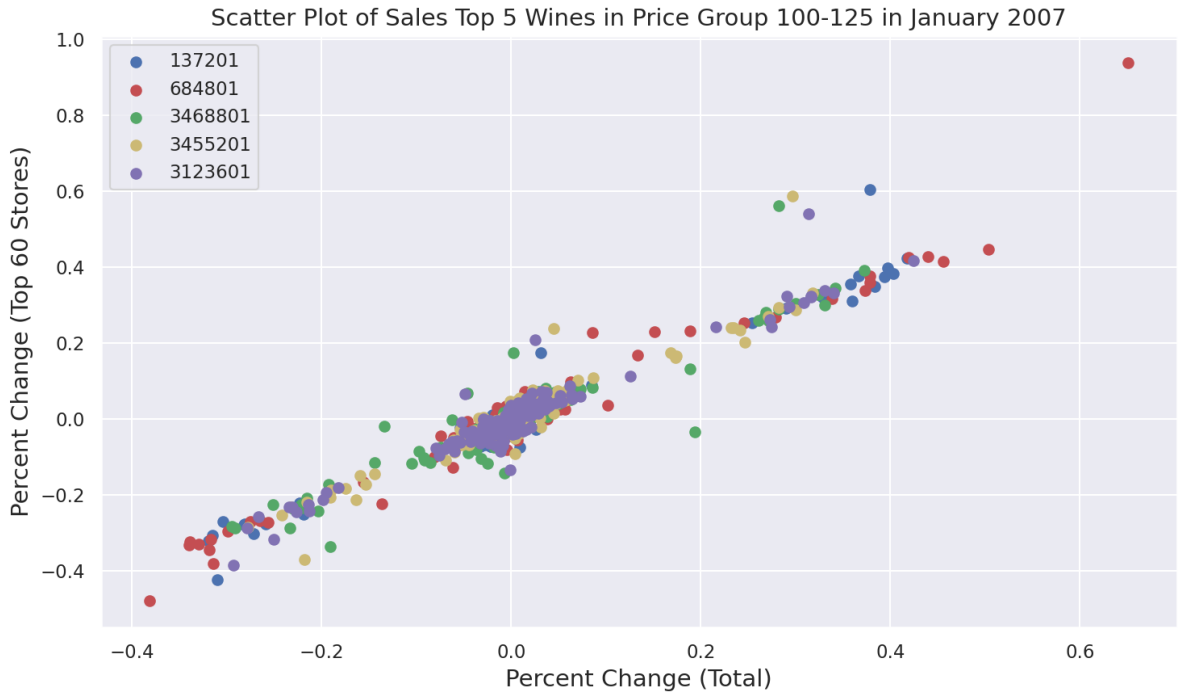



**Figure 2.7:** Scatter plot of the percent change in *net sales* and the six month rolling sum of *liter* for the five top ranked products in PG100 in January 2007.

By comparing Figure 2.3 and Figure 2.1, we know that all of these products remain in *basisutvalget* throughout the observed time period. A selection of five popular wines cannot be assumed to be representative of all red wines, but one can presume that the less common products are even more likely to be sold in one of the 60 largest stores than in the small stores or online compared to the popular products. If a higher proportion of a product is sold in one of the 60 largest stores rather than elsewhere, this will increase correlation, as these sales numbers will be accounted for in both *net sales* and *liter*. This theory is strengthened by the analysis of a less popular red wine in Figure 2.8, where *net sales* and the rolling sum of *liter* follow each other closely.

This presents a couple of options on how to forecast and how to evaluate rank. One option is to focus on the total sales, as a forecast of this represents the actual sales numbers a wholesaler can expect. Ranking based on these values would result in similar, but not exactly correct results, even with training data. Option two is to only forecast the sales in the top 60 stores, as this value defines the rankings that decide which sales category the products are in. A third option is to forecast using total sales and use the correlation of the percent change to transform into expected sales for the top 60 stores before ranking.

From *net sales*, we can either use the values as they are or extract the sales for individual months by taking the difference from month to month. Doing this gives us a slightly shorter training set, but gives a similar basis for forecasting as the total sales data, removing any benefits of option one and three.

The importance for a product to be put in *basisutvalget* is shown in Figure 2.8. The article in question is the red wine that switches sales category most often when disregarding

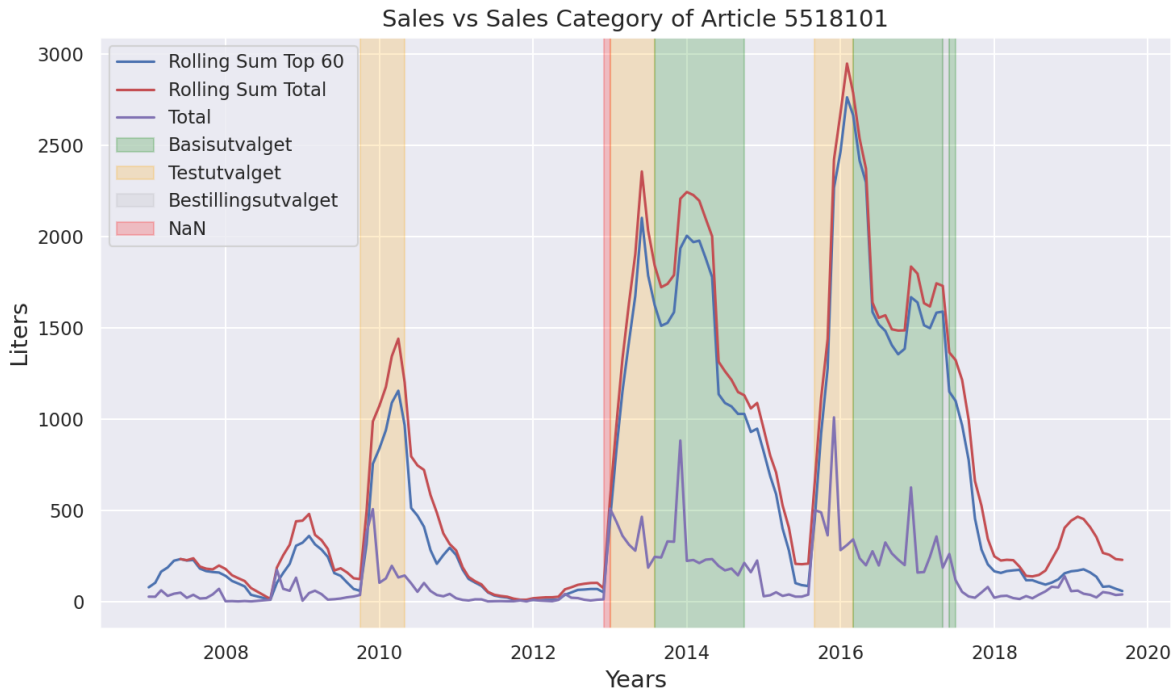



**Figure 2.8:** Sales compared to the sales category of article 5518101. The wine is guaranteed a spot on the store shelves when in *basisutvalget* or *testutvalget*, causing increased sales in these periods. The rolling sum of total sales is added to compare to *net sales* which will be forecasted. Raw total sale is added to show the immediate effect of shifting sales category.

changes to NaN and back. Three times it was put in *testutvalget*, but only two times did it manage to sell enough to enter *basisutvalget*, and then only for a limited period before returning to *bestillingsutvalget*. It is clearly visible from the figure that being available in the store increases sales numbers considerably. The difference between the total sales and *net sales* is small, showing that most of the sales for this product were made in the 60 largest stores.

Of the 2406 products analyzed in PG100-PG300, the number of times each product switched sales category is shown in Table 2.4. These numbers do not distinguish between *Nan* or actual changes, meaning that the actual occurrences of change are lower. This shows that most products seldom switch sales category.

| Changes of sales category | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Occurrences | 527 | 429 | 1062 | 200 | 123 | 38 | 13 | 5 | 6 | 1 | 0 | 1 | 1 |

**Table 2.4:** Number of times a product switches between *basisutvalget*, *bestillingsutvalget*,*testutvalget*, *tilleggsutvalget*, *partiuvalget*, or *NaN* in the time period January 2007 - September 2019.

# Chapter 3

# Theory

## 3.1 Background

#### Wine Studies

Very few studies have been submitted on Norwegian wine sales, but a new study on the effect of temperature and holidays on alcoholic beverages in the USA show similar trends to the ones observed in our data. Here, large spikes appear for thanksgiving and Christmas holidays, with a clear dip in January. As thanksgiving is not celebrated in Norway, the lack of this spike in our data is expected. The study shows that both red and white wine are sensitive to temperatures throughout the year, red wine sales have a dip in the warmer season, while white wine is more popular in the warmer seasons. The temperature sensitivity was highest for the coolest regions [7]. The cooler regions in the USA have a climate more similar to the Norwegian climate, making these temperature sensitivities probable for Norway as well.

A study on the buyer-seller relation in Norwegian wine imports shows that most relationships between importers and exporters are short-lived. More than 75 % end after less than two years. They find that wines with high quality, as assumed by high costs, tend to increase the duration of these relationships. They also reported that the size of the initial trade and a weakening of the currency in exporting country positively impacts duration. A discussion on the exporter-importer ratio highlights the that the limited shelf space due to the monopoly causes a large competition among the importers to sell the products known to sell among the Norwegian consumers [8]. This competition is confirmed in a news article from 2016 which writes about a culture of tough and dirty competition between importers who steal exporters from each other with promises of improved sales. A theory behind this is the availability of the sales data at Vinmonopolet and declining sales from *bestillingsutvalget* after 2010 [9].

A study from 2013 analyzes the Norwegian wine monopoly and the effects on the market. A high tax rate per unit of alcohol means that cheap wines become relatively expensive in Norway, while the expensive products cost similar to abroad. These taxes also make the prices in Norway higher than in neighboring countries, therefore Norwegians have a tendency to buy alcohol abroad, especially in Sweden. The study revealed through interviews that strategic tasting sessions for journalists and supply chain observers are used as marketing strategies [10]. The effect of such strategies are confirmed by a study which finds that a 10 % increase in newspapers' scores lead to a 16-18 % increase in sales of wines [11].

#### Forecasting Time Series

Machine learning is becoming increasingly popular and is proving very successful for tasks such as speech recognition, translations, law usage, and autonomous vehicles [12]. Future competitive advantages of utilizing "big" data are expected to be large, and the benefits of widespread usage of Artificial Intelligence are expected to be increasingly exploited by people and organizations in the future [13].

The success of machine learning in forecasting is a topic of disagreement. A comparison of ARIMA and LSTM for forecasting time series showed that deep learning methods are superior to traditional methods. The empirical study showed that LSTM models obtained 84-87 % reduction in error rates compared to ARIMA [14]. On the other hand, it is suggested in another paper that the papers that claim machine learning superiority in forecasting are limited by conclusions based on too few time series, that the forecasts are mainly short-term, and they are not sufficiently compared to benchmarks. Another concern about machine learning methods, is their lack of capability to specify uncertainty, finding the confidence intervals of forecasts can be just as important as the forecasts themselves [12].

In the M4 competition, where forecasting models were tested on 100 000 time series, the best performing models were hybrids, specifically combinations using statistical and/or machine learning methods. The pure machine learning models had surprisingly poor performance; this is assumed to be caused by overfitting. The top three models used information from multiple time series to predict individual time series [15]. The winning model used a Dynamic Computational Graph Neural Network that mixes a standard exponential smoothing model with an advanced LSTM network into a common framework [16].

#### Ranking Time Series

Learning to rank is an emerging topic, but so far it has mainly been focused on information retrieval. Few studies have attempted to forecast ranking for time-series data, this is normally predicted by experts or survey. A single paper was found on this topic, using a learning to rank algorithm to rank top mobile games. They concluded that using LambdaMART, a combination of Multiple Additive Regression Tree (MART) and LambdaRank, a gradient function, was the best algorithm, and that time attributes improved the performance measure [17].

## 3.2 Performance Measures

### 3.2.1 Ranking Order

To evaluate the results of our ranking forecasts, we need a method to compare two rankings. A common method to measure rank correlation, is Spearman's $\rho$, which measures monotonic relationships between two variables,

$$\rho_{R_1,R_2} = \frac{\text{cov}(R_1,R_2)}{\sigma_{R_1}\sigma_{R_2}}. \tag{3.1}$$

When all of the ranking values are $n$ distinct integers, this can be shortened to

$$\rho_{R_1,R_2} = \frac{6\sum_{i=1}^{n} d_i^2}{n(n^2-1)}, \tag{3.2}$$

where $d_i$ is the difference between the pairwise ranking values. A couple of issues make this metric inadequate for our purposes. We are not necessarily interested in evaluating the correlation or monotonic relationship between the rankings; we know that these exist and want a method to measure how good one ranking compares to the correct ranking. We also wish to limit the evaluation to a section of the ranking; the integer values will therefore not be limited to $[1, n]$ and $d_i$ can get too large to correctly limit $\rho$ between $[-1, 1]$.

In addition to Spearman's $\rho$, we wish to use a score that is more straightforward. This score, $S$, penalizes distance from correct position, but limits the penalty, to avoid letting single large mistakes destroy the score of otherwise good rankings. This score is defined as

$$S = 1 - \frac{\sum_{i=1}^{RL} \max\{|d_i|, l\}}{RL \cdot l}, \tag{3.3}$$

where $d_i$ is the distance from correct ranking value for product $i$, $RL$ is the ranking limit, and $l$ is a chosen limit. This gives us $S \in [0, 1]$ where 0 is a scenario where every ranking value is more than $l$ places off target and 1 is a perfect ranking. We will use $l = 10$ throughout this thesis.

To compare these metrics, we will look at three different rankings, $R_A = [1, 2, 3, 4, 5, 6, 7, 8, 9, 20]$, $R_B = [1, 2, 10, 3, 4, 5, 6, 7, 8, 9]$, and $R_C = [2, 1, 3, 6, 5, 7, 4, 9, 10, 8]$, where $R^* = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]$ is the correct order they are compared to.

$R_A$ shows a ranking where the evaluated values' order is correct. However, a less popular product has a too high score, pushing the tenth most popular product out of the evaluated list. $R_B$ shows a correct ranking except for one product that is given a too high score and shifts all succeeding products one rank lower. $R_C$ has all of the correct products, but the order is quite mixed up.

Table 3.1 shows these rankings score using Spearman's $\rho$ and $S$-score. We observe that Spearman's $\rho$ gives a perfect score to $R_A$ , while the $S$-score penalizes the mistake in the 10th spot. The shift in $R_B$ has a much higher consequence for Spearman's $\rho$ than the $S$-score, while $R_C$ is similar with both metrics.

| | $R_A$ | $R_B$ | $R_C$ |
|---|---|---|---|
| $\rho$ | 1.00 | 0.66 | 0.87 |
| $S$ | 0.90 | 0.86 | 0.88 |

**Table 3.1:** Metrics of three different ranking examples, $R_A = [1, 2, 3, 4, 5, 6, 7, 8, 9, 20]$, $R_B = [1, 2, 10, 3, 4, 5, 6, 7, 8, 9]$, and $R_C = [2, 1, 3, 6, 5, 7, 4, 9, 10, 8]$, compared to a correct ranking $R^* = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]$. $l$ is set to 10 for $S$.

For these short examples, Spearman's $\rho$ varies greatly, but for larger lists, Spearman's $\rho$ gives higher and more stable results. As Spearman's $\rho$ is a tested method, we will be using both measurements to evaluate the ranking results.

### 3.2.2 Classification

To evaluate whether the forecasted rankings fulfill their purpose of forecasting which product selection a product will be placed in, we will classify products by their status change. The classes are defined as

- *in* - product entering *basisutvalget*
- *stay* - product staying in *basisutvalget* or *bestillingsutvalget*

- *out* - product leaving *basisutvalget*.

The confusion matrix for this multi-class classification is shown in Table 3.2, where $\text{tp}_i$ is true positive, $\text{tn}_i$ is true negative, $\text{fp}_i$ is false positive, and $\text{fn}_i$ is false negative for their class $i \in \{in, stay, out\}$.

| | | True | | |
|---|---|---|---|---|
| | | in | stay | out |
| Predicted | in | $\text{tp}_{\text{in}}/\text{tn}_{\text{stay}}/\text{tn}_{\text{out}}$ | $\text{fp}_{\text{in}}/\text{fn}_{\text{stay}}$ | $\text{fp}_{\text{in}}/\text{fn}_{\text{out}}$ |
| | stay | $\text{fn}_{\text{in}}/\text{fp}_{\text{stay}}$ | $\text{tn}_{\text{in}}/\text{tp}_{\text{stay}}/\text{tn}_{\text{out}}$ | $\text{fp}_{\text{stay}}/\text{fn}_{\text{out}}$ |
| | out | $\text{fn}_{\text{in}}/\text{fp}_{\text{out}}$ | $\text{fn}_{\text{stay}}/\text{fp}_{\text{out}}$ | $\text{tn}_{\text{in}}/\text{tn}_{\text{stay}}/\text{tp}_{\text{out}}$ |

**Table 3.2:** Confusion matrix with three classes.

From this, we can calculate the most common performance measures in classification, precision, recall, and the $\text{F}_\beta$ score, given by

$$\text{PRC}_i = \frac{\text{tp}_i}{\text{tp}_i + \text{fp}_i}, \tag{3.4}$$

$$\text{RCL}_i = \frac{\text{tp}_i}{\text{tp}_i + \text{fn}_i}, \tag{3.5}$$

and

$$\text{F}_\beta \text{ score}_i = \frac{(\beta^2+1)\text{Precision}_i\text{Recall}_i}{\beta^2\text{Precision}_i + \text{Recall}_i}, \tag{3.6}$$

where $\beta = 1$ gives the harmonic mean of precision and recall. Precision is also called the positive predictive value and is the fraction of predictions which are correct out of all predictions for that specific class. Recall can be called the positivity rate or sensitivity, and gives us the fraction of correct predictions out of all occurrences of that specific class.

For this project, $\text{RCL}_{\text{in}}$ and $\text{RCL}_{\text{out}}$ are considered the most relevant classification metrics, as we wish to have the highest probability of picking up which products that risk heading out of *basisutvalget* and which products that have a chance of entering *basisutvalget*. $\text{PRC}_{\text{out}}$ and $\text{PRC}_{\text{out}}$ gives us insight on how often predictions of *in* or *out* are correct and is worth optimizing to give credibility to predictions of these two least common classes. The *stay* class is strongly represented and least interesting. Studying $\text{PRC}_{\text{stay}}$ and $\text{PRC}_{\text{stay}}$ can reveal whether the models are too stable and would reveal majority class classification for a direct classification problem. However, a performance measure evaluating ranking can reveal the first issue, and the second issue is irrelevant for our models; we will therefore drop these performance measures. Precision and recall both give valuable information separately, but F1 score is good for comparing results between models, therefore all three performance measures will be used.

### 3.2.3 Degree of Change

To measure the amount of change in the ranking lists, we will present a metric called *Shift*. This metric shows the relative amount of forecasted ranking change to actual ranking change, defined by

$$\text{Shift}_t^h = \frac{\sum_{i=1}^{RL_t} |R_i^{t+h} - R_i^t|}{\sum_{i=1}^{RL_t} |R_i^{t+h^*} - R_i^{t^*}|}, \tag{3.7}$$

where $RL_t$ is the ranking limit at time $t$, $h$ is the number of time steps forecasted, $R_i^t$ is the forecasted ranking value at time $t$ for product $i$, and $R_i^{t^*}$ is the actual ranking value. This metric assumes that the ranking limit stays fixed during forecasted period.

This metric does not say anything about whether changes to rankings are correct, but shows whether a model's sales forecasts in general are more or less stable than the actual changes in sales.

## 3.3 Time Series

Time series are series of observations with equal time between each observation. These can be discrete or continuous, infinite or finite, but each observation needs to be associated with a time $t$. The most common are discrete, finite time series, such as those studied in this thesis. Time series are described differently in many papers and books in this field, but we shall mainly use the mathematical terminology as presented in [18].

We will be using the stochastic process $\{Y_t\}$ for $t = 1, 2, \ldots$, where each $Y_t$ is a random variable, as an example time series. A random walk for $t = 1, 2, \ldots$ can be written as $Y_t = Y_{t-1} + e_t = e_1 + \ldots e_{t-1} + e_t$ where $\{e_t\}$ are unobserved, independent, identically distributed (iid) random variables with mean zero and variance $\sigma_e^2$, called white noise.

Important properties for $\{Y_t\}$ are the mean, autocovariance, and autocorrelation functions. The mean at time $t$ is described by

$$\mu_t = E(Y_t). \tag{3.8}$$

The autocovariance function between observations at time $t$ and $s$ is given by

$$\gamma_{t,s} = Cov(Y_t, Y_s) = E[(Y_t - \mu_t)(Y_s - \mu_s)]. \tag{3.9}$$

The autocorrelation function between observations at time $t$ and $s$ is given by

$$\rho_{t,s} = Corr(Y_t, Y_s) = \frac{\gamma_{t,s}}{\sqrt{\gamma_{t,t}\gamma_{s,s}}}. \tag{3.10}$$

Time series are defined as weakly stationary if $\mu_t$ is constant for all $t$ and $\gamma_{t,t-k} = \gamma_{0,k}$ for all $t$ and $k$. $\{Y_t\}$ is strictly stationary if the joint distribution is the same for $Y_1, Y_2, \ldots, Y_n$ as for $Y_{1+k}, Y_{2+k}, \ldots, Y_{n+k}$ for all $k$ and $n > 0$.

### Backward Shift Operator

To express time series in an orderly fashion, we will be introducing the backward shift operator $B$, defined by

$$BY_t = Y_{t-1}. \tag{3.11}$$

Applying the backward shift operator twice gives

$$B(BY_t) = B^2 Y_t = Y_{t-2}. \tag{3.12}$$

Taking the first difference in a time series, we get

$$Y_t - Y_{t-1} = Y_t - BY_t = (1 - B)Y_t, \tag{3.13}$$

while taking the second difference, we get

$$Y_t - Y_{t-1} - (Y_{t-1} - Y_{t-2}) = Y_t - 2Y_{t-1} + Y_{t-2} = (1 - 2B + B^2)Y_t = (1 - B)^2 Y_t. \tag{3.14}$$

Continuing this pattern we get the $d^{\text{th}}$ difference by $(1-B)^d Y_t$.

To express this, we will be introducing the backward shift operator $B$, defined by

$$BY_t = Y_{t-1}, \tag{3.15}$$

which applied twice gives

$$B(BY_t) = B^2 Y_t = Y_{t-2}. \tag{3.16}$$

Taking the first difference in a time series, we get

$$Y_t - Y_{t-1} = Y_t - BY_t = (1 - B)Y_t, \tag{3.17}$$

while taking the second difference, we get

$$Y_t - Y_{t-1} - (Y_{t-1} - Y_{t-2}) = Y_t - 2Y_{t-1} + Y_{t-2} = (1 - 2B + B^2)Y_t = (1 - B)^2 Y_t. \tag{3.18}$$

Continuing this pattern we get the $d^{\text{th}}$ difference by $(1-B)^d Y_t$.

### 3.3.1 ARMA

Assuming weak stationarity, we will look at the most common traditional models. When the white noise can describe a time series for each previous time step and corresponding weights,

$$Y_t = e_t + \theta_1 e_{t-1} + \theta_2 e_{t-2} + \cdots = (1 + \theta_1 B + \theta_2 B^2 + \ldots)e_t, \tag{3.19}$$

where

$$\sum_{i=1}^{\infty} \theta_i^2 < \infty, \tag{3.20}$$

we have a general linear process. When this process can be modeled with only the last $q$ white noise terms and the remaining weights are zero, this becomes a moving average of order $q$, MA($q$),

$$Y_t = (1 + \theta_1 B + \theta_2 B^2 + \cdots + \theta_q B^q)e_t. \tag{3.21}$$

When a time series can be modeled by regression on the $p$ previous observed values and current white noise, it can be modeled by an autoregressive model AR($p$),

$$Y_t = (\phi_1 B + \phi_2 B^2 + \cdots + \phi_p B^p)Y_t + e_t. \tag{3.22}$$

Combining these two methods gives us the autoregressive moving average, ARMA($p, q$) model

$$(1 - \phi_1 B - \phi_2 B^2 - \cdots - \phi_p B^p)Y_t = (1 + \theta_1 B + \theta_2 B^2 + \cdots + \theta_q B^q)e_t. \tag{3.23}$$

### 3.3.2 ARIMA

Non-stationary time series are time series that can be expressed by

$$Y_t = X_t + \mu_t \tag{3.24}$$

where $X_t$ is the stationary function and $\mu_t$ is a non-stationary function expressing the mean of $Y_t$. For non-stationary time series, we wish to look at the change between consecutive observations to get a stationary time series. If this is not enough, we could also study the change of the change and so forth. If taking the $d^{\text{th}}$ difference of $\{Y_t\}$ gives a weakly stationary time series that can be fitted with an ARMA$(p,q)$ model, we can fit $\{Y_t\}$ with an Autoregressive Integrated Moving Average (ARIMA) model. The ARIMA$(p,d,q)$ model is then given by

$$(1-\phi_1 B - \cdots - \phi_p B^p)(1-B)^d Y_t = c + (1+\theta_1 B + \cdots + \theta_q B^q)e_t, \tag{3.25}$$

where $c$ is the average change between observations. A positive $c$ means that the time series has a positive trend, while a negative $c$ means that it tends to have a negative trend. A random walk with no drift would require an ARIMA(0,1,0) model with $c = 0$, while a random walk with drift would require an ARIMA(0,1,0) with $c \neq 0$.

### 3.3.3 SARIMA

Time series can also have seasonal trends, such as daily temperature variations, sales spikes on weekends or yearly seasons. Removing such a seasonal trend is done by taking a lag-$s$ difference,

$$Y_t - Y_{t-s} = (1-B^s)Y_t, \tag{3.26}$$

where $s$ is the number of seasons. An example of this with lag 12 is given in Figure 3.1, where we also take a regular difference with lag 1 after the seasonal differencing. In this example, we see a large spike every December, which is removed by the seasonal differencing. The second differencing stabilizes the stationarity, but at the expense of an increased standard deviation of the white noise, $\sigma_e^2$.

Similar to above, if $X_t = (1-B)^d(1-B^s)^D Y_t$ can be modeled by an ARMA$(p,q)$ model, then $Y_t$ can be modeled by the seasonal ARIMA model, SARIMA$(p,d,q)(P,D,Q)_s$, where $P,D$, and $Q$ are the seasonal equivalents of $p,d$, and $q$. For this model we will introduce the polynomials $\Phi(B)$ and $\Theta(B)$, which are the seasonal equivalents of the polynomials $\phi(B)$ and $\theta(B)$ we have seen before. They are defined as

$$\begin{aligned}
\phi(B) &= 1-\phi_1 B - \phi_2 B^2 - \cdots - \phi_p B^p \\
\Phi(B) &= 1-\Phi_1 B^s - \Phi_2 B^{2s} - \cdots - \Phi_P B^{Ps} \\
\theta(B) &= 1-\theta_1 B - \theta_2 B^2 - \cdots - \theta_q B^q \\
\Theta(B) &= 1-\Theta_1 B^s - \Theta_2 B^{2s} - \cdots - \Theta_Q B^{Qs}.
\end{aligned} \tag{3.27}$$

Using these, the SARIMA$(p,d,q)(P,D,Q)_s$ model is given by

$$\phi(B)\Phi(B^s)(1-B)^d(1-B^s)^D Y_t = \theta(B)\Theta(B^s)e_t. \tag{3.28}$$

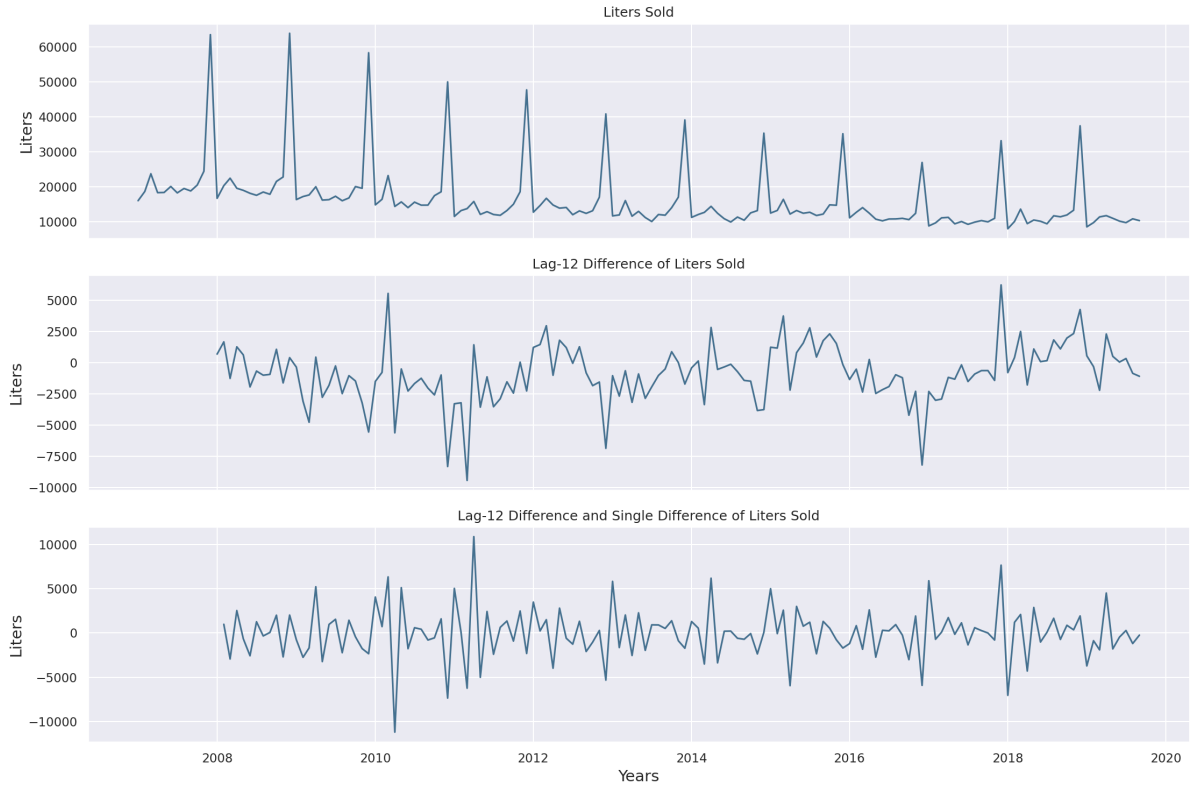



**Figure 3.1:** The top plot is the raw series of sales of article number 137201. The middle plot is the same time series differenced with lag 12. The bottom plot is the time series differenced with lag 12 and once again with lag 1.

The SARIMAX model is a SARIMA model with exogenous factors. The SARIMAX model is given by

$$\begin{aligned} \phi(B)\Phi(B^s)(1-B)^d(1-B^s)^D U_t &= \theta(B)\Theta(B^s)e_t \\ Y_t &= \beta_t X_t + U_t, \end{aligned} \tag{3.29}$$

where $X_t$ are the exogenous variables and $\beta_t$ the coefficients in a linear regression. Exogenous variables are variables that are determined outside of the model before they are imposed on the model. These affect the model without being affected back.

### 3.3.4 Forecasting

Forecasting with ARIMA models is a recursive process starting with calculating $\hat{Y}_{t+1}, \hat{Y}_{t+2}, \ldots$. To forecast $\hat{Y}_{t+h}$, $Y_t$ is isolated on left hand side, all time-steps are shifted by $h$, future observations $Y_{t+1}, Y_{t+2} \ldots Y_{t+h-1}$ are replaced with previous forecasts $\hat{Y}_{t+1}, \hat{Y}_{t+2} \ldots \hat{Y}_{t+h-1}$, future errors are replaced with zero, and the past errors are replaced with the corresponding residuals. We will present an example with ARIMA(1,1,2),

$$(1-\phi_1 B)(1-B)Y_t = (1+\theta_1 B+\theta_2 B^2)e_t, \tag{3.30}$$

which expands to

$$(1-B-\phi_1 B+\phi_1 B^2)Y_t = (1+\theta_1 B+\theta_2 B^2)e_t. \tag{3.31}$$

Applying the backshift operators,

$$Y_t - Y_{t-1} - \phi_1 Y_{t-1} + \phi_1 Y_{t-2} = e_t + \theta_1 e_{t-1} + \theta_2 e_{t-2}, \tag{3.32}$$

and isolating $Y_t$ we get,

$$Y_t = Y_{t-1} + \phi_1 Y_{t-1} - \phi_1 Y_{t-2} + e_t + \theta_1 e_{t-1} + \theta_2 e_{t-2}, \tag{3.33}$$

which we shift by a single time-step

$$Y_{t+1} = Y_t + \phi_1 Y_t - \phi_1 Y_{t-1} + e_{t+1} + \theta_1 e_t + \theta_2 e_{t-1}. \tag{3.34}$$

Assuming that Equation 3.30 is fitted with $\hat{\phi}_1$, $\hat{\theta}_1$, and $\hat{\theta}_2$, we replace $e_{t+1}$ with zero, while $e_t$ and $e_{t-1}$ are replaced with the observed residuals $\hat{e}_t$ and $\hat{e}_{t-1}$. This gives us the forecast

$$\hat{Y}_{t+1|t} = Y_t + \hat{\phi}_1 Y_t - \hat{\phi}_1 Y_{t-1} + \hat{\theta}_1 \hat{e}_t + \hat{\theta}_2 \hat{e}_{t-1}. \tag{3.35}$$

Forecasting $\hat{Y}_{t+2|t}$, is done similarly to Equation 3.35, but by shifting two time-steps, replacing $Y_{t+1}$ with $\hat{Y}_{t+1|t}$, and setting both $e_{t+2}$ and $e_{t+1}$ to zero. This gives us

$$\hat{Y}_{t+2|t} = \hat{Y}_{t+1|t} + \hat{\phi}_1 \hat{Y}_{t+1|t} - \hat{\phi}_1 Y_t + \hat{\theta}_2 \hat{e}_t. \tag{3.36}$$

## 3.4 Machine Learning

Machine learning problems are most commonly split into supervised, unsupervised, and reinforcement learning. Supervised learning is used for classification and regression problems where the target is known. Classification problems predict class labels, while regression problems predict numerical values. Unsupervised learning is used for clustering or density estimation problems where the target is unknown or not used for training. Clustering problems aim to separate data into groups that share similar traits, while density estimation problems estimate the parameters of density functions for the samples. Reinforcement learning is used in environments where the model maps a situation to an action trying to achieve a given goal, using feedback to learn, not fixed data set. Reinforcement problems could be chess games or monitoring sensors and adjusting valves to stabilize systems. We will only be using supervised learning for this thesis.

### 3.4.1 Neural Networks

Neural networks were named by their inspiration from neurons in the human brain, which has approximately 100 billion neurons, linked by an average of 1000 synapses[19]. We are far away from managing to construct such complex networks at this moment, but a smaller example of a neural network is shown in Figure 3.2 to highlight the principles. Each dot is an artificial neuron and the lines between are connections equivalent to synapses. The three neurons on the left represent the input layer, the three columns in the middle represent three hidden layers with six neurons each, and the three neurons on the right represent the output layer. The edges connecting each neuron to all neurons in the closest preceding and succeeding layers make this a fully connected neural network.

The number of hidden layers plus the output layer give the depth of a model, and with increasing depth comes increasing complexity. There is no clear definition, but networks with two or more hidden layers are considered deep learning.

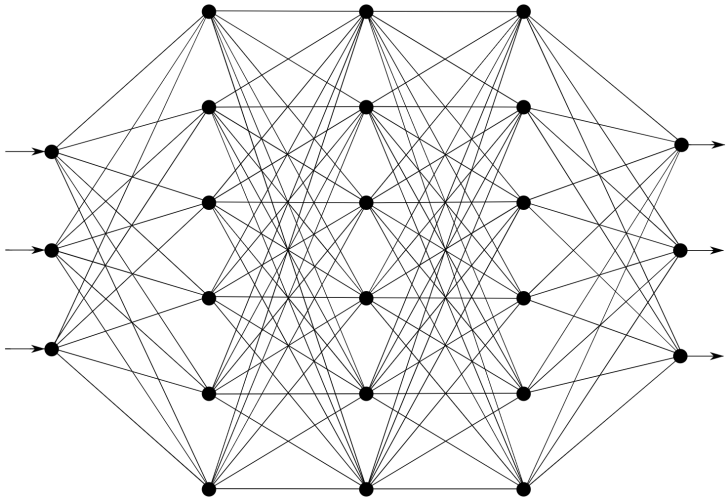


**Figure 3.2:** Example of a fully connected neural network with three hidden layers, each with six neurons.

### Hidden Units

In a feedforward network, each neuron takes input from the preceding layer, calculates a new value, and passes this on to each neuron in the next layer. An example of a neuron in the first hidden layer of Figure 3.2 is shown in Figure 3.3. The neurons in the hidden layers, normally called hidden units, have gates that transform their input with an affine transformation $\boldsymbol{z} = \boldsymbol{W}^T\boldsymbol{x} + \boldsymbol{b}$ before applying an activation function $g(\boldsymbol{z})$. Each layer $l$ has a weight matrix $\boldsymbol{W}_l$ of size (# input × # neurons in $l$) and bias $\boldsymbol{b}_l$, a vector of length (# neurons in $l$). These are updated when the network is training. An example for a single neuron is given in Figure 3.3, where $\boldsymbol{w}^{[i]}$ is the vector of weights from $\boldsymbol{W}$ for neuron $i$ and $b^{[i]}$ is the $i$th bias.

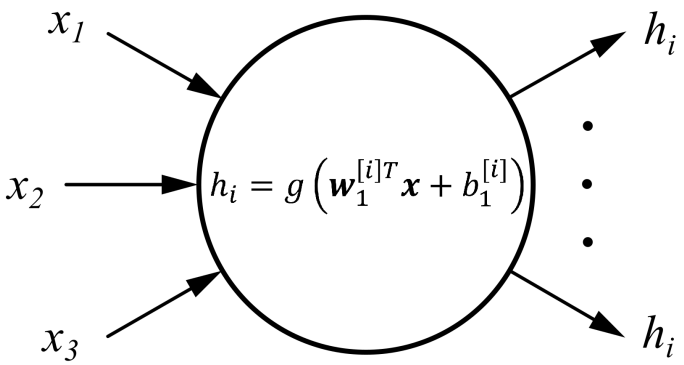



**Figure 3.3:** Example of a neuron with a single gate in the first hidden layer in Figure 3.2, with three input values, weights and bias. Layers with a single gate per neuron are the most common layers, called *dense layers*.

The other hidden layers function similarly, but use input from the previous layer instead of $x$. The activation functions $g(z)$ are applied element-wise to squash the linear transformation with various methods. Three of the most common activation functions are rectified linear unit, ReLU,

$$\text{ReLU}(z) = \max\{0, z\}, \tag{3.37}$$

the logistic sigmoid activation function,

$$\sigma(z) = \frac{1}{1+e^{-z}}, \tag{3.38}$$

and the hyperbolic tangent activation function,

$$\tanh(z) = \frac{e^{z}-e^{-z}}{e^{z}+e^{-z}}, \tag{3.39}$$

which are shown in figure 3.4. The functions differ in linearity, continuity, negativity, and behavior around $z = 0$.

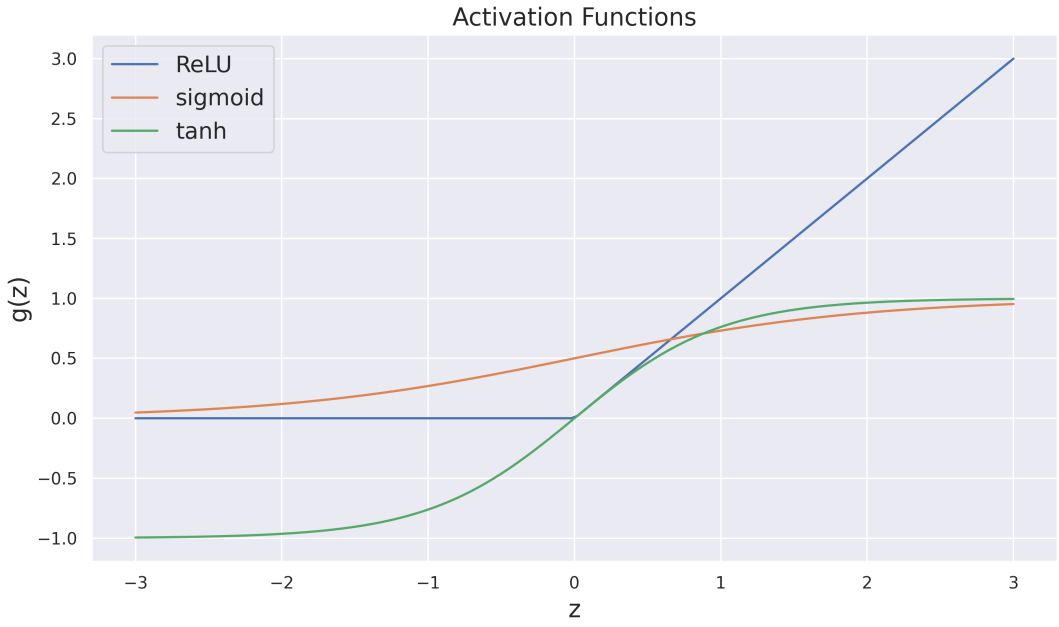



**Figure 3.4:** Three common activation functions in hidden units.

### Optimization

Models are optimized by minimizing the cost function during training. The cost function is the same as the loss function, but the loss function is calculated for each training sample, while the cost function is calculated for the whole training set and can also contain regularization terms. Two standard loss functions for regression models are Root Mean Square Error (RMSE) and Mean Absolute Error (MAE). These are given by

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \tag{3.40}$$

and

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i|, \tag{3.41}$$

where $\hat{y}_i = f(x_i, \boldsymbol{\theta})$ is the predicted value of $y_i$ and $\boldsymbol{\theta}$ consists of all weights and biases in the model. RMSE penalizes large errors more than MAE and has the benefit of being differentiable. MAE is easier to interpret and is more robust with outliers. Considering the

instability of the market we will be analyzing, outliers are to be expected and MAE is the preferred loss function.

When using the loss function over all samples, it is often referred to as the cost function and denoted by $J(\boldsymbol{\theta})$. The negative gradient of the cost function, $-\nabla J(\boldsymbol{\theta})$, gives the direction the parameters need to be adjusted to decrease the cost function the most. Backpropagation is used to calculate the gradient in a time-efficient manner. It calculates the gradients of the last layer first and uses these results and the chain rule to calculate the preceding layers in a backward iterative manner.

In addition to finding which direction the parameters need to be shifted to optimize the cost function, we also need to take into account local minima and the learning rate, how large steps to take, as shown in Figure 3.5. The red arrows indicate a case where the learning rate is set too low; this causes slow progress, as the updates to the parameters are very small. The opposite case is shown with the yellow arrows where the learning rate is set too high, and the cost function can end up diverging instead of converging towards the minimum. The green arrows show a better adaptive learning rate but highlight the risk of getting stuck local minima.

How the weights and biases are initialized impact the occurrences of local minima. The parameters are commonly initialized to small random values. The randomness helps avoiding symmetry, which could lead to similar functions in nodes [20]. The parameters are set to small values to avoid nodes being too active or not active enough, which would make them insensitive to training [21].

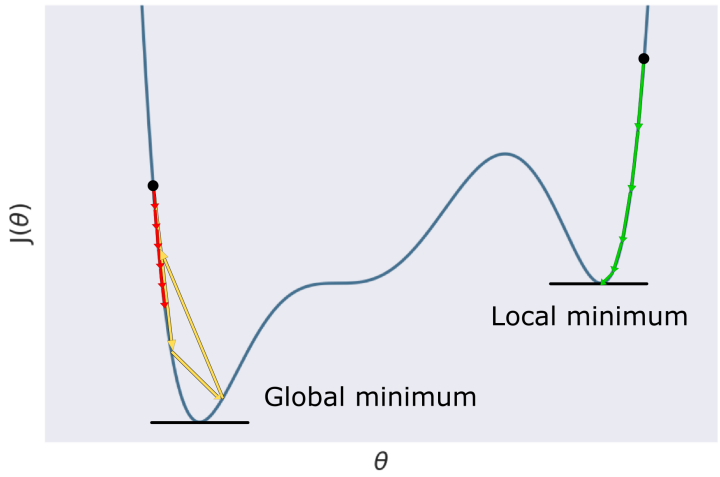



**Figure 3.5:** Plot of the cost function for a parameter in one dimension. The learning rate is too low for the red arrows, too large for the yellow arrows, and good for the green arrows. Without a stochastic element, the function risks getting stuck in local minima, as shown by the green arrows.

A common optimization algorithm is Stochastic Gradient Descent (SGD). Where regular gradient descent calculates the gradient of the cost function on the whole training set, SGD calculates an estimate of the gradient on random batches of training samples [22]. SGD works iteratively, calculating $\boldsymbol{\theta}_{i+1} = \boldsymbol{\theta}_i - \eta \nabla J(\boldsymbol{\theta}_i)$ for $i = 1, ...n$ where $n$ fulfills some stopping criteria and $\eta$ is the learning rate. As shown in Figure 3.5, the learning rate can be adaptive. A common stopping criterion is monitoring the validation loss and stopping after a set number of iterations without improvement, called early stopping. Early stopping is a regularization method that can help prevent *overfitting*. *Overfitting* is a phenomenon that occurs when noise and peculiarities are fit into the model, which increases the training accuracy but can

lead to a less general model which performs worse on new data[23].

A popular stochastic optimization method *Adam* was suggested by Kingma and Ba in 201[24]. Adam is based upon two extensions of SGD, AdaGrad[25] and RMSProp[26], both of which benefit from per-parameter learning rates, giving an advantage on noisy problems and sparse gradients. Adam uses only first-order gradients with small memory requirements, is computationally efficient, is easy to implement, and handles large problems well.

### Recurrent Neural Networks

The earliest and most "straightforward" neural networks are *feedforward neural networks*. These are networks where information only flows in one direction, without any cycles forming between or within the neurons. *Recurrent Neural networks* (RNNs) are networks with internal memory. This memory is constructed by feeding the output back into the cell to evaluate both the new input and the previous output for each computation. Feeding a single time series $x = [x_0, x_1, ..., x_T]$ into a simple RNN, as shown in Figure 3.6, the first sample $x_0$ has no internal memory, while the last sample will have memory from all previous samples influencing the internal input $h_{T-1}$. This can be denoted by

$$h_t = g(b + \boldsymbol{W}^T x_t + \boldsymbol{U}^T h_{t-1}) \tag{3.42}$$

where $\boldsymbol{U}$ is a matrix of the recurrent weights.

A simple model such as this can have a poor long-term memory, as the first hidden outputs are gradually overloaded by the newer input. This issue is handled by the long short-term memory (LSTM) network, suggested by Hochreiter and Schmidhuber in 1997[27].

An advantage to RNNs is their ability to operate over sequences of vectors and their flexibility regarding input and output shape, one-to-one, one-to-many, many-to-one, many-to-many. Allowing the use of sequences as input lets the model assume ordered dependency among samples in the same sequence, making analyzing time series possible. Some of the disadvantages to RNNs are that they are difficult to train, the recurrent nature can cause slow computation, and they are prone to the vanishing or exploding gradient problem.

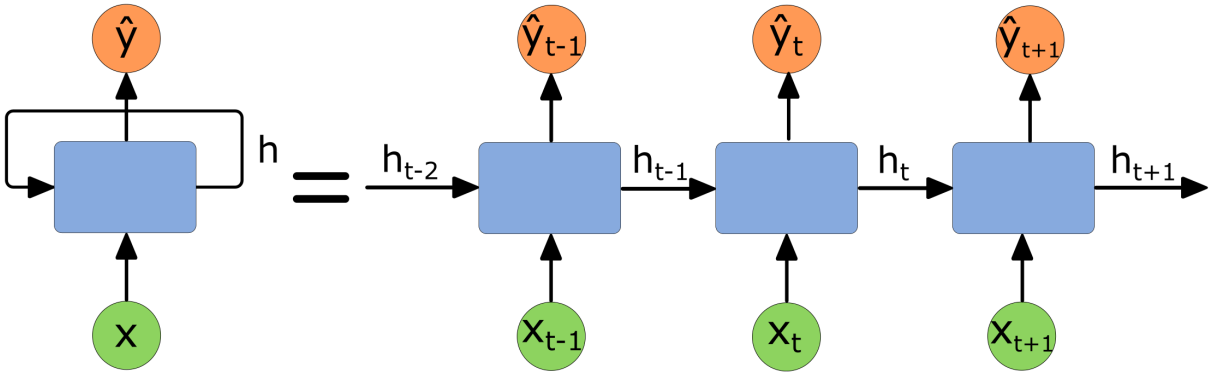



**Figure 3.6:** Example of a recurrent neural network, in folded form on the left and unfolded form on the right.

### Vanishing Gradient Problem

The vanishing gradient problem is an issue that can occur when using gradient-based optimization methods and backpropagation. Vanishing gradients are gradients that become too

small for weights and biases to know which direction they should move. This effect occurs as new gradients are calculated by multiplying multiple previous gradients using the chain rule from backpropagation. For an RNN, this can be shown by

$$\frac{\partial \mathrm{J}}{\partial \boldsymbol{\theta}} = \sum_{t=1}^{T} \frac{\partial \mathrm{J}_t}{\partial \boldsymbol{\theta}} \tag{3.43}$$

where

$$\frac{\partial \mathrm{J}_t}{\partial \boldsymbol{\theta}} = \frac{\partial \mathrm{J}_t}{\partial h_t} \frac{\partial h_t}{\partial h_k} \frac{\partial h_k}{\partial \boldsymbol{\theta}}, \tag{3.44}$$

and $\frac{\partial h_t}{\partial h_k}$ is a product of varying length, dependent on $t$ and $k$,

$$\frac{\partial h_t}{\partial h_k} = \prod_{i=k+1}^{t} \frac{\partial h_i}{\partial h_{i-1}} = \prod_{i=k+1}^{t} \frac{\partial h_i}{\partial z_i} \frac{\partial z_i}{\partial h_{i-1}} = \prod_{i=k+1}^{t} g'(z_i) \boldsymbol{U}^T. \tag{3.45}$$

For the last term we have used Equation 3.42 such that $h_i = g(z_i)$ and $\frac{\partial (b + \boldsymbol{W}^T x_t + \boldsymbol{U}^T h_{t-1})}{\partial h_{t-1}} = \boldsymbol{U}^T$.

The derivative of the $\sigma$ and hyperbolic tangent functions $g'(z_i)$ are

$$\frac{d\sigma(z)}{dz} = \sigma(z) \cdot (1 - \sigma(z)) \tag{3.46}$$

and

$$\frac{d \tanh(z)}{dz} = \frac{\cosh^2(z) - \sinh^2(z)}{\cosh^2(z)}, \tag{3.47}$$

both of which are limited to $[0,1]$, as opposed to ReLU, which has the derivative

$$\frac{d\mathrm{ReLU}(z)}{dz} = \begin{cases} 0 & \text{for } z < 0 \\ 1 & \text{for } z \geq 0. \end{cases} \tag{3.48}$$

The product of $\sigma$ and hyperbolic tangent activation functions will therefore have an increasing minimizing effect with increasing width of an RNN.

Given the eigendecomposition $\boldsymbol{U}^T = \boldsymbol{V} \mathrm{diag}(\boldsymbol{\lambda}) \boldsymbol{V}^{-1}$, we have after $t$ steps

$$\boldsymbol{U}^{T^t} = (\boldsymbol{V} \mathrm{diag}(\boldsymbol{\lambda}) \boldsymbol{V}^{-1})^t = \boldsymbol{V} \mathrm{diag}(\boldsymbol{\lambda})^t \boldsymbol{V}^{-1}. \tag{3.49}$$

If any of the eigenvalues $\lambda_i$ have an absolute value that is not close to 1, they will either vanish if they are smaller than 1 or explode if they are greater[28]. The second case introduces an alternative problem, called the *exploding gradient problem*, where exploding gradients make learning unstable.

Vanishing and exploding gradients are not only an issue for wide RNNs, deep feedforward neural networks can also face this problem. One proposition to overcome this problem was made by Schmidhuber in 1992, suggesting to divide and conquer by using a multi-level hierarchy of recurrent networks, pretraining each level separately with unsupervised learning before final tuning through backpropagation [29]. The most common technique to avoid the vanishing gradient problem is to use an LSTM network.

### 3.4.2 LSTM

The neural networks we will use in this thesis are all LSTM networks. LSTMs are a form of RNNs which handle long-term memory better than traditional RNNs. In addition to the normal hidden state $h_t$, LSTMs pass on a cell state $c_t$ which is only slightly adjusted per cell, preserving training from old input. A diagram of an LSTM cell is shown in Figure 3.7, where we can observe how four gates interact to update the cell state and hidden state. Each of these gates have their own bias $b$, input weights $\boldsymbol{W}$, and recurrent weights $\boldsymbol{U}$.

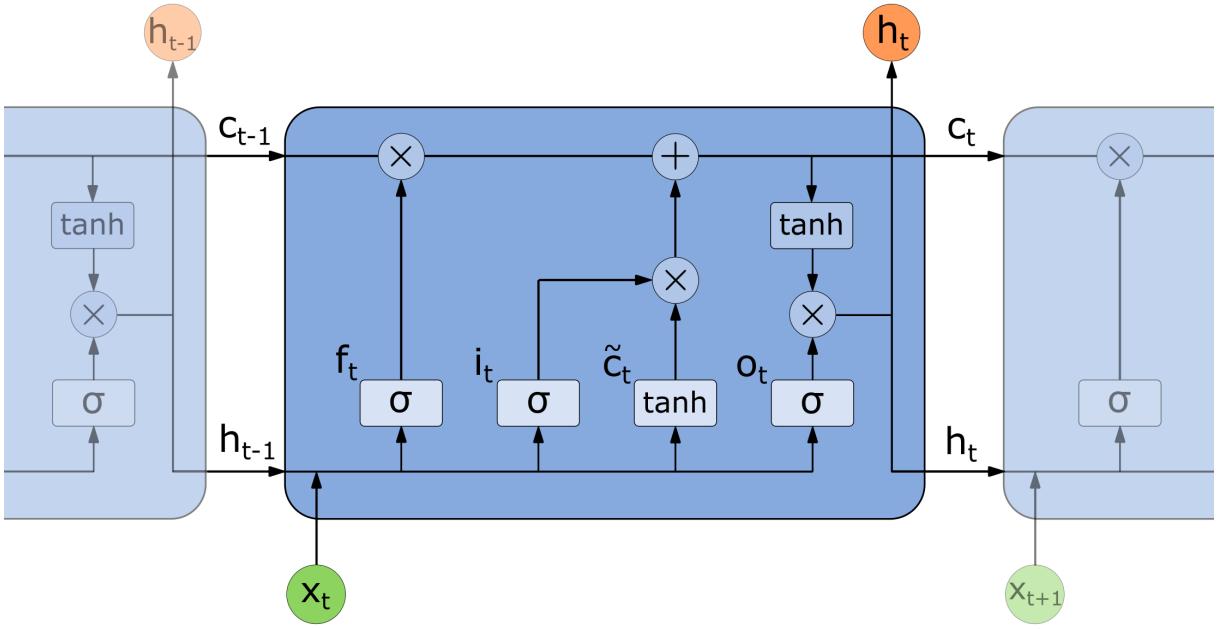



**Figure 3.7:** Diagram of an LSTM cell. It differs from the traditional RNN cell in Figure 3.6 by the additional cell state $c_t$ and internal gates.

The four gates are the forget gate

$$f_t = \sigma\left(b^f + \boldsymbol{W}^{fT}\boldsymbol{x}_t + \boldsymbol{U}^{fT}\boldsymbol{h}_{t-1}\right), \tag{3.50}$$

the input gate

$$i_t = \sigma\left(b^i + \boldsymbol{W}^{iT}\boldsymbol{x}_t + \boldsymbol{U}^{iT}\boldsymbol{h}_{t-1}\right), \tag{3.51}$$

the cell update gate

$$\tilde{c}_t = \tanh\left(b^{\tilde{c}} + \boldsymbol{W}^{\tilde{c}T}\boldsymbol{x}_t + \boldsymbol{U}^{\tilde{c}T}\boldsymbol{h}_{t-1}\right), \tag{3.52}$$

and the output gate

$$o_t = \sigma\left(b^o + \boldsymbol{W}^{oT}\boldsymbol{x}_t + \boldsymbol{U}^{oT}\boldsymbol{h}_{t-1}\right). \tag{3.53}$$

The forget-, input-, and output gates start out the same with random bias and weights, but their weights and biases are adjusted differently as the model is trained, giving them functions fitting their names.

The cell state is given by

$$c_t = f_t c_{t-1} + i_t \tilde{c}_t. \tag{3.54}$$

The forget gate has a $\sigma$ activation function giving values in the range [0,1], therefore multiplying this with the input cell state $c_{t-1}$ lowers the values, having a "forgetting" effect. Adding to this the product of a $\sigma$ activation function and a tanh activation function with

ranges [0,1] and [-1,1] adds new memory to the cell state without overbearing the previous memories.

One copy of the cell state is passed on to the next recursion, while the other is squashed by a tanh function before it is multiplied with the output gate to construct the hidden state

$$h_t = \tanh\left(c_t\right)o_t, \tag{3.55}$$

which is passed on to the next recursion and the next layer. As the hidden state is in the range [-1,1], LSTM networks normally have a dense output layer with an activation function fit for the problem in hand.

# Chapter 4

# Experimental Setup

The experiments for this thesis will only be executed on the largest group of red wines. The principles behind the analysis and the bulk of the code could be used on any product sold by Vinmonopolet, though smaller sales numbers and fewer products are assumed to give a poorer basis for machine learning.

The red wines are grouped by price before ranking, as described in section 2.1. The price groups we will analyze are named PG100, PG125, PG150, PG175, PG200, PG250, PG300. The number in each name expresses the lower limit of the price range, while the upper limit is given by the next price group, with the exception of PG300 which has the price range $[300, 400)$.

All models are tested on targets in the time period August 2017 to September 2019, with training data size and input selected as needed for each model. Only one price group is evaluated at a time, and only products that are in the evaluated price group at month $t$ are kept in the test set. The size of the tests sets are presented in Table 4.1.

| Price group | 100 | 125 | 150 | 175 | 200 | 250 | 300 |
|---|---|---|---|---|---|---|---|
| Test samples | 3480 | 4675 | 2282 | 2101 | 2307 | 2723 | 1585 |

**Table 4.1:** Number of samples that each price group is tested on.

No data in the time period January 2007 to December 2007 was used in any of the experiments presented in this thesis. This data was kept available for taking the 12 month difference of *net sales* without large changes to the code. Initial experiments showed little or no improvement upon using differenced data, and no further experimentation was done on this topic.

## 4.1 Data Processing

A large part of the time spent on this project was spent handling the data, shaping it into a format that could be fed into the LSTM, making the code easily adaptable to varying types and amounts of features and input sizes, and setting the results up in ranking systems. We will give a simplified walk-through of the process below, specifically for the LSTM model. The setup for the SARIMA model and persistence forecasts are based on the same process but with a few variations. A link to the Github repository is given in Appendix C.

### Merging Data

One of the challenges with the data is that it is presented in three files. These files have some overlapping features, but not all of these are guaranteed to have the same values from file to file. An example of this is the alcohol percentage feature in *Sales* and *Products*. In *Products*, a product can have multiple rows, one for each vintage, all with separate alcohol percentages. *Sales* has only one row for each product for each month, with a single value for alcohol percentage and a single value for vintage, despite multiple vintages of this product being sold simultaneously. Merging these columns we observe that the vintage-alcohol combinations are not consistent with the two files.

Not all of the rows in *Sales* are included in *Ranks*. By first left merging *Ranks* and *Sales*, then left merging this with *Products*, we produced a single data frame which only includes products with a ranking value. The merging process can produce rows which are near duplicates, with only one or few columns differing. Cases like these can occur when the same products are delivered by multiple importers. The duplicates were removed by aggregating the liters sold and applying this value to the first occurring case while removing the rest. Removing all unnecessary columns, we have a single dataframe with rows containing date, article number, and all features specific for that product at that date, such as ranking, ranking limit, price, etc..

### Sorting Data

The ranking lists are divided by price groups, therefore, to analyze each ranking list separately, the data frame needs to be split into separate data frames for each price group. The downside to doing this, is that these data frames only have rows for their products as long as they are in that specific price group, losing historic data.

The solution to this issue was to identify all products that ever appear in the price group(s) we are training on, gather all rows on these products, and later remove excess data from the test set. By excess data, we mean all samples that are not in the price group of the ranking list we are evaluating at time $t$.

To make all features easily callable, separate data frames were constructed for each feature, using article numbers as columns, dates as indices, and filling them with that specific feature's values.

### Constructing Samples

The samples were constructed as arrays. These arrays contain one-hot encoded features, time dependent features, article number, and date, in that specific order. Each one hot encoded feature had a width equivalent of the number of options for that feature, i.e. *bestillingsutvalget* was coded with [0,1,0,0], giving this feature a width of four. A sample containing only a single one-hot encoded feature of width four and *net sales* with 20 months input and three months output would have the shape $S[i] = [o_1, o_2, o_3, o_4, n_{t-19}, n_{t-18}, \ldots, n_t, n_{t+1}, n_{t+2}, n_{t+3}$, article nr., $t]$, where $i = 0, 1, \ldots, I$ where $I$ is the total number of samples, $t \in$ {August 2009 - June 2019}, and the article numbers are given by the ranking lists at time $t$. By gathering all the information about a sample into a single array, we can keep track of the samples and extract necessary information using indexing when necessary. A separate

function takes the input $X$ and output $y$ from the samples, giving them the shapes

$$X[i] = \begin{bmatrix} o_1 & o_2 & o_3 & o_4 & n_{t-19} \\ o_1 & o_2 & o_3 & o_4 & n_{t-18} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ o_1 & o_2 & o_3 & o_4 & n_t \end{bmatrix} \quad \text{and} \quad y[i] = \begin{bmatrix} n_{t+1} & n_{t+2} & n_{t+3} \end{bmatrix},$$

and feeding them into the LSTM.

**Ranking**

By adding the forecasted values of the test set to the end of $S[k]$, where $k \in$ test samples, and removing the values used as input, we get $\hat{S}[k] = [\text{article nr.}, t, \hat{n}_{t+1}, \hat{n}_{t+2}, \hat{n}_{t+3}]$. This set of forecasts is made into a data frame with article number and date as indices.

A subset of the original data frame containing *ranking* is merged with the forecasts to get this feature at time $t$. Shifting the subset with one month and merging again, we get the same feature for time $t+1$. Repeating this two more times, we get a data frame containing the ranking values for each product at times $t$, $t+1$, $t+2$, and $t+3$. By also adding the ranking limit to the data frame, it is possible to rank the forecasts and compare them to the actual ranking values and extract all necessary information to evaluate the results using various performance measures.

## 4.2 Models

### 4.2.1 Persistence Forecast

For a simple baseline, we will use persistence forecasts on *net sales* for $t+1$, $t+2$, and $t+3$ before ranking the wines for all three months. By assuming persistent sales instead of persistent ranking values, we have the possibility to compare sales MAE with the other models as well as the ranking results. In addition, this method avoids missing and duplicated ranking values when products shift between price groups.

The MAE is expected to be high for the persistence forecast, as the sales values have high seasonal variations. The ranking scores are not expected to be impacted as highly by the seasonal variations, as all products are within the same product group and should have somewhat parallel sales trends.

### 4.2.2 SARIMAX

Finding separate SARIMA coefficients for each time series is a highly time consuming process, therefore multiple time series were analyzed and the combination of coefficients which most often gave the best model were chosen for a global model. This was a SARIMA$(1,1,0)$ $(1,0,1)_{12}$ model. The coefficients and residuals are found for each time series for each product. The input for fitting each model is from January 2008 to $t$, where $t$ is in the range July 2017 to June 2019.

For the few time series that did not manage to use the SARIMA model, a persistence forecast was filled in instead. Two exogenous features were tested on the SARIMA model, *Top sales* and *Price*, both described further down.

### 4.2.3 LSTM

After extensive testing, the model which gave the most consistent good results and had a reasonable runtime was a model consisting of 2 LSTM layers and 1 dense output layer without activation function, each with 64 neurons. The Adam optimizer with learning rate 0.001 was used, with MAE as loss function and accuracy as metric. The models had 20 months input and 3 months output. The training and validation sets were resampled enough to be split into batches of size 64. Weights from the model were copied to an identical model with batch size 1 for online forecasting, making the number of test samples unaffected by batch size. Epochs were set to 100, but an early stopping function with patience=4 on validation loss stopped the learning after the model stopped improving, usually between 15 and 40 epochs.

The models were trained on data from the time period August 2009 to November 2015 and validated on data from December 2015 to December 2016. These dates $t$ do not include the 19 preceding months used in the input $[x_{t-19}, x_{t-18} \ldots, x_t]$ or 3 succeeding months used as targets $[y_{t+1}, y_{t+2}, y_{t+3}]$. A quarantine period between the validation set and test set of 6 months was added to lower time dependent bias on the test results.

To avoid flooding the training and validation sets with data from unpopular wines, only products that have at some point had a ranking value $R \leq 1.2 \times \max\{RL_{pg}\}$ were used. The $\max\{RL_{pg}\}$ is the highest ranking limit that has been in use per evaluated price group; these are visible in Figure 2.1. By training on some products that never reached *basisutvalget*, we hope that the model will better recognize these bad trends in new products.

## 4.3 Features

The targets we are forecasting are *net sales*, but in addition to using *net sales* as input, we will be testing different features to see if they contain data that improve the LSTM learning. Appendix B gives a full overview of the columns in *Sales*, *Rankings*, and *Products* containing data and metadata. Many of the available metadata classes are interesting to analyze and use as features, but only a few with specific qualities are used. These qualities are lack of scarcity, assumed relevance, and that the values are quantifiable or can be one-hot encoded with limited variables.

The features that are one-hot encoded are:

- *Price group* - whether a product is in price group $[0, 100)$,$[100, 125)$, $[125, 150)$, $[150, 175)$, $[175, 200)$, $[200, 250)$,$[250, 300)$, $[300, 400)$, or $[400, 100000)$, width=9
- *Selection* - whether a product is in *basisutvalget*, *bestillingsutvalget*, *Testutvalget*, or other, width=4
- *Newness* - whether a product was first sold within 12 months, width=1

These features are only added for month $t$. The features that are time dependent and added for $[t-19, t-18, \ldots, t]$ are:

- *Price - Segmentpris*
- *Top sales - Net sales* for the wine currently ranking as number one in the evaluated price group

These features, along with *net sales* are scaled using a Min-Max scaler.

## 4.4 Ranking

Despite the focus on time series forecasting of liters sold per wine, the main goal is to forecast whether a product is likely to leave *basisutvalget* or enter it the next three months. To do this, we need to rank the products correctly, in the same fashion Vinmonopolet does. Three different variables need to be decided for our task: *net sales* for all products, *price* for all products, and *ranking limit* for the specific price group. The product prices are set by the importers and the ranking limit is set by Vinmonopolet, both of these variables are relatively stable with mainly small changes. To focus our work on the predictability of sales, we will therefore assume that prices stay constant for the next three months and that the ranking limits are known.

Assuming that the prices are constant rather than that they are known, leads to products appearing in the correct ranking lists for $[t+1, t+2, t+3]$, that are not evaluated in the forecasted ranking lists. Likewise products will disappear from the correct ranking lists when their new price is out of the boundary of the price group. These occurrences will both negatively impact the performance measures, but this is considered more realistic than knowing the true prices and less problematic than forecasting the prices and having a combination of this issue and forecasting errors.

From the ranking lists we can classify the products as defined in subsection 3.2.2. The number of occurrences of each class in each price group during the testing period is shown in Table 4.2.

| Price Group | 100 | 125 | 150 | 175 | 200 | 250 | 300 |
|---|---|---|---|---|---|---|---|
| in | 14 | 33 | 13 | 9 | 16 | 6 | 11 |
| stay | 2835 | 1904 | 1173 | 990 | 1024 | 513 | 558 |
| out | 31 | 31 | 14 | 9 | 16 | 9 | 7 |

**Table 4.2:** Nr of occurences of each class at time $t+1$ for each price group in the time period August 2017 - September 2019. *In* is the class of products entering *basisutvalget*, *stay* is the class of products staying in *basisutvalget* or *bestillingsutvalget*, and *out* is the class of products leaving *basisutvalget*.

## 4.5 Experiments

The experiments are split into five groups, this lets us focus separately on the different price groups, features, and input size. The experiments are:

- Experiment 1: Testing simple models on each price group to see whether some price groups are easier to forecast than others
- Experiment 2: Testing different features on PG175 with only PG175 as input to see if these can improve forecasts
- Experiment 3: Testing different price groups as input on PG175 to see if diversity and a larger training set improves forecasts
- Experiment 4: Combining the best results of the two previous bullet points to search for an optimal model for PG175
- Final model: The resulting model from experiment 4 is tested on all price groups.

All experiments using LSTM models are run three times due to their stochastic nature, only the averages are presented in chapter 5. Ideally we would have more than three runs

per model and do experiments 2,3, and 4 on all price groups, but time limits our testing capacity.

# Chapter 5

# Results and Discussion

## 5.1 Results

The complete results are presented in Appendix A. Except for the final model, we will only be presenting the forecasts for $t+1$ in this section. This is because the analytic gain is small compared to the disadvantage to presenting similar results three times over.

### 5.1.1 Experiment 1: Price Groups

A comparison of how well the different price groups are forecasted with a persistence forecast, SARIMA model, and a simple LSTM model trained on the same price group it forecasts. The results for $t+1$, $t+2$, and $t+3$ are presented in Tables A.1,A.2, and A.3 consecutively and the most important results for $t+1$ are presented in Figures 5.1 and 5.2.

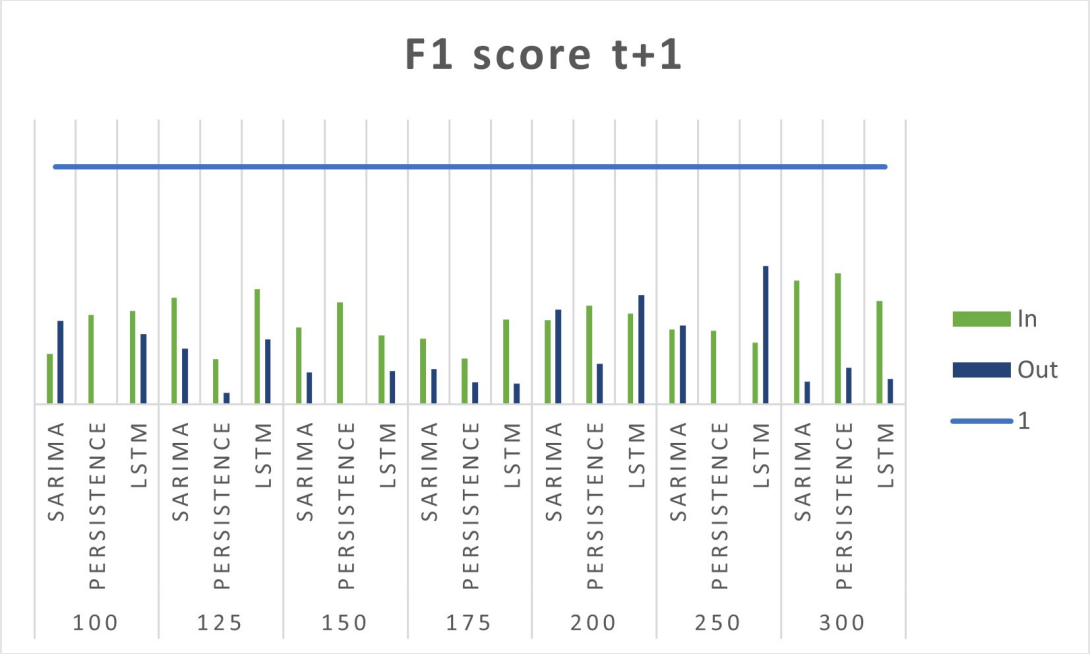
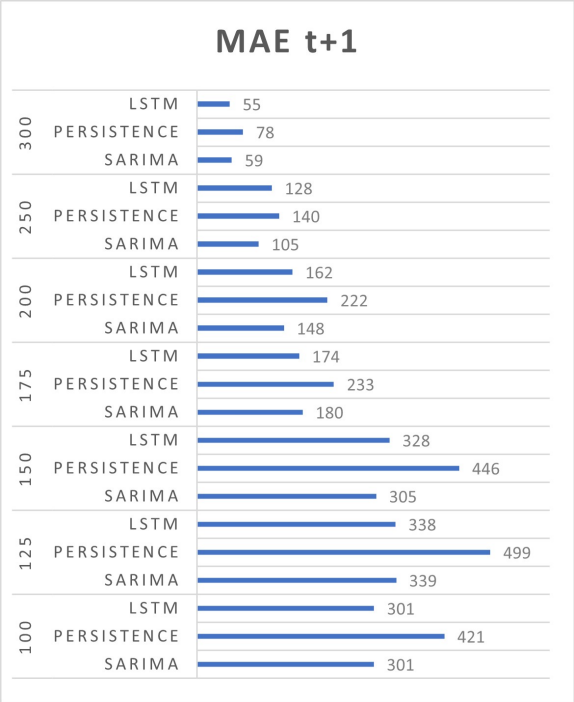
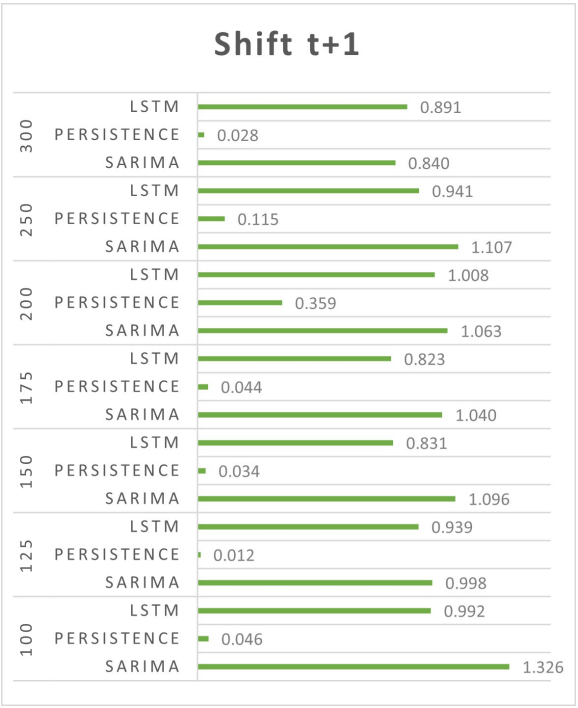


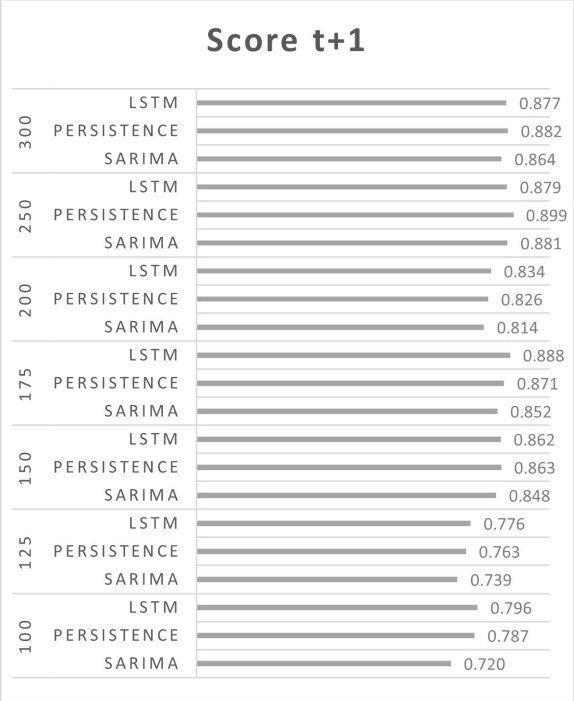

**Figure 5.1:** A comparison of the predictability of which products will enter or leave *basisutvalget* for different price groups of red wine at month $t+1$. No additional features are used in any of the models and the LSTM is only trained on the relevant price group. An F1 score of 1 is best.



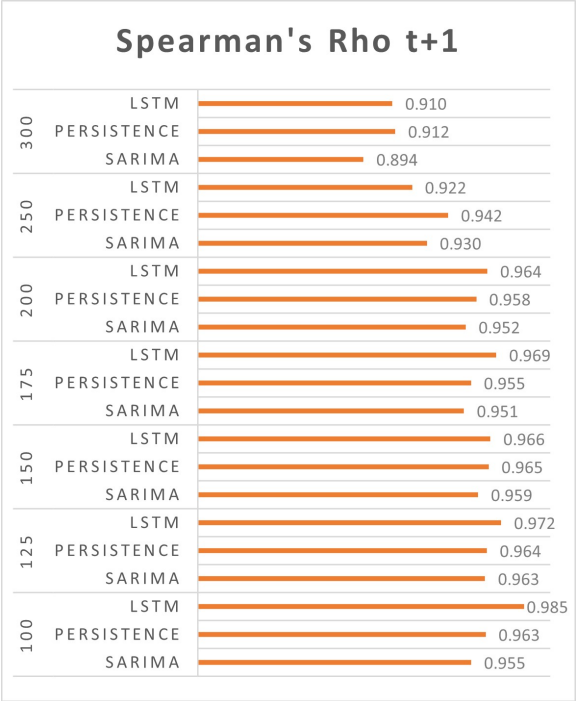

**(a)** Mean Absolute Error of forecast of *net sales*. Low MAE is best.




**(b)** Relative shifts in ranks produced from forecasts of *net sales* compared to actual shifts. Shift=1 is best.




**(c)** Measure of disorder between forecasted and actual ranks, l=10. *Score*=1 is best.




**(d)** Measure of rank correlation between forecasted and actual ranks. $\rho = 1$ is best.

**Figure 5.2:** A comparison of the predictability of the different price groups of red wine at month $t + 1$. No additional features are used in any of the models and the LSTM is only trained on the relevant price group.

### 5.1.2 Experiment 2: Features

Various features are tested on both the SARIMA and LSTM models for price group 175. The complete results are presented in Table A.4 and the most important results are presented in Figures 5.3 and 5.4.

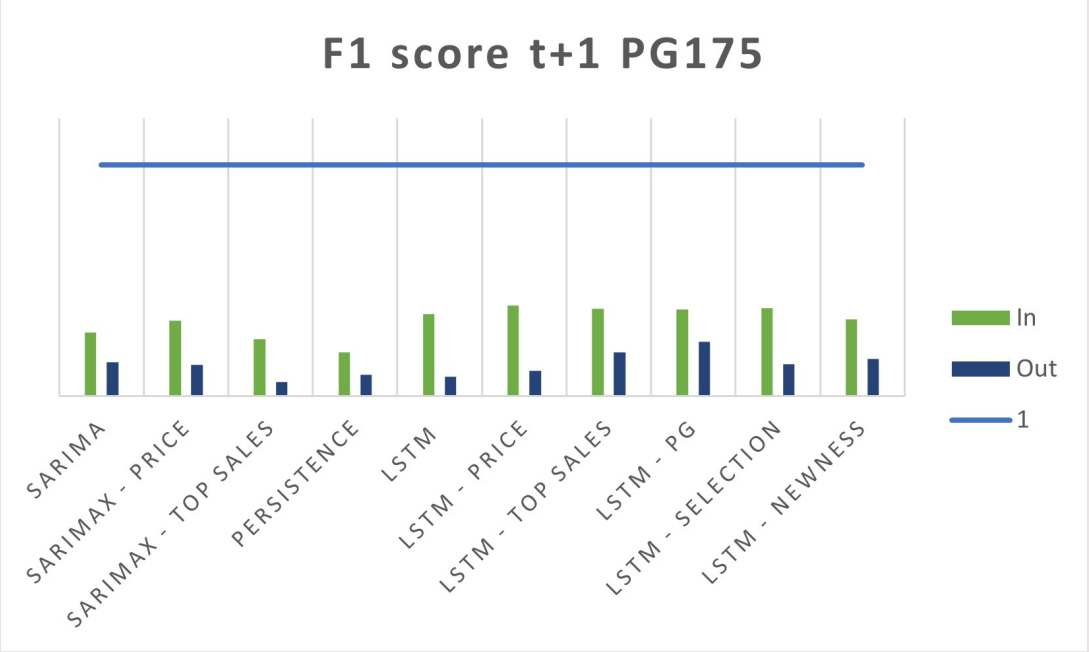
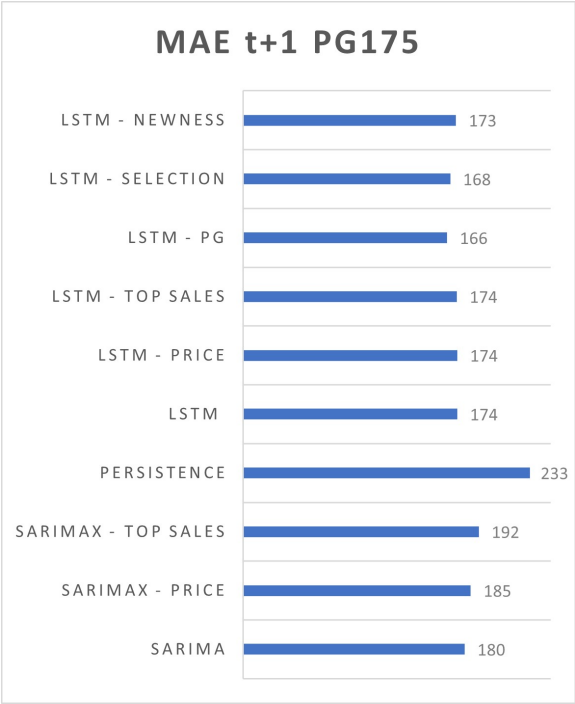
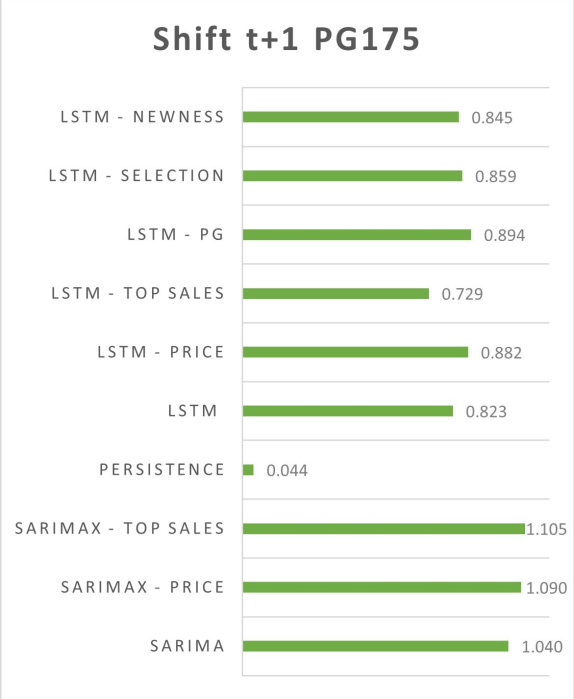


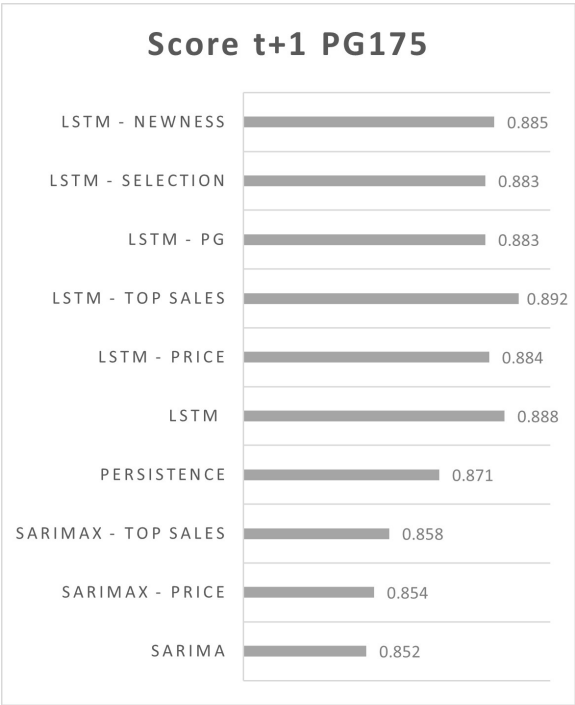

**Figure 5.3:** A comparison of features for the SARIMAX and LSTM models for price group 175 (175-199.99 NOK) at month $t+1$. The LSTM is only trained on price group price group 175. The F1 score for *In* is for the products classified as entering *basisutvalget* and *Out* is for the products classified as leaving *basisutvalget*. An F1 score of 1 is best.



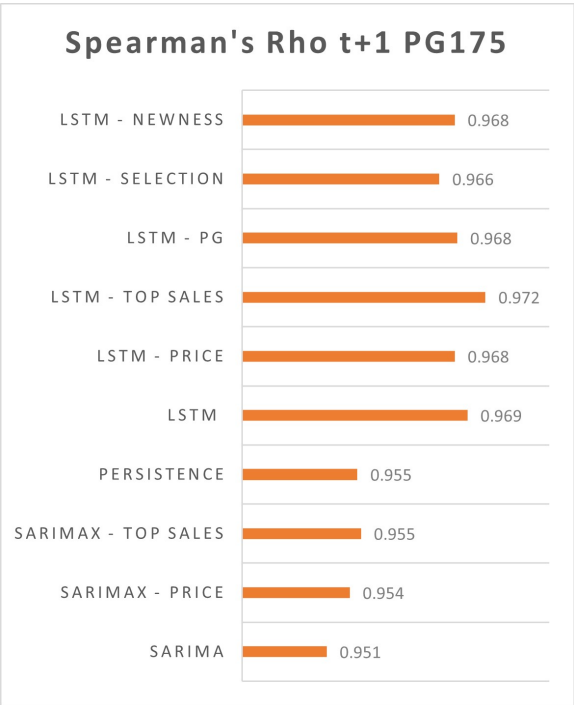

**(a)** Mean Absolute Error of forecast of *net sales*. Low MAE is best.




**(b)** Relative shifts in ranks produced from forecasts of *net sales* compared to actual shifts. Shift=1 is best.




**(c)** Measure of disorder between forecasted and actual ranks, l=10. *Score*=1 is best.




**(d)** Measure of rank correlation between forecasted and actual ranks. $\rho = 1$ is best.

**Figure 5.4:** A comparison of features for the SARIMAX and LSTM models for price group 175 (175-199.99 NOK). The LSTM is only trained on price group price group 175.

### 5.1.3 Experiment 3: Input Size

Testing various combinations of consecutive price groups to train the LSTM model on. The complete results are presented in Table A.5 and the most important results are presented in Figures 5.5 and 5.6. Be aware that the numbers are price groups not prices, therefore 100-300 is 100-399.99 NOK, 125-250 is 125-299.99 NOK, 150-200 is 150-249.99 NOK, 100-175 is 100-199.99 NOK and 175-300 is 175-399.99 NOK.

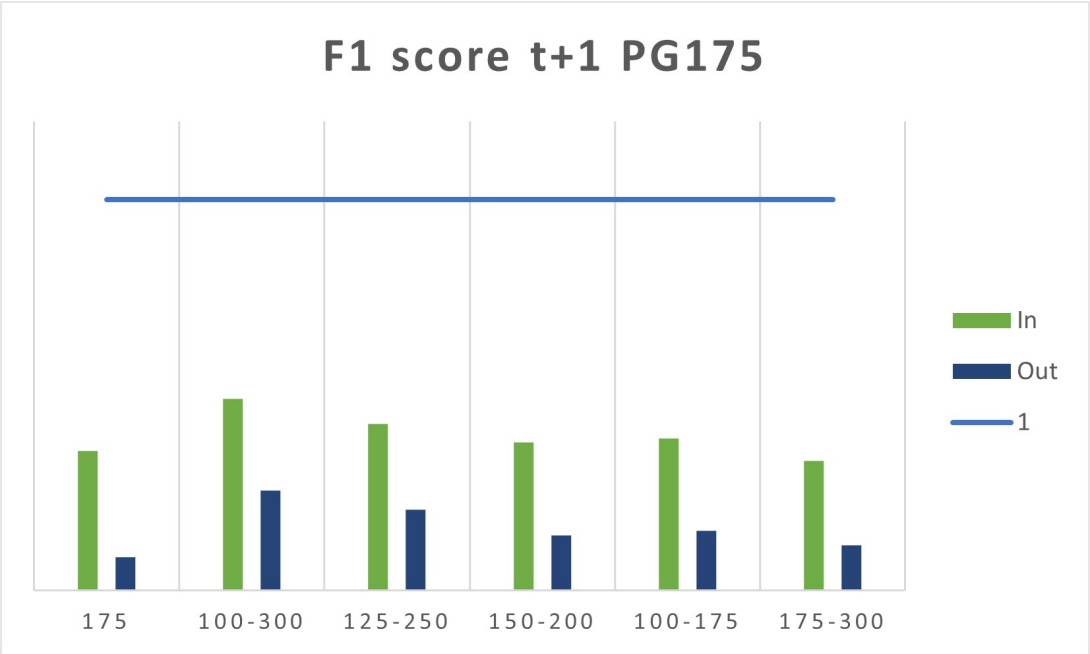
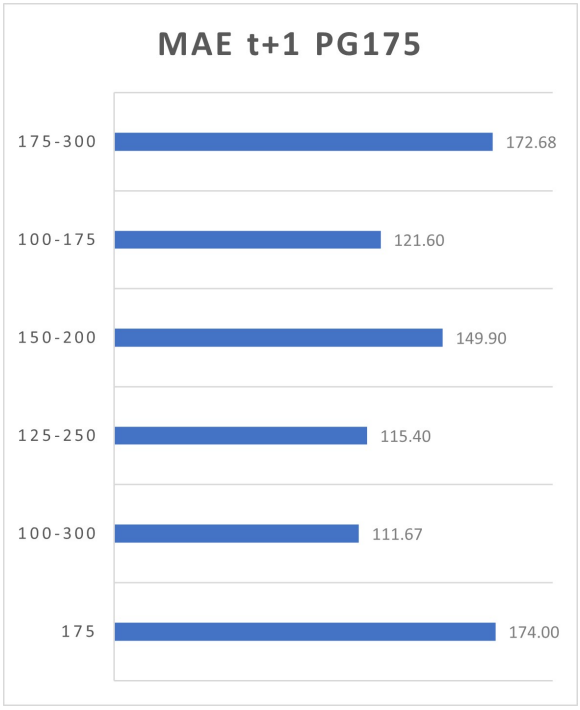
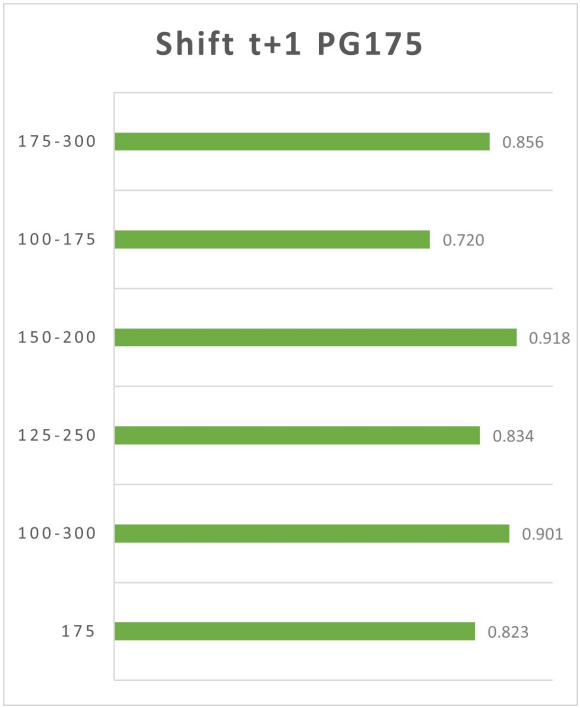


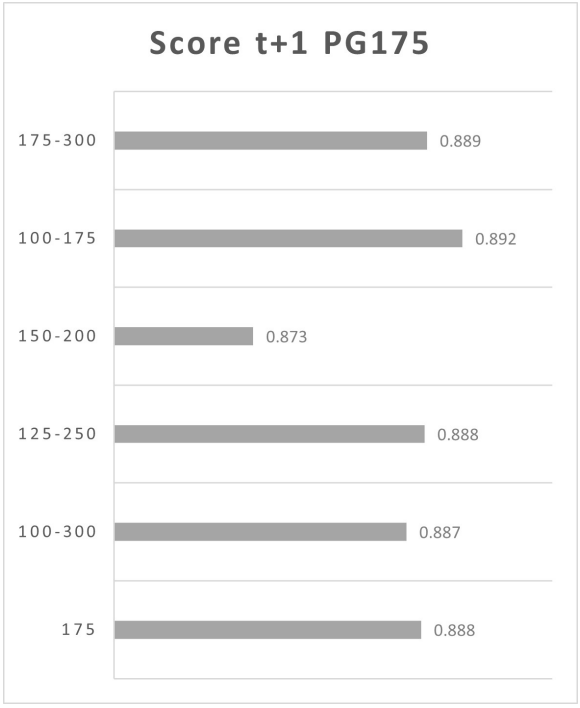

**Figure 5.5:** LSTM models with no features. Comparing combinations of price groups to use for training and validation sets for price group 175 (175-199.99 NOK) at month $t+1$. The F1 score for *In* is for the products classified as entering *basisutvalget* and *Out* is for the products classified as leaving *basisutvalget*. An F1 score of 1 is best.



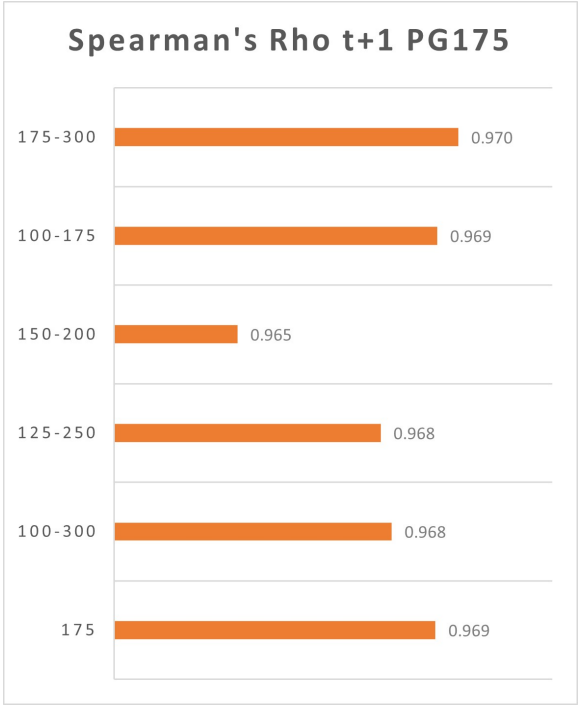

**(a)** Mean Absolute Error of forecast of *net sales*. Low MAE is best.




**(b)** Relative shifts in ranks produced from forecasts of *net sales* compared to actual shifts. Shift=1 is best.




**(c)** Measure of disorder between forecasted and actual ranks, l=10. *Score*=1 is best.




**(d)** Measure of rank correlation between forecasted and actual ranks. $\rho = 1$ is best.

**Figure 5.6:** LSTM models with no features. Comparing combinations of price groups to use for training and validation sets for price group 175 (175-199.99 NOK) at month $t + 1$.

### 5.1.4 Experiment 4: Optimizing Model

Experiment 2 shows that the features only slightly impact the performance of the models. *Top sales* scores slightly better than the other features on *score* and $\rho$. Studying Table A.4, we also observe that this feature performs slightly better than the other features for some of the performance measures for times $t+2$ and $t+3$ as well. This feature is dependent on which price group is tested, as it contains *net sales* of the best ranked product in that price group.

The *PG* feature does not show any improved performance in Figures 5.1 or 5.4. This is not surprising, considering it is only trained on one price group, PG175, and the feature does not vary from sample to sample. As this feature is more interesting when trained on multiple price groups, we will also look further at this feature. By studying these two features, we can compare the benefits of looking at a feature specialized on a single price group vs a feature specialized for multiple price groups.

Figures 5.5 and 5.6 show that training on all price groups give the highest F1 score and the lowest MAE. We will therefore try to optimize the model using all price groups with combinations of the features discussed above, all features and no features. The complete results are presented in Table A.6 and the most important results are presented in Figures 5.7 and 5.8.

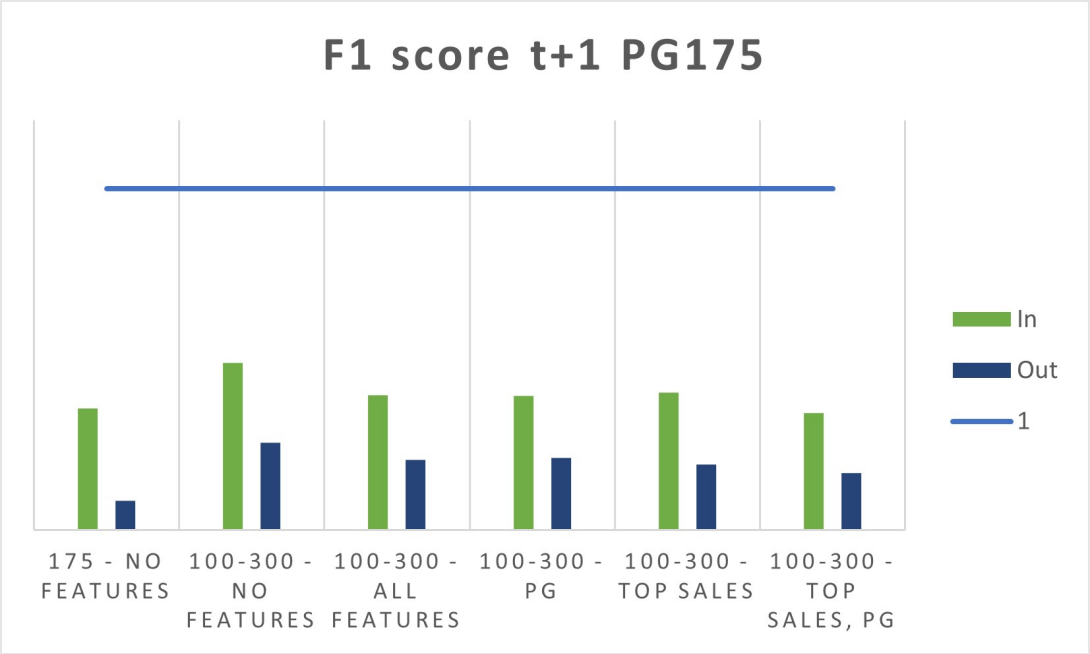
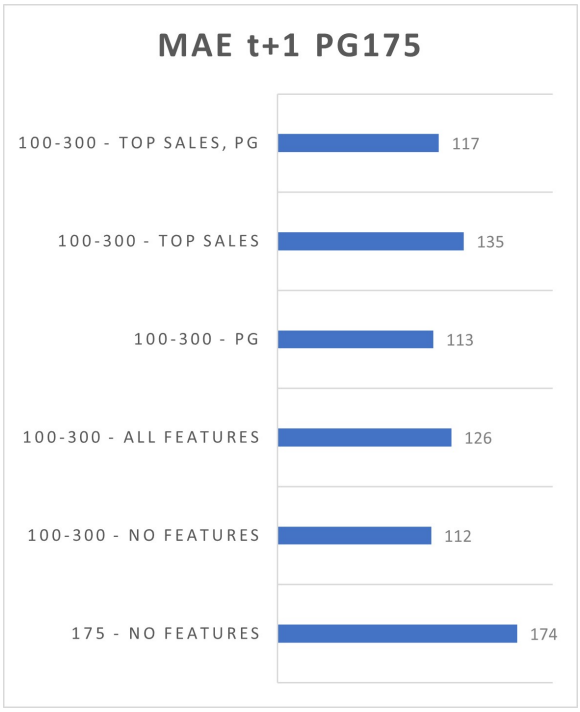
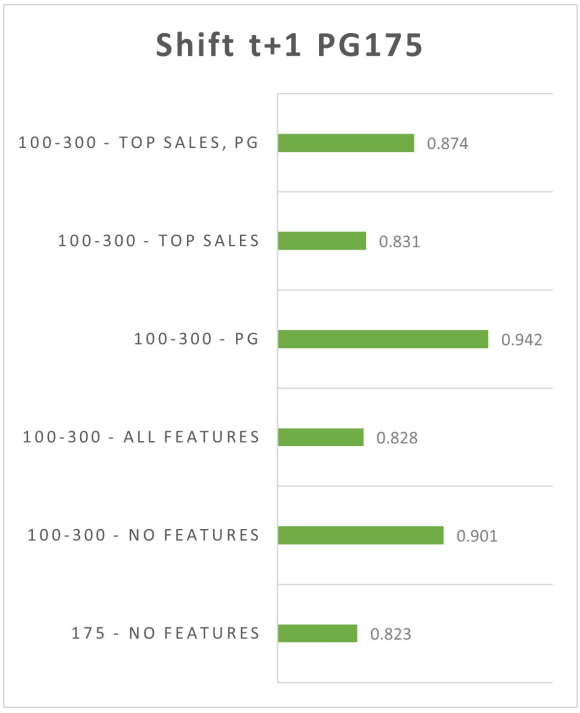


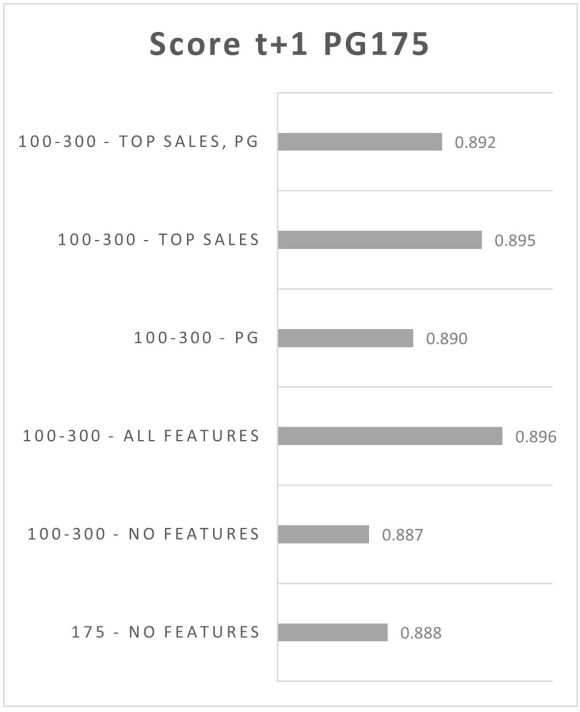

**Figure 5.7:** LSTM models with no features. Comparing combinations of price groups to use for training and validation sets for price group 175 (175-199.99 NOK) at month $t+1$. The F1 score for *In* is for the products classified as entering *basisutvalget* and *Out* is for the products classified as leaving *basisutvalget*. An F1 score of 1 is best. må oppdateres



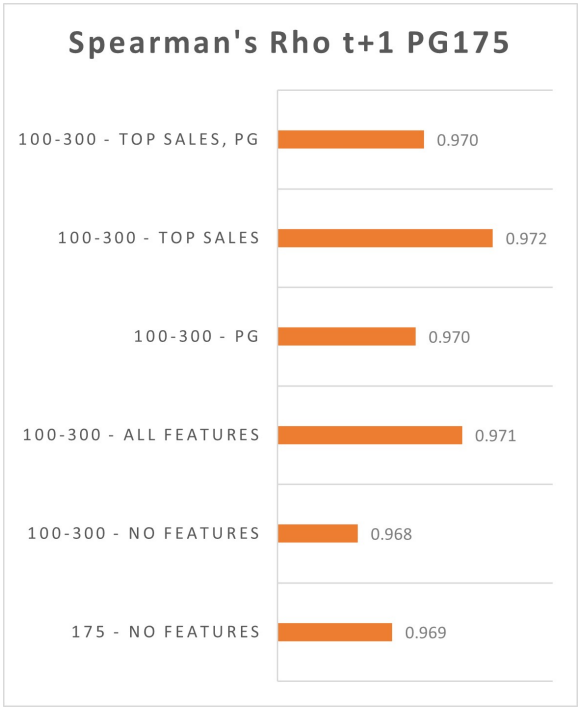

**(a)** Mean Absolute Error of forecast of *net sales*. Low MAE is best.




**(b)** Relative shifts in ranks produced from forecasts of *net sales* compared to actual shifts. Shift=1 is best.




**(c)** Measure of disorder between forecasted and actual ranks, l=10. *score*=1 is best.




**(d)** Measure of rank correlation between forecasted and actual ranks. $\rho = 1$ is best.

**Figure 5.8:** LSTM models with no features. Comparing combinations of price groups to use for training and validation sets for price group 175 (175-199.99 NOK) at month $t + 1$. må oppdateres

### 5.1.5 Final Model

Experiment 4 shows that the model trained on all price groups with no features performs slightly better with regards to the F1 score and MAE. The F1 score can be considered the most important performance measure since our main research question is a question about classification. Using the model without features can help avoid overfitting, and a single model can be fitted for use on all of the price groups. This model is therefore chosen and tested on all price groups. The complete results for the final LSTM model are presented in Table A.7 and the most important results are presented in Table 5.1. Tables 5.2 and 5.3 present the results of the SARIMA and persistence forecasts for comparison.

| PG | Time | MAE | Score | $\rho$ | Shift | $PRC_{in}$ | $PRC_{out}$ | $RCL_{in}$ | $RCL_{out}$ |
|---|---|---|---|---|---|---|---|---|---|
| | t+1 | 298.08 | 0.798 | 0.965 | 1.021 | 0.263 | 0.296 | 0.548 | 0.280 |
| 100 | t+2 | 411.88 | 0.695 | 0.927 | 0.890 | 0.324 | 0.371 | 0.430 | 0.262 |
| | t+3 | 615.68 | 0.612 | 0.886 | 0.833 | 0.292 | 0.334 | 0.333 | 0.211 |
| | t+1 | 241.16 | 0.787 | 0.975 | 0.938 | 0.519 | 0.228 | 0.465 | 0.215 |
| 125 | t+2 | 355.05 | 0.689 | 0.945 | 0.775 | 0.684 | 0.352 | 0.565 | 0.340 |
| | t+3 | 512.02 | 0.600 | 0.909 | 0.711 | 0.645 | 0.308 | 0.492 | 0.281 |
| | t+1 | 204.69 | 0.878 | 0.969 | 0.799 | 0.250 | 0.143 | 0.410 | 0.143 |
| 150 | t+2 | 276.48 | 0.805 | 0.939 | 0.738 | 0.329 | 0.192 | 0.476 | 0.203 |
| | t+3 | 410.07 | 0.742 | 0.905 | 0.716 | 0.317 | 0.220 | 0.452 | 0.253 |
| | t+1 | 111.67 | 0.887 | 0.968 | 0.901 | 0.357 | 0.197 | 0.780 | 0.367 |
| 175 | t+2 | 161.00 | 0.832 | 0.939 | 0.807 | 0.448 | 0.253 | 0.630 | 0.303 |
| | t+3 | 242.00 | 0.770 | 0.892 | 0.774 | 0.457 | 0.232 | 0.531 | 0.275 |
| | t+1 | 65.85 | 0.901 | 0.943 | 0.922 | 0.230 | 0.556 | 0.444 | 0.148 |
| 200 | t+2 | 109.26 | 0.839 | 0.881 | 0.871 | 0.201 | 0.581 | 0.375 | 0.216 |
| | t+3 | 150.99 | 0.788 | 0.842 | 0.778 | 0.196 | 0.667 | 0.300 | 0.160 |
| | t+1 | 127.55 | 0.879 | 0.922 | 0.941 | 0.240 | 0.668 | 0.284 | 0.518 |
| 250 | t+2 | 167.13 | 0.822 | 0.857 | 0.937 | 0.274 | 0.633 | 0.444 | 0.190 |
| | t+3 | 221.92 | 0.767 | 0.807 | 0.863 | 0.348 | 0.283 | 0.394 | 0.130 |
| | t+1 | 57.16 | 0.884 | 0.919 | 0.849 | 0.421 | 0.116 | 0.394 | 0.143 |
| 300 | t+2 | 83.84 | 0.832 | 0.875 | 0.766 | 0.639 | 0.163 | 0.333 | 0.139 |
| | t+3 | 112.41 | 0.780 | 0.829 | 0.750 | 0.605 | 0.277 | 0.310 | 0.244 |

**Table 5.1:** Results from the final LSTM model for all price groups. Trained on all price groups with no additional features.

## 5.2 Discussion

### 5.2.1 Expectations

This research topic engages easily conversation and is mainly met by skepticism. This skepticism is often based on personal experience as consumers, feeling that their choices are made by random recommendations, random eye-catching bottles, or good experience with specific products that they once discovered by random. If their choices feel this random, why should the market as whole be systematic enough to forecast?

An argument as to why this would not work is that critics' reviews in books, newspapers and on TV have a large, sudden, and possibly short lived impact on the sales numbers which

| PG | Time | MAE | Score | $\rho$ | Shift | $PRC_{in}$ | $PRC_{out}$ | $RCL_{in}$ | $RCL_{out}$ |
|---|---|---|---|---|---|---|---|---|---|
| | t+1 | 300.93 | 0.720 | 0.955 | 1.326 | 0.226 | 0.343 | 0.500 | 0.343 |
| 100 | t+2 | 548.45 | 0.566 | 0.891 | 1.378 | 0.169 | 0.338 | 0.500 | 0.355 |
| | t+3 | 811.26 | 0.474 | 0.824 | 1.437 | 0.323 | 0.304 | 0.262 | 0.310 |
| | t+1 | 339.00 | 0.739 | 0.963 | 0.998 | 0.482 | 0.241 | 0.424 | 0.225 |
| 125 | t+2 | 605.00 | 0.609 | 0.911 | 0.960 | 0.458 | 0.326 | 0.373 | 0.313 |
| | t+3 | 882.00 | 0.509 | 0.851 | 0.971 | 0.373 | 0.250 | 0.305 | 0.254 |
| | t+1 | 305.00 | 0.848 | 0.959 | 1.096 | 0.250 | 0.133 | 0.462 | 0.143 |
| 150 | t+2 | 492.00 | 0.757 | 0.920 | 0.916 | 0.323 | 0.227 | 0.476 | 0.217 |
| | t+3 | 703.00 | 0.671 | 0.864 | 0.961 | 0.359 | 0.276 | 0.500 | 0.276 |
| | t+1 | 180.00 | 0.852 | 0.951 | 1.040 | 0.200 | 0.105 | 0.444 | 0.222 |
| 175 | t+2 | 322.00 | 0.765 | 0.887 | 1.050 | 0.259 | 0.125 | 0.389 | 0.188 |
| | t+3 | 463.00 | 0.698 | 0.827 | 1.070 | 0.270 | 0.091 | 0.370 | 0.136 |
| | t+1 | 148.00 | 0.814 | 0.952 | 1.063 | 0.265 | 0.563 | 0.555 | 0.313 |
| 200 | t+2 | 246.00 | 0.728 | 0.897 | 0.915 | 0.351 | 0.438 | 0.565 | 0.259 |
| | t+3 | 341.00 | 0.661 | 0.840 | 0.873 | 0.463 | 0.333 | 0.594 | 0.189 |
| | t+1 | 105.00 | 0.881 | 0.930 | 1.107 | 0.231 | 0.667 | 0.500 | 0.222 |
| 250 | t+2 | 183.00 | 0.813 | 0.851 | 1.000 | 0.235 | 0.667 | 0.500 | 0.250 |
| | t+3 | 255.00 | 0.753 | 0.787 | 0.968 | 0.217 | 0.636 | 0.454 | 0.304 |
| | t+1 | 59.00 | 0.864 | 0.894 | 0.840 | 0.438 | 0.071 | 0.636 | 0.143 |
| 300 | t+2 | 100.00 | 0.794 | 0.824 | 0.882 | 0.478 | 0.091 | 0.524 | 0.167 |
| | t+3 | 138.00 | 0.734 | 0.750 | 0.851 | 0.481 | 0.038 | 0.464 | 0.067 |

**Table 5.2:** Results from the SARIMA ranks for all price groups.

can not be forecasted. These reviews might not be possible to forecast beforehand, but the aftereffects of such spikes could have long-term forecastable tendencies that an LSTM model could pick up. The actual influence from reviews could also be lower than expected. Many recommended products are only available to a certain quantity or have to be ordered, which could put off customers who are interested in getting their products straight away. These customers could end up buying other products instead, possibly products recommended as a good alternative with similar qualities by the employees at Vinmonopolet.

Just as wine trends change over time, food trends change as well. As wines are often recommended on the basis of which meal they are planned to be served with, we can surmise that food trends will have some influence on the wine trends. Employees are trained to know which wines go well with which dishes based on characters such as acidity, boldness, dryness, and tastes such as oak, chocolate or leather. Some of these characteristics are available in *Products*, but this information was too scarce to be used effectively by the LSTM model. Whether they would have improved the accuracy of the model or just introduced more noise is unsure. Some might say that wines are too individual in taste and that preferences vary too much from person to person, from setting to setting, and from meal to meal, for them to be viewed in an objective manner as this thesis does. There are wines priced at 200 NOK which connoisseurs would value at 600 NOK, and wines priced at 600 NOK which an average person would value at 200 NOK. There are also white wines which could be mistaken for red wines, which says a lot about how much wine can vary in taste and how difficult they are to describe.

The high sales numbers for the most popular wines and the large number of products available imply that while people are creatures of habit, they are also willing to explore a

| PG | Time | MAE | Score | $\rho$ | Shift | $PRC_{in}$ | $PRC_{out}$ | $RCL_{in}$ | $RCL_{out}$ |
|---|---|---|---|---|---|---|---|---|---|
| | t+1 | 420.78 | 0.787 | 0.963 | 0.046 | 0.375 | 0.000 | 0.375 | 0.000 |
| 100 | t+2 | 778.96 | 0.669 | 0.918 | 0.028 | 0.250 | 0.063 | 0.429 | 0.000 |
| | t+3 | 1096.05 | 0.592 | 0.876 | 0.021 | 0.194 | 0.000 | 0.095 | 0.014 |
| | t+1 | 499.00 | 0.763 | 0.964 | 0.012 | 0.444 | 0.111 | 0.121 | 0.033 |
| 125 | t+2 | 923.00 | 0.630 | 0.920 | 0.007 | 0.444 | 0.286 | 0.067 | 0.042 |
| | t+3 | 1304.00 | 0.538 | 0.874 | 0.005 | 0.555 | 0.286 | 0.061 | 0.032 |
| | t+1 | 446.00 | 0.863 | 0.965 | 0.034 | 0.400 | 0.000 | 0.461 | 0.000 |
| 150 | t+2 | 824.00 | 0.775 | 0.927 | 0.020 | 0.533 | 0.000 | 0.381 | 0.000 |
| | t+3 | 1151.00 | 0.712 | 0.890 | 0.015 | 0.500 | 0.000 | 0.250 | 0.000 |
| | t+1 | 233.00 | 0.871 | 0.955 | 0.044 | 0.167 | 0.083 | 0.222 | 0.111 |
| 175 | t+2 | 425.00 | 0.784 | 0.907 | 0.026 | 0.333 | 0.000 | 0.222 | 0.000 |
| | t+3 | 598.00 | 0.725 | 0.858 | 0.020 | 0.417 | 0.000 | 0.185 | 0.000 |
| | t+1 | 222.00 | 0.826 | 0.958 | 0.359 | 0.313 | 0.250 | 0.625 | 0.125 |
| 200 | t+2 | 402.00 | 0.737 | 0.908 | 0.210 | 0.250 | 0.200 | 0.304 | 0.074 |
| | t+3 | 564.00 | 0.666 | 0.850 | 0.160 | 0.259 | 0.250 | 0.219 | 0.081 |
| | t+1 | 140.00 | 0.899 | 0.942 | 0.115 | 0.285 | 0.000 | 0.333 | 0.000 |
| 250 | t+2 | 256.00 | 0.827 | 0.876 | 0.064 | 0.333 | 0.000 | 0.250 | 0.000 |
| | t+3 | 356.00 | 0.770 | 0.811 | 0.049 | 0.333 | 0.000 | 0.182 | 0.000 |
| | t+1 | 78.00 | 0.882 | 0.912 | 0.028 | 0.714 | 0.167 | 0.455 | 0.143 |
| 300 | t+2 | 141.00 | 0.817 | 0.849 | 0.018 | 0.857 | 0.143 | 0.286 | 0.083 |
| | t+3 | 194.00 | 0.759 | 0.786 | 0.014 | 0.833 | 0.000 | 0.179 | 0.000 |

**Table 5.3:** Results from the persistence forecast ranks for all price groups.

wide variety of new wine. This willingness to test different wines could partly be an effect of the difficulty to describe wine and define personal preferences, but could also be an effect of curiosity and availability.

Some products are more recognizable than others. Most employees and many consumers will know which product you refer to if you mention "pinnevinen", meaning "the stick wine", which is characterized by a piece of grape vine hung on the bottle, or the bottle with a hand-print indentation in the glass. Wine labels could also have motifs that seem fitting for special occasions or holidays. Characterizations like these highly influence the choices of consumers and is a feature that this project probably would have benefited from. Vinmonopolet has pictures of most products available on their website. Collecting these and using a Convolutional Neural Network could be an interesting development within this research field.

Another factor we did not take into consideration is vintage. Wineries, districts, and even countries can have good and bad seasons, which could affect both taste and quantities produced. Weather and climate change could also affect prices, which could move products between price groups or make the products undesirable for the importers whom therefore stop importing them to Norway.

An important aspect with regards to external influence is that the Norwegian law forbids commercials for alcoholic drinks. This puts this market in a distinctive position compared to other products and compared to other countries. Lack of advertisement and product placement significantly reduce the influence towards specific products, leveling the playing field and removing the variance in sales trends that would occur if products went on and off sale and were sporadically advertised.

### 5.2.2 Price groups

To answer the research question about which price groups are easier to analyze, we look at Figure 5.2a from experiment 1 and observe that the MAE of the forecasted time series are lower for the more expensive products. As shown in Table 2.3, the sales numbers decrease as prices increase. The lower MAE is presumably caused by the small sales, not because the trends are more predictable. On the contrary, one would expect more expensive wines to be especially dependent on critics reviews, quality of the harvest, batch orders, and other variables we can not take into account in our forecasts. The results from the forecasted ranking lists shown in Figures 5.1 and 5.2 show little preference towards any price group. PG100, PG200, and PG250 have higher $F1_{in}$ scores than the other price groups, and PG125 and PG300 have slightly higher $F1_{out}$ scores than the other price groups.

### 5.2.3 Features

In experiment 2 we study the impact of adding features to the SARIMA and LSTM models. Looking at the SARIMA model first, we observe in Figure 5.3 that *price* slightly improves the F1 score in, while *top sales* has lower F1 scores than without a feature. In Figure 5.4a we observe that the model with no features gives a slightly smaller MAE and in Figures 5.4c and 5.4d *top sales* gives slightly better ordered ranking lists. If there are any benefits to using a SARIMAX model instead of a SARIMA model, they are too small to observe. Going for the motto "less is more", we would suggest not to use these exogenous variables for these time series.

Figure 5.3 shows for the LSTM models that the features have most impact on $F1_{out}$ which is especially high for *price* and *top sales*, $F1_{in}$ is relatively stable for all features. Figure 5.4 shows that all of the models have relatively stable performance measures, only *top sales* stands a bit out, with lower *shift* and higher *score* and $\rho$. *Top sales* and *price* are further examined in experiment 4, where the models are trained on all price groups. Looking at Figure 5.7, we observe that the model with no features performs better than all of the other combinations of features. It is not clear whether this is due to overfitting or if the features are not correlated enough to *net sales*. Since the model with all features has a relatively high F1 score and the highest *score* in Figure 5.8c, overfitting is less likely to be the cause.

That *newness* did not improve the LSTM model can imply that not all new products behave in a similar fashion. It can also imply that they do not behave differently enough to products that have been on the market for a while for the feature to make difference. If there had been a clear difference between old and new products, this feature would have been interesting to use for analysis when adding new products to the market.

*Selection* was feature that was expected to have an impact on the results after observing the clear differences when changing sales category in Figure 2.8, products in *basisutvalget* or *testutvalget* have high sales numbers and the others have small sales numbers. By only feeding the status at time $t$ instead of for times $[t-19, t-18, \ldots, t]$, the feature lost the possibility to observe recent changes in sales category, which could have been highly relevant, but it could also be that the increase or decrease in *net sales* is instantaneous enough when changing sales category that the feature still would be superfluous.

*Price group* as a feature would seem beneficial when training on multiple price groups, as the consumer profiles for the different price groups are expected to have various purchasing behaviors. As we observe in section 2.1 and Figure 2.5, general inflation in prices makes what once would be considered a medium expensive price group, a cheap price group today.

By training on old data and testing on new data, the purchasing trends for each price group could have changed, giving a possible downside to this feature.

*Top sales* was expected to work well when training on a single price group and when training on multiple price groups, as the most popular wine in a product group is assumed to be representative of purchasing behavior within that price group. The lack of improvement could be due to many reasons. Sales trends for the most popular products could be different from the less popular products, which are the products most relevant for the F1 score and the thesis. Most sales trends could be so similar that *top sales* is superfluous. The feature could also lead to overfitting.

*Price* was expected to perform well for the same reasons as *price group*, but with the benefit of including historic as well as current data and comparing the prices of products within the same price group. Products with prices near the lower limit of the price range could be expected to sell more than the more expensive products in that price group, as Table 2.3 shows that less expensive products are more popular.

That none of the features manage to improve the model, and that the performance measures disagree on which models perform slightly better than the others, could mean that there are too many unknown factors influencing the sales numbers for us to forecast them with any success. Some features that would have been interesting to test but were rejected due to scarcity or too many classes were country, district, taste, smell, and quality. Features that would have been interesting to add but were not part of the data, are critics reviews, articles related to wines available online, and bottle appearance.

### 5.2.4 Input Size

In experiment 3 we try to answer whether training on a single price group or multiple price groups gives the best performance. If training on a single product group performed best, we could have assumed that the difference in consumer behavior between the price groups outweighed the positive effect on increasing sample size for a machine learning model. As seen in Figure 5.5, using multiple price groups for training clearly improves the LSTM model. Comparing the results from Tables A.1, A.2, A.2, and 5.1, and taking the MAE average over all price groups, we find that the average is lowered by 25.5 %, 25.7 %, and 23.4 % for times $t+1$, $t+2$, and $t+3$ when using all price groups as input instead of a single price group. The more samples used for training, the better the F1 score becomes. Using price groups 100-175 show better results than using price group 175-300. This could be caused by the larger number of samples in the cheaper price groups, that trends in the cheaper price groups are more similar to the trends in PG175, or a combination of these two reasons. Figure 5.6a confirms that using more samples improve the forecasts, and the other figures in Figure 5.6 show varying results. The results in experiment 3 are not surprising, with machine learning more data is often useful as long as the quality of the data is good. A possible method to obtain more data would be to train on more product groups, such as white wine, rosé wine, and sparkling wines. More data is not necessarily good, training on these products could have a negative impact on the model. These products could be more common in the summer half of the year, customers might be more or less attached to specific products, and each product group could have different fashions. Using product group as a feature could lower the negative effect of such differences, but an improved model is not guaranteed.

### 5.2.5 Final Models

#### Forecasts

Studying the results in Tables 5.1, 5.2, and 5.3, we find that for all price groups the LSTM model has an average MAE of 158, 224, and 324 for $t+1$, $t+2$, and $t+3$, the SARIMA model has an average MAE of 205, 357, and 513 for $t+1$, $t+2$, and $t+3$, and the persistence forecast has an average MAE of 291, 536, and 752 for $t+1$, $t+2$, and $t+3$. The LSTM model has the lowest MAE and it only increases with 105 % from $t+1$ to $t+3$ versus the 150 % and 158 % increase for the SARIMA and persistence forecasts consecutively. The higher MAE for the persistence forecast is expected for time series with such dramatic seasonal changes as shown in Figure 2.2. Since the values we are forecasting, *net sales*, are a rolling sum of six months, the time series are split into high season and low season, where high seasons are the six months that contain the spike in December and low seasons are the six months that don't. This is visible in Figure 2.6. These plateaus cause an effect where for 10 out of 12 months the persistence forecast of $t+1$ will be close to the real value, for 8 out of 12 months the persistence forecast of $t+2$ will be close to the real value, and for 6 out of 12 months the persistence forecast of $t+3$ will be close to the real value.

#### Ranking

For *score* and $\rho$ the differences between the three models are small. The LSTM ranking is on average best, while the persistence ranking is second best. This implies that the LSTM forecasts are better than assuming that no changes are made to the ranks. This could also imply that the SARIMA forecasts are worse than assuming no change, but this is disputed by some of the other performance measures.

The *shift* measure does not directly measure the accuracy of the forecasted ranks, but gives an impression of whether the forecasted ranks in general change more or less than the actual ranks change. The SARIMA ranks' *shift* is closer to one than than the LSTM ranks' lower *shift*, implying that the SARIMA forecasts are not as parallel as the LSTM forecasts. By parallel forecasts we mean time series that do not cross each other, the higher selling product does not sell less than the lower selling product for any month. *Shift* for the persistence ranks should in theory be zero, as it forecasts no change in sales from month to month. The deviation from zero is a product of the assumptions we made and the way the code works. The code is written so that the forecast assumes that the price is stable, therefore no products are forecasted to leave because of a change in price. When products actually change price groups, they get a new rank in the new price group, which is the rank that is compared to the forecasted "old" rank. This fault could be fixed by removing these products from the evaluation, but these changes impact all of the performance measures on the ranks. Removing these products would cause unrealistic good scores, as unexpected price changes are expected to occur when competing importers decide the sales prices of their products.

#### Classification

$\text{Precision}_{\text{in}}$ is the fraction of predictions that a product will enter *basisutvalget* that are correct. The final results show that the $\text{precision}_{\text{in}}$ is low for all models. The LSTM ranks have highest results, managing correct guesses over half of the times for PG125 and PG300. The SARIMA ranks have also better precision in these two price groups than the other price groups, but never manages a score above 0.5. The persistence ranks have a slightly higher

$\text{precision}_{\text{in}}$ than the SARIMA ranks, except for PG300 which has unusually high results. The $\text{precision}_{\text{in}}$ for the persistence ranks would have been zero if no changes were made to prices or ranking limit. Whenever the ranking limit is increased, products that have a ranking value just above the ranking limit would enter *basisutvalget*. Assuming that the ranking lists stayed somewhat stable during this period, this would be caught up by the persistence ranks. Looking at Figure 2.1, we see that the ranking limit is increased at various occasions for PG125, PG150, PG200, PG250, and PG300 throughout the test period August 2017 - September 2019. The ranking limit for PG175 stays the same and for PG100 only decreases. This confirms the theory that increasing the ranking limit accounts for part of the high precision, as PG100 and PG175 have the lowest $\text{precision}_{\text{in}}$. That the precision is not zero for these two price groups is accounted for by price changes. Whenever a product leaves the evaluated price group, the products with a higher ranking value have their ranking lowered by one and a new product enters *basisutvalget*. The LSTM ranks for PG100 and PG175 do not have the lowest $\text{precision}_{\text{in}}$, which shows us that these forecasts are better than assuming parallel trends. The $\text{precision}_{\text{in}}$ for the SARIMA ranks is lower for PG100, PG175, and PG250 (whose ranking limit increases the least), which could imply that these ranks stay more stable than for the LSTM, but this is contradicted by the higher *shift* measures.

$\text{Precision}_{\text{out}}$ is the fraction of predictions that a product will leave *basisutvalget* that are correct. Also $\text{precision}_{\text{out}}$ is low for all models. The LSTM ranks manage to correctly guess over half of the times for PG200 and PG250. The SARIMA ranks are slightly lower, only predicting correctly over half of the times for PG250. Similarly to $\text{precision}_{\text{in}}$, $\text{precision}_{\text{out}}$ is expected to be impacted by changing prices and ranking limits, but here the precision is closer to zero for the persistence ranks, and the highest scores do not coincide with the price groups where the ranking limit is lowered. Why there is no correlation between the ranking limit being lowered and the precision, is unclear. It could be because the ranking limits are assumed the same for $t+1$, $t+2$, and $t+3$ as $t$, but this should only delay this effect, not remove it.

The code for calculating rankings is set up in such a way that it only looks at products that are in its price group at time $t$. Products that are removed from the price group are evaluated until the time they disappear. Products that enter the price group at time $t+1$, $t+2$, or $t+3$ are not evaluated for the ranking lists at these times, only when they appear at time $t$ are they evaluated for times $t+1$, $t+2$, and $t+3$. This fault in the code means that new products are appearing in *basisutvalget* to push other products out of *basisutvalget* without being picked up in the forecasted ranking list, as is done when products disappear and $\text{precision}_{\text{in}}$ gets a higher value. This is one of the reasons why $\text{F1}_{\text{in}}$ is consistently higher than $\text{F1}_{\text{out}}$; this impacts all models, but is only clearly visible when studying the persistence model.

$\text{Recall}_{\text{in}}$ is the amount of cases where a product enters *basisutvalget* that are forecasted out of the actual amount of cases. Also here the scores are low for all models. The SARIMA ranks score slightly better with $\text{recall}_{\text{in}}$ than the LSTM ranks, managing to score over 0.5 on average for PG200 and PG300, while the LSTM ranks only manage this for PG175. This relatively high score for PG175 could be a consequence of the LSTM model being tuned on this price group.

The LSTM ranks score slightly better than the SARIMA ranks for $\text{recall}_{\text{out}}$, but neither of them manage an average of 0.5 for any of the price groups. Another way of saying this is that out of the products that leave *basisutvalget*, only a few of them are predicted by our models. This can probably in large part be due to new products entering the price group, pushing the products near the ranking limit out of *basisutvalget*. As discussed above, the

code does not evaluate products that will appear in the future, therefore cases like this will not be picked up, resulting in low $\text{recall}_{\text{out}}$ for all models. For the persistence ranks, the same argument goes for $\text{recall}_{\text{in}}$ and $\text{recall}_{\text{out}}$ as for $\text{precision}_{\text{in}}$ and $\text{precision}_{\text{out}}$.

### Stability in Rankings

To analyze why the classification performance measures are this low, we look at Table 4.2. Very few products are classified as *in* or *out*, telling us that the ranking lists are stable, even near the ranking limit. The symmetry between the occurrences of classes *in* and *out* is expected, when one product goes in, another must go out. The imbalance for PG100 can probably be explained by the steep decline in ranking limit for that price group in the test period, pushing products out of *basisutvalget*.

We expect the most popular products to have somewhat stable trends, as is shown in Figure 2.6, but new products or products with few sporadic sales would be hard to fit to a model. The lack of improvement in the models when adding features could mean that the stable products are already fitted, but that there is too much noise that can not be fitted by the features at hand without overfitting. When we only forecast three months ahead, it is not surprising that the changes in sales are too small to affect the ranking order of the products. Products would have to sell almost equal amounts at time $t$ and have a clear differences in trends for the models to forecast that the products would switch places.

The stability in the ranking lists tell us that the most important factors in deciding who enters or leaves *basisutvalget* are not shifts in sales, but rather how many products that are launched directly into *basisutvalget* and products changing price groups. These are factors that our simplified model did not take into consideration. In section 2.2 we observed that 41 products were launched in *basisutvalget* and 10 in *testutvalget* in 2018. To make space for these products in *basisutvalget*, the products with highest ranking value within *basisutvalget* are pushed over the ranking limit and moved to *bestillingsutvalget*. Expanding the model to take this into account, we should be able to better predict which products are at risk to leave *basisutvalget*. To do this, we could forecast the $n$ products that will be nearest the threshold, assuming that $n$ products are launched into *basisutvalget*. This would make it even harder to predict which products that will sell well enough to enter *basisutvalget*, as they would not only have to perform better than the worst ranking product in *basisutvalget*, they would have to perform better than $n+1$ products if $n$ products are launched directly into *basisutvalget*.

### Traditional vs Modern Methods

Overall the LSTM model performed better than the SARIMA model and the persistence model. The LSTM model could probably benefit from being tested and tuned for each price group separately, and the SARIMA model could probably be improved by having parameters selected uniquely for each time series, but this would probably not change the results dramatically. The persistence forecast works well as a benchmark to analyze results with, as it is predictable, easy to analyze, and uncovers flaws in the code. The SARIMA model works well as a "model to beat" and did in some cases surpass the LSTM model. Considering how evenly they often scored, for quick work where the products are analyzed separately, a SARIMA model would be recommended, as it requires far less data handling, data processing, and tuning than an LSTM.

The models all perform relatively well forecasting *net sales*, but these results are not reflected in the precision, recall or F1 score. Looking at Table 2.4, we observe that most products seldom switch between the sales categories, even over a 12 year time period. This

is confirmed in Table 4.2 where we observe that the classes *in* and *out* constitute a very small proportion of events in the test period, causing large variability in the results and making them difficult to forecast. It would seem that the ranks are too stable around the ranking limit for us to forecast this with any reliability. Changes in ranking limit and changes in prices that lead to a change in price group seem to account for a large part of the true positive cases of products entering *basisutvalget*, making the actual predictability even lower than they appear in the results.

### 5.2.6 Performance Measures

We have looked at performance measures that evaluate the forecasts, the ranking lists, the shifts in ranks, and the classification of products leaving and entering *basisutvalget*.

#### Forecasts

The MAE was a necessary metric for tuning the LSTM model. This metric gave interpretable results that were essential for deciding which models that could forecast *net sales* best.

#### Ranking

The *shift* measure provides information that is difficult to interpret and often disputes what the other performance measures present. As *shift* does not specify whether the relative ranking changes actually occur for the products that are supposed to shift rank, this performance measure seems to be more misguiding than beneficial.

Both the *score* and $\rho$ measure how ordered the forecasted ranking lists are to the true ranking lists. Only the products within the ranking limit are evaluated using these methods and none of the methods weigh any products higher than others. Normally rank evaluation weighs the lowest valued ranks the highest, as most ranking problems require finding the top ranked products. For our case, we are more interested in the worst products, the ones that are closest to the ranking limit and most at risk to be removed from *basisutvalget*. Despite being most interested in the worst ranked products, we do not train our models on this performance measure and chose therefore to weigh all products equally to get an impression of how well all forecasts perform. The small changes in these performance measures imply that a ranking measure that weighs the products near the ranking limit higher could be beneficial.

Comparing *score* and $\rho$ over the price groups for SARIMA ranking in Table 5.2 and persistence ranking in Table 5.3, we notice that the difference of *score* between the two models varies the most for the price groups with a higher ranking limit and the difference of $\rho$ between the two models varies the most for the price groups with few lower ranking limit. In Figures 5.2c and 5.2d we observe that the price groups with higher ranking limit have a higher $\rho$ and that the price groups with lower ranking limit have higher *score*. These performance measures are not used on all test samples, they are only used on the products with an actual ranking value within the ranking limit, which varies strongly between price groups, as shown in Figure 2.1. We discover, therefore, that both of these performance measures are dependent on sample length. As long as the test samples are the same, we can use $\rho$ and *score* to compare results, but these measures will shift slightly with the ranking limits and can not be used to compare one price group with another, as can be seen again in Tables 5.1, 5.2, and 5.3.

In Figures 5.4c and 5.4d, 5.6c and 5.6d, and 5.8c and 5.8d, we observe that there is a high correlation between $\rho$ and *score* when used on the same test set, therefore only one of the metrics are needed. *Score* has the advantage of simplicity and that the upper limit for distance penalty is set at need. Spearman's rank correlation coefficient has the advantage of being a known and tried method, making it more relatable and reliable. Looking at the LSTM results for the largest and smallest price groups, PG125 and PG300, at time $t + 1$ in Figure 5.2c and Figure 5.2d, we see that the absolute difference between the values for *score* are 0.101 and the absolute difference between the values for $\rho$ are 0.062. *Score* seems to be more impacted by sample size than $\rho$, putting it at a disadvantage. For further work $\rho$ would be the preferred performance measure for rank evaluation where all products are equally weighted.

### Classification

Precision and recall were the two most interesting metrics with regards to our main research topic. The metrics themselves were not the problem behind our varying results, rather the distribution of the classes. The products we evaluated were classified by when their ranking value became lower or equal to the ranking limit, we did not use the classes they were originally categorized with in the database, as these categories were decided by more factors than only ranking. Manually looking through the ranking lists, multiple cases emerged where products were classified as being in *basisutvalget* despite having a slightly too high ranking value. If-cases that would take into account additional rules could have been added, i.e. that new products in *basisutvalget* are guaranteed a spot for 12 months, but this would move the focus away from what we wished to discover, which is whether we could forecast trends accurately enough to predict whether a product would sell enough to rise in rank above the ranking limit, or sell too little and fall below the limit.

### Alternatives

For this thesis we constructed absolute ranking lists from the forecasts. This means that the ranking lists do not differentiate between when the products with consecutive ranking values perform similarly or are separated by thousands of liters. By finding the confidence intervals for the forecasts of each product, it would be possible to use the overlap between these intervals to assign a range of probable ranking values per product. As found in chapter 3, machine learning models are not specifically built for finding confidence intervals. There are methods that can be used to circumvent this, but considering the reasonable results of the SARIMA model, using the readily available confidence intervals of the SARIMA forecasts could be a better suggestion. By using this suggested method, we could make the results of a forecast more applicable than the method used in this project.

*Score* and $\rho$ could to our advantage be switched with a ranking measure that weighs the products near the ranking limit higher. A possible method to do this would be to flip around the ranking list and use a standard ranking measure which prioritizes the highest ranks.

# Chapter 6

# Conclusion

The monopolistic alcohol market and law against advertisement for alcoholic drinks should make sales data from this market ideal for forecasting. Without advertising and with a product taste that is hard for the average person to describe, consumers have to look somewhere for advice or innovation. Advice is often given by critics' reviews or employees at Vinmonopolet, and innovation can be found in fancy bottles or previous successful purchases. Consumers seem to be both creatures of habit and neophile, willing to try new products. If all of the consumers were creatures of habit, the sales data would be easy to forecast, but the research question's necessity would also disappear. As it is, new products are launched every other month, pushing the least sold products out of *basisutvalget* and off the shelves in the stores. This gives the consumers a steady stream of new products to try, and gives the importers a hard fight to keep their products in sale.

*Net sales*, the six-month rolling sum of sales in the top 60 stores, was forecasted for three months using LSTM, SARIMAX, and persistence forecasts. A significant spike in sales every December for most red wines gives *net sales* a high season and low season of 6 months each with relatively stable values. These plateaus cause the forecasts to get a relatively small MAE, even when using traditional methods. The final LSTM, SARIMA, and persistence one-month forecasts had an average MAE of 158, 205, and 291 consecutively. The MAE of the three-month forecasts for the same models increased with 105 %, 150 %, and 158 %, proving that the LSTM model works best for long-term forecasts and performs better than the traditional models overall.

No apparent improvement was made to the SARIMA model when price and shifted sales numbers of the top-ranked product were added as exogenous values. Price, price group, selection, newness, or sales numbers of the top-ranked product showed no clear improvement to the LSTM model when added as features. We can therefore conclude that none of the evaluated features manage to improve the forecasts, but we can not exclude possible improvements from other features.

The average MAE for all price groups was lowered by 25 % when training the LSTM model on all price groups instead of individual price groups, concluding that training the model on all of the price groups is best.

The MAE was lower for the more expensive price groups, which was expected as these price groups have lower sales numbers and, therefore, lower variation. The other performance measures implied that most price groups performed similarly, but these performance measures were dependent on sample size and could not be used for comparison between price groups. We can therefore not conclude whether any price groups were easier to forecast than the others, especially with regards to predicting which products will enter and

which products will leave *basisutvalget*.

A reasonable way to evaluate performance of the ranking would be to avoid absolute ranking and evaluate the forecasts with other methods. Ranking forecasted sales turned out to be a flawed method for recognizing products on the verge of changing product selection. Only significant differences in trends for products with similar sales numbers at time $t$ would lead to a change in ranking order when we are only forecasting three months. Too much information is lost by analyzing ranks instead of sales numbers, information about spacing between the products, and whether the trends are increasing or decreasing for the individual products. The performance measures we used weighed all ranks equally, a performance measure weighing the products with higher ranking value heavier could with benefit have been used, if sticking to absolute ranking. The best performance measures were MAE and F1 score, but the classes were not distributed evenly enough for the F1 score to give clear results.

To focus the analysis on the predictability of changes in ranks, not look at the complete market model straight away, we simplified the model to only look for products that would leave or enter *basisutvalget* on the basis of ranking value. The ranking values around the ranking limit proved to be quite stable, and the few changes in product selection that did occur were seldom forecasted. It seems as if the stable products are forecasted well, but there are too many unknown variables to forecast the unstable products with the current model.

The stability of the ranks imply that the most important factors influencing entry and exit of *basisutvalget* are the number of new products launched directly into *basisutvalget* and products shifting price range. Therefore, our model does not manage to forecast which products that will leave or enter *basisutvalget* with any precision as it is today. Expanding the model to include these factors, in addition to analyzing forecasts directly or with ranking intervals instead of absolute ranks, it could be possible to construct a model that could define risk levels for products near the ranking level. Producing a model that identifies products that are likely to enter *basisutvalget* would be difficult with the current performance of the forecasts.

## 6.1 Future Work

Using other methods to compare the forecasts could improve the usefulness and performance of the model. One example would be to find the confidence intervals for each forecast, which would make it possible to use the overlapping intervals to define sets of ranking intervals for each product; fewer ranks per product interval would mean increased confidence in the forecast. The forecasts could presumably be improved by using Natural Language Processing to analyze online reviews of the articles and taking in reviews from wine apps such as Vivino, which contain ratings and reviews from thousands of users. Bottle and label appearance could be used as a feature with image processing using Convolutional Neural Networks.

Expanding the model to take all contingencies into account, to simulate the real market, would make it more valuable to importers. The factors that would need to be taken into consideration are products entering at product launches, shifts in price ranges, and products that are protected for the first 12 months. Expanding the work to incorporate more products could improve performance and applicability for importers. Analysis on the effect of deliberate price changes could also be relevant in the business aspect, if lowering or raising

prices could shift products between price groups and impact sales numbers enough to enter *basisutvalget*.

# Bibliography


[1] FHI, *Alkoholomsetningen i norge*, https://www.fhi.no/nettpub/alkoholinorge/omsetning-og-bruk/alkoholomsetningen-i-norge/ (Last accessed: 20.05.2021), 2019.

[2] CAN, *Alkoholkonsumtionen i sverige 2018*, https://www.can.se/publikationer/alkoholkonsumtionen-i-sverige-2018/ (Last accessed: 21.05.2021), 2019.

[3] Danmarks Statistikk, *Statistikkbanken*, https://www.dst.dk/da/Statistik/emner/priser-og-forbrug/forbrug/forbrug-og-salg-af-alkohol-og-tobak (Last accessed: 21.05.2021).

[4] Ø. Horverak, 'The norwegian state alcohol monopoly. between teetotalism and moderation,' *Nordic Studies on Alcohol and Drugs*, vol. 18, no. 1, pp. 7–23, 2001. DOI: 10.1177/145507250101800119. eprint: https://doi.org/10.1177/145507250101800119. [Online]. Available: https://doi.org/10.1177/145507250101800119.

[5] J. Sverdrup, *Et statsmonopol blir til: vinmonopolet frem til 1932*. Fabritius, 1972.

[6] International Organisation of Vine and Wine, *2019 statistical report on world vitiviniculture*, https://www.oiv.int/public/medias/6782/oiv-2019-statistical-report-on-world-vitiviniculture.pdf, 2019.

[7] M. Hirche, J. Haensch and L. Lockshin, 'Comparing the day temperature and holiday effects on retail sales of alcoholic beverages – a time-series analysis,' 2021. [Online]. Available: https://doi.org/10.1108/IJWBR-07-2020-0035.

[8] U. Landazuri-Tveteraas, F. Asche and H.-M. Straume, 'Dynamics of buyer-seller relations in norwegian wine imports,' *Journal of Wine Economics*, vol. 16, no. 1, pp. 68–85, 2021. DOI: 10.1017/jwe.2020.37.

[9] B. D. Kristiansen, 'Sjokkert over hva norske vinimportører forteller om andre firmaer,' *Dagens næringsliv*, 20th May 2016, https://www.dn.no/smak/vin/geir-mosether/vinmonopolet/jan-tore-oskal/-sjokkert-over-hva-norske-vinimportorer-forteller-om-andre-firmaer/1-1-5646814(Last accessed: 25.05.2021).

[10] M.B. Lai, A. Cavicchi, K. Rickertsen, A.M. Corsi and L. Casini, 'Monopoly and wine: The norwegian case,' *British Food Journal*, vol. 115, no. 2, pp. 314–326, 2013. DOI: 10.1108/00070701311302267. [Online]. Available: https://doi.org/10.1108/00070701311302267.

[11] Ø. Horverak, 'Wine journalism—marketing or consumers' guide?' *Marketing Science*, vol. 28, no. 3, pp. 573–579, May 2009. DOI: 10.1287/mksc.1090.0489. [Online]. Available: https://ideas.repec.org/a/inm/ormksc/v28y2009i3p573-579.html.

[12] S. Makridakis, E. Spiliotis and V. Assimakopoulos, 'Statistical and machine learning forecasting methods: Concerns and ways forward,' *PloS one*, vol. 13, no. 3, e0194889, 2018.

[13] S. Makridakis, 'The forthcoming artificial intelligence (ai) revolution: Its impact on society and firms,' *Futures*, vol. 90, pp. 46–60, 2017.

[14] S. Siami-Namini, N. Tavakoli and A. Siami Namin, 'A comparison of arima and lstm in forecasting time series,' in *2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA)*, 2018, pp. 1394–1401. DOI: 10.1109/ICMLA.2018.00227.

[15] S. Makridakis, E. Spiliotis and V. Assimakopoulos, 'The m4 competition: 100,000 time series and 61 forecasting methods,' *International Journal of Forecasting*, vol. 36, no. 1, pp. 54–74, 2020, M4 Competition, ISSN: 0169-2070. DOI: https://doi.org/10.1016/j.ijforecast.2019.04.014. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0169207019301128.

[16] S. Smyl, 'A hybrid method of exponential smoothing and recurrent neural networks for time series forecasting,' *International Journal of Forecasting*, vol. 36, no. 1, pp. 75–85, 2020.

[17] A. Ramadhan and M. L. Khodra, 'Ranking prediction for time-series data using learning to rank (case study: Top mobile games prediction),' *2014 International Conference of Advanced Informatics: Concept, Theory and Application (ICAICTA)*, pp. 214–219, 2014.

[18] J. D. Cryer, *Time series analysis*. Springer, 1986, vol. 286.

[19] R. Jahn, *How neurons talk to each other*, https://www.mpg.de/10743509/how-neurons-talk-to-each-other (Last accessed: 21.06.2021), 2016.

[20] D. E. Rumelhart, G. E. Hinton and R. J. Williams, 'Learning internal representations by error propagation,' California Univ San Diego La Jolla Inst for Cognitive Science, Tech. Rep., 1985.

[21] L. Wessels and E. Barnard, 'Avoiding false local minima by proper initialization of connections,' *IEEE Transactions on Neural Networks*, vol. 3, no. 6, pp. 899–905, 1992. DOI: 10.1109/72.165592.

[22] L. Bottou, 'Large-scale machine learning with stochastic gradient descent,' in *Proceedings of COMPSTAT'2010*, Springer, 2010, pp. 177–186.

[23] T. Dietterich, 'Overfitting and undercomputing in machine learning,' *ACM computing surveys (CSUR)*, vol. 27, no. 3, pp. 326–327, 1995.

[24] D. P. Kingma and J. Ba, 'Adam: A method for stochastic optimization,' *arXiv preprint arXiv:1412.6980*, 2014.

[25] J. Duchi, E. Hazan and Y. Singer, 'Adaptive subgradient methods for online learning and stochastic optimization.,' *Journal of machine learning research*, vol. 12, no. 7, 2011.

[26] T. Tieleman and G. Hinton, 'Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude,' *COURSERA: Neural networks for machine learning*, vol. 4, no. 2, pp. 26–31, 2012.

[27] S. Hochreiter and J. Schmidhuber, 'Long short-term memory,' *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[28] I. Goodfellow, Y. Bengio and A. Courville, *Deep Learning*. MIT Press, 2016, ISBN: 9780262035613.

[29] J. Schmidhuber, 'Learning complex, extended sequences using the principle of history compression,' *Neural Computation*, vol. 4, no. 2, pp. 234–242, 1992.

# Appendix A

# Results

Results of the experiments described in chapter 4 and presented in chapter 5. Abbreviations and functions helpful for interpreting the results:

- PG - price group, intervals in NOK listed in section 2.1
- Model -
  - SARIMAX - Seasonal Autoregressive Integrated Moving Average with exogenous variable, Equation 3.29
  - Persistence - forecasting the same sales value for $t+1$,$t+2$, and $t+3$ as was given for $t$
  - LSTM - Long Short-Term Memory, described in subsection 3.4.2
- Features - additional input to LSTM or SARIMAX models, listed in section 4.3
- MAE - Mean Absolute Error, Equation 3.41
- Score - a new method of measuring disorder between two ranking lists, Equation 3.3, $l = 10$
- $\rho$ - Spearman's $\rho$, a measure of rank correlation, Equation 3.1
- Shift - relative shifts in ranks produced from forecasts compared to actual shifts, Equation 3.7
- PRC - precision, Equation 3.4
- RCL - recall, Equation 3.5
- F1 - $F_1$-score, Equation 3.6
- *in* - product entering *basisutvalget*
- *stay* - product staying in *basisutvalget* or *bestillingsutvalget*
- *out* - product leaving *basisutvalget*

| | | Month $t+1$ | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| PG | Model | MAE | Score | $\rho$ | Shift | $PRC_{in}$ | $PRC_{stay}$ | $PRC_{out}$ | $RCL_{in}$ | $RCL_{stay}$ | $RCL_{out}$ | $F1_{in}$ | $F1_{stay}$ | $F1_{out}$ |
| 100 | SARIMA | 300.93 | 0.720 | 0.955 | 1.326 | 0.226 | 0.990 | 0.355 | 0.200 | 0.978 | 0.347 | 0.212 | 0.984 | 0.351 |
| | Persistence | 420.78 | 0.787 | 0.963 | 0.046 | 0.375 | 0.986 | 0.000 | 0.375 | 0.972 | 0.000 | 0.375 | 0.979 | NaN |
| | LSTM | 301.33 | 0.796 | 0.985 | 0.992 | 0.293 | 0.993 | 0.267 | 0.593 | 0.986 | 0.333 | 0.393 | 0.990 | 0.296 |
| 125 | SARIMA | 339.00 | 0.739 | 0.963 | 0.998 | 0.480 | 0.977 | 0.240 | 0.420 | 0.981 | 0.230 | 0.448 | 0.979 | 0.235 |
| | Persistence | 499.00 | 0.763 | 0.964 | 0.012 | 0.440 | 0.970 | 0.110 | 0.120 | 0.993 | 0.030 | 0.189 | 0.981 | 0.047 |
| | LSTM | 337.85 | 0.776 | 0.972 | 0.939 | 0.495 | 0.979 | 0.280 | 0.478 | 0.978 | 0.266 | 0.486 | 0.978 | 0.273 |
| 150 | SARIMA | 305.00 | 0.848 | 0.959 | 1.096 | 0.250 | 0.984 | 0.130 | 0.460 | 0.974 | 0.140 | 0.324 | 0.979 | 0.135 |
| | Persistence | 446.00 | 0.863 | 0.965 | 0.034 | 0.400 | 0.982 | 0.000 | 0.460 | 0.984 | 0.000 | 0.428 | 0.983 | NaN |
| | LSTM | 327.52 | 0.862 | 0.966 | 0.831 | 0.230 | 0.984 | 0.131 | 0.387 | 0.972 | 0.146 | 0.289 | 0.978 | 0.138 |
| 175 | SARIMA | 180.00 | 0.852 | 0.951 | 1.040 | 0.200 | 0.988 | 0.110 | 0.440 | 0.967 | 0.220 | 0.275 | 0.977 | 0.147 |
| | Persistence | 233.00 | 0.871 | 0.955 | 0.044 | 0.170 | 0.985 | 0.080 | 0.220 | 0.979 | 0.110 | 0.192 | 0.982 | 0.093 |
| | LSTM | 174.00 | 0.888 | 0.969 | 0.823 | 0.283 | 0.988 | 0.070 | 0.480 | 0.975 | 0.110 | 0.356 | 0.981 | 0.086 |
| 200 | SARIMA | 148.00 | 0.814 | 0.952 | 1.063 | 0.260 | 0.982 | 0.560 | 0.560 | 0.972 | 0.310 | 0.355 | 0.977 | 0.399 |
| | Persistence | 222.00 | 0.826 | 0.958 | 0.359 | 0.310 | 0.980 | 0.250 | 0.630 | 0.973 | 0.130 | 0.416 | 0.976 | 0.171 |
| | LSTM | 162.50 | 0.834 | 0.964 | 1.008 | 0.289 | 0.985 | 0.478 | 0.562 | 0.970 | 0.445 | 0.382 | 0.977 | 0.461 |
| 250 | SARIMA | 105.00 | 0.881 | 0.930 | 1.107 | 0.230 | 0.980 | 0.670 | 0.500 | 0.979 | 0.220 | 0.315 | 0.979 | 0.331 |
| | Persistence | 140.00 | 0.899 | 0.942 | 0.115 | 0.290 | 0.975 | 0.000 | 0.330 | 0.988 | 0.000 | 0.309 | 0.981 | NaN |
| | LSTM | 127.55 | 0.879 | 0.922 | 0.941 | 0.240 | 0.979 | 0.668 | 0.284 | 0.970 | 0.518 | 0.260 | 0.975 | 0.583 |
| 300 | SARIMA | 59.00 | 0.864 | 0.894 | 0.840 | 0.440 | 0.982 | 0.070 | 0.640 | 0.961 | 0.140 | 0.521 | 0.971 | 0.093 |
| | Persistence | 78.00 | 0.882 | 0.912 | 0.028 | 0.710 | 0.979 | 0.170 | 0.450 | 0.987 | 0.140 | 0.551 | 0.983 | 0.154 |
| | LSTM | 55.44 | 0.877 | 0.910 | 0.891 | 0.418 | 0.977 | 0.120 | 0.455 | 0.968 | 0.093 | 0.436 | 0.973 | 0.105 |

**Table A.1:** Results for all price groups with no additional input or features at time $t+1$.

Month $t+2$

| PG | Model | MAE | Score | $\rho$ | Shift | $PRC_{in}$ | $PRC_{stay}$ | $PRC_{out}$ | $RCL_{in}$ | $RCL_{stay}$ | $RCL_{out}$ | $F1_{in}$ | $F1_{stay}$ | $F1_{out}$ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | SARIMA | 548.45 | 0.566 | 0.891 | 1.378 | 0.169 | 0.971 | 0.338 | 0.500 | 0.984 | 0.355 | 0.253 | 0.978 | 0.346 |
| 100 | Persistence | 778.96 | 0.669 | 0.918 | 0.028 | 0.250 | 0.962 | 0.063 | 0.429 | 0.991 | 0.000 | 0.316 | 0.976 | NaN |
| | LSTM | 447.67 | 0.687 | 0.965 | 0.945 | 0.354 | 0.987 | 0.330 | 0.531 | 0.981 | 0.351 | 0.425 | 0.984 | 0.340 |
| | SARIMA | 605.00 | 0.609 | 0.911 | 0.960 | 0.458 | 0.963 | 0.326 | 0.373 | 0.969 | 0.313 | 0.411 | 0.966 | 0.319 |
| 125 | Persistence | 923.00 | 0.630 | 0.920 | 0.007 | 0.444 | 0.948 | 0.286 | 0.067 | 0.995 | 0.042 | 0.116 | 0.971 | 0.073 |
| | LSTM | 483.53 | 0.676 | 0.942 | 0.832 | 0.389 | 0.644 | 0.188 | 0.336 | 0.647 | 0.194 | 0.361 | 0.645 | 0.191 |
| | SARIMA | 492.00 | 0.757 | 0.920 | 0.916 | 0.323 | 0.975 | 0.227 | 0.476 | 0.967 | 0.217 | 0.385 | 0.971 | 0.222 |
| 150 | Persistence | 824.00 | 0.775 | 0.927 | 0.020 | 0.533 | 0.969 | 0.000 | 0.381 | 0.985 | 0.000 | 0.444 | 0.977 | NaN |
| | LSTM | 435.24 | 0.786 | 0.933 | 0.789 | 0.308 | 0.969 | 0.149 | 0.440 | 0.968 | 0.134 | 0.362 | 0.969 | 0.141 |
| | SARIMA | 322.00 | 0.765 | 0.887 | 1.050 | 0.259 | 0.975 | 0.125 | 0.389 | 0.958 | 0.188 | 0.311 | 0.966 | 0.150 |
| 175 | Persistence | 425.00 | 0.784 | 0.907 | 0.026 | 0.333 | 0.970 | 0.000 | 0.222 | 0.979 | 0.000 | 0.266 | 0.974 | NaN |
| | LSTM | 250.67 | 0.840 | 0.940 | 0.815 | 0.514 | 0.982 | 0.179 | 0.722 | 0.968 | 0.250 | 0.600 | 0.975 | 0.209 |
| | SARIMA | 246.00 | 0.728 | 0.897 | 0.915 | 0.351 | 0.970 | 0.438 | 0.565 | 0.967 | 0.259 | 0.433 | 0.968 | 0.326 |
| 200 | Persistence | 402.00 | 0.737 | 0.908 | 0.210 | 0.250 | 0.960 | 0.200 | 0.304 | 0.971 | 0.074 | 0.274 | 0.965 | 0.108 |
| | LSTM | 230.87 | 0.769 | 0.892 | 0.853 | 0.338 | 0.969 | 0.530 | 0.600 | 0.966 | 0.363 | 0.432 | 0.967 | 0.431 |
| | SARIMA | 183.00 | 0.813 | 0.851 | 1.000 | 0.235 | 0.968 | 0.667 | 0.500 | 0.970 | 0.250 | 0.320 | 0.969 | 0.364 |
| 250 | Persistence | 256.00 | 0.827 | 0.876 | 0.064 | 0.333 | 0.958 | 0.000 | 0.250 | 0.990 | 0.000 | 0.286 | 0.974 | NaN |
| | LSTM | 167.13 | 0.822 | 0.857 | 0.937 | 0.274 | 0.953 | 0.633 | 0.444 | 0.980 | 0.190 | 0.339 | 0.966 | 0.292 |
| | SARIMA | 100.00 | 0.794 | 0.824 | 0.882 | 0.478 | 0.962 | 0.091 | 0.524 | 0.941 | 0.167 | 0.500 | 0.951 | 0.118 |
| 300 | Persistence | 141.00 | 0.817 | 0.849 | 0.018 | 0.857 | 0.954 | 0.143 | 0.286 | 0.987 | 0.083 | 0.429 | 0.970 | 0.105 |
| | LSTM | 82.58 | 0.825 | 0.872 | 0.774 | 0.667 | 0.948 | 0.056 | 0.312 | 0.974 | 0.075 | 0.425 | 0.961 | 0.064 |

**Table A.2:** Results for all price groups with no additional input or features at time $t+2$.

Month $t+3$

| PG | Model | MAE | Score | $\rho$ | Shift | $PRC_{in}$ | $PRC_{stay}$ | $PRC_{out}$ | $RCL_{in}$ | $RCL_{stay}$ | $RCL_{out}$ | $F1_{in}$ | $F1_{stay}$ | $F1_{out}$ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 100 | SARIMA | 811.26 | 0.474 | 0.824 | 1.437 | 0.323 | 0.974 | 0.304 | 0.262 | 0.965 | 0.310 | 0.289 | 0.970 | 0.307 |
| | Persistence | 1096.05 | 0.592 | 0.876 | 0.021 | 0.194 | 0.991 | 0.000 | 0.095 | 0.990 | 0.014 | 0.128 | 0.990 | 0.000 |
| | LSTM | 624.00 | 0.609 | 0.940 | 0.889 | 0.352 | 0.982 | 0.326 | 0.467 | 0.979 | 0.312 | 0.402 | 0.980 | 0.319 |
| 125 | SARIMA | 882.00 | 0.509 | 0.851 | 0.971 | 0.373 | 0.943 | 0.250 | 0.305 | 0.951 | 0.254 | 0.336 | 0.947 | 0.252 |
| | Persistence | 1304.00 | 0.538 | 0.874 | 0.005 | 0.555 | 0.929 | 0.286 | 0.061 | 0.995 | 0.032 | 0.110 | 0.961 | 0.058 |
| | LSTM | 703.84 | 0.591 | 0.904 | 0.761 | 0.414 | 0.634 | 0.175 | 0.313 | 0.633 | 0.164 | 0.357 | 0.633 | 0.170 |
| 150 | SARIMA | 703.00 | 0.671 | 0.864 | 0.961 | 0.359 | 0.969 | 0.276 | 0.500 | 0.960 | 0.276 | 0.418 | 0.964 | 0.276 |
| | Persistence | 1151.00 | 0.712 | 0.890 | 0.015 | 0.500 | 0.958 | 0.000 | 0.250 | 0.987 | 0.000 | 0.333 | 0.972 | NaN |
| | LSTM | 617.30 | 0.729 | 0.897 | 0.728 | 0.348 | 0.968 | 0.152 | 0.369 | 0.962 | 0.138 | 0.358 | 0.965 | 0.145 |
| 175 | SARIMA | 463.00 | 0.698 | 0.827 | 1.070 | 0.270 | 0.962 | 0.091 | 0.370 | 0.941 | 0.136 | 0.312 | 0.951 | 0.109 |
| | Persistence | 598.00 | 0.725 | 0.858 | 0.020 | 0.417 | 0.955 | 0.000 | 0.185 | 0.981 | 0.000 | 0.256 | 0.968 | NaN |
| | LSTM | 363.00 | 0.769 | 0.894 | 0.755 | 0.378 | 0.971 | 0.200 | 0.581 | 0.965 | 0.227 | 0.458 | 0.968 | 0.213 |
| 200 | SARIMA | 341.00 | 0.661 | 0.840 | 0.873 | 0.463 | 0.957 | 0.333 | 0.594 | 0.964 | 0.189 | 0.520 | 0.960 | 0.241 |
| | Persistence | 564.00 | 0.666 | 0.850 | 0.160 | 0.259 | 0.942 | 0.250 | 0.219 | 0.971 | 0.081 | 0.237 | 0.956 | 0.122 |
| | LSTM | 321.21 | 0.696 | 0.877 | 0.786 | 0.459 | 0.960 | 0.456 | 0.479 | 0.959 | 0.333 | 0.469 | 0.960 | 0.385 |
| 250 | SARIMA | 255.00 | 0.753 | 0.787 | 0.968 | 0.217 | 0.955 | 0.636 | 0.454 | 0.955 | 0.304 | 0.294 | 0.955 | 0.411 |
| | Persistence | 356.00 | 0.770 | 0.811 | 0.049 | 0.333 | 0.939 | 0.000 | 0.182 | 0.990 | 0.000 | 0.235 | 0.964 | NaN |
| | LSTM | 221.92 | 0.767 | 0.807 | 0.863 | 0.348 | 0.970 | 0.283 | 0.394 | 0.974 | 0.130 | 0.369 | 0.972 | 0.178 |
| 300 | SARIMA | 138.00 | 0.734 | 0.750 | 0.851 | 0.481 | 0.945 | 0.038 | 0.464 | 0.927 | 0.067 | 0.472 | 0.936 | 0.048 |
| | Persistence | 194.00 | 0.759 | 0.786 | 0.014 | 0.833 | 0.933 | 0.000 | 0.179 | 0.985 | 0.000 | 0.295 | 0.958 | NaN |
| | LSTM | 112.61 | 0.780 | 0.828 | 0.707 | 0.502 | 0.953 | 0.085 | 0.298 | 0.969 | 0.089 | 0.374 | 0.961 | 0.087 |

**Table A.3:** Results for all price groups with no additional input or features at time $t+3$.

| Time | Model | Features | MAE | Score | $\rho$ | Shift | $PRC_{in}$ | $PRC_{stay}$ | $PRC_{out}$ | $RCL_{in}$ | $RCL_{stay}$ | $RCL_{out}$ | $F1_{in}$ | $F1_{stay}$ | $F1_{out}$ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | SARIMA | None | 180.00 | 0.852 | 0.951 | 1.040 | 0.200 | 0.988 | 0.110 | 0.440 | 0.967 | 0.220 | 0.275 | 0.977 | 0.147 |
| | SARIMAX | Price | 185.00 | 0.854 | 0.954 | 1.090 | 0.230 | 0.989 | 0.100 | 0.560 | 0.964 | 0.220 | 0.326 | 0.976 | 0.138 |
| | SARIMAX | Top sales | 191.543 | 0.858 | 0.955 | 1.105 | 0.217 | 0.988 | 0.045 | 0.286 | 0.978 | 0.097 | 0.247 | 0.983 | 0.062 |
| | Persistence | None | 233.00 | 0.871 | 0.955 | 0.044 | 0.170 | 0.985 | 0.080 | 0.220 | 0.979 | 0.110 | 0.192 | 0.982 | 0.093 |
| | LSTM | None | 174.00 | 0.888 | 0.969 | 0.823 | 0.283 | 0.988 | 0.070 | 0.480 | 0.975 | 0.110 | 0.356 | 0.981 | 0.086 |
| $t+1$ | LSTM | Price | 174.00 | 0.884 | 0.968 | 0.882 | 0.303 | 0.988 | 0.090 | 0.560 | 0.975 | 0.147 | 0.394 | 0.981 | 0.112 |
| | LSTM | Top sales | 173.67 | 0.892 | 0.972 | 0.729 | 0.297 | 0.989 | 0.153 | 0.520 | 0.977 | 0.257 | 0.378 | 0.983 | 0.192 |
| | LSTM | Price group | 165.67 | 0.883 | 0.968 | 0.894 | 0.283 | 0.990 | 0.183 | 0.557 | 0.974 | 0.330 | 0.376 | 0.982 | 0.236 |
| | LSTM | Selection | 168.33 | 0.883 | 0.966 | 0.859 | 0.300 | 0.988 | 0.113 | 0.520 | 0.976 | 0.183 | 0.380 | 0.982 | 0.140 |
| | LSTM | Newness | 173.00 | 0.885 | 0.968 | 0.845 | 0.257 | 0.988 | 0.127 | 0.480 | 0.973 | 0.220 | 0.334 | 0.981 | 0.161 |
| | SARIMA | None | 322.00 | 0.765 | 0.887 | 1.050 | 0.259 | 0.975 | 0.125 | 0.389 | 0.958 | 0.188 | 0.311 | 0.966 | 0.150 |
| | SARIMAX | Price | 332.00 | 0.761 | 0.887 | 1.150 | 0.235 | 0.977 | 0.129 | 0.444 | 0.946 | 0.250 | 0.307 | 0.961 | 0.170 |
| | SARIMAX | Top sales | 333.49 | 0.768 | 0.888 | 1.083 | 0.333 | 0.965 | 0.086 | 0.556 | 0.961 | 0.111 | 0.417 | 0.963 | 0.097 |
| | Persistence | None | 425.00 | 0.784 | 0.907 | 0.026 | 0.333 | 0.970 | 0.000 | 0.222 | 0.979 | 0.000 | 0.266 | 0.974 | NaN |
| | LSTM | None | 250.67 | 0.840 | 0.940 | 0.815 | 0.514 | 0.982 | 0.179 | 0.722 | 0.968 | 0.250 | 0.600 | 0.975 | 0.209 |
| $t+2$ | LSTM | Price | 237.00 | 0.831 | 0.938 | 0.824 | 0.493 | 0.983 | 0.177 | 0.704 | 0.967 | 0.250 | 0.580 | 0.975 | 0.207 |
| | LSTM | Top sales | 257.33 | 0.836 | 0.943 | 0.730 | 0.479 | 0.981 | 0.224 | 0.648 | 0.970 | 0.292 | 0.551 | 0.976 | 0.254 |
| | LSTM | Price group | 240.00 | 0.834 | 0.939 | 0.842 | 0.487 | 0.982 | 0.183 | 0.722 | 0.966 | 0.271 | 0.582 | 0.974 | 0.218 |
| | LSTM | Selection | 240.33 | 0.832 | 0.937 | 0.825 | 0.473 | 0.981 | 0.169 | 0.667 | 0.968 | 0.229 | 0.554 | 0.974 | 0.195 |
| | LSTM | Newness | 248.33 | 0.829 | 0.936 | 0.876 | 0.456 | 0.981 | 0.167 | 0.685 | 0.964 | 0.250 | 0.548 | 0.973 | 0.200 |
| | SARIMA | None | 463.00 | 0.698 | 0.827 | 1.070 | 0.270 | 0.962 | 0.910 | 0.370 | 0.941 | 0.136 | 0.312 | 0.951 | 0.237 |
| | SARIMAX | Price | 480.00 | 0.689 | 0.824 | 1.160 | 0.311 | 0.970 | 0.171 | 0.518 | 0.932 | 0.318 | 0.389 | 0.951 | 0.222 |
| | SARIMAX | Top sales | 477.16 | 0.701 | 0.828 | 1.105 | 0.556 | 0.946 | 0.188 | 0.481 | 0.940 | 0.136 | 0.516 | 0.943 | 0.158 |
| | Persistence | None | 598.00 | 0.725 | 0.858 | 0.020 | 0.417 | 0.955 | 0.000 | 0.185 | 0.981 | 0.000 | 0.256 | 0.968 | NaN |
| | LSTM | None | 363.00 | 0.769 | 0.894 | 0.755 | 0.378 | 0.971 | 0.200 | 0.581 | 0.965 | 0.227 | 0.458 | 0.968 | 0.213 |
| $t+3$ | LSTM | Price | 355.33 | 0.764 | 0.889 | 0.776 | 0.484 | 0.970 | 0.190 | 0.568 | 0.959 | 0.242 | 0.523 | 0.964 | 0.213 |
| | LSTM | Top sales | 359.67 | 0.770 | 0.900 | 0.622 | 0.483 | 0.968 | 0.245 | 0.524 | 0.971 | 0.256 | 0.503 | 0.970 | 0.250 |
| | LSTM | Price group | 340.00 | 0.769 | 0.894 | 0.764 | 0.512 | 0.971 | 0.219 | 0.593 | 0.963 | 0.258 | 0.550 | 0.967 | 0.237 |
| | LSTM | Selection | 336.33 | 0.773 | 0.898 | 0.769 | 0.469 | 0.970 | 0.248 | 0.519 | 0.964 | 0.273 | 0.493 | 0.967 | 0.260 |
| | LSTM | Newness | 349.33 | 0.767 | 0.890 | 0.811 | 0.441 | 0.969 | 0.228 | 0.506 | 0.960 | 0.273 | 0.471 | 0.964 | 0.248 |

**Table A.4:** Results for different features, no additional price groups used for training and validation, tested on PG175.

| Time | PG | MAE | Score | $\rho$ | Shift | $PRC_{in}$ | $PRC_{stay}$ | $PRC_{out}$ | $RCL_{in}$ | $RCL_{stay}$ | $RCL_{out}$ | $F1_{in}$ | $F1_{stay}$ | $F1_{out}$ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 175 | 174.00 | 0.888 | 0.969 | 0.823 | 0.283 | 0.988 | 0.070 | 0.480 | 0.975 | 0.110 | 0.356 | 0.981 | 0.086 |
| | 100,125,150,175,200,250,300 | 111.67 | 0.887 | 0.968 | 0.901 | 0.357 | 0.992 | 0.197 | 0.780 | 0.972 | 0.367 | 0.490 | 0.982 | 0.256 |
| t+1 | 125,150,175,200,250 | 115.40 | 0.888 | 0.968 | 0.834 | 0.335 | 0.990 | 0.169 | 0.581 | 0.979 | 0.264 | 0.425 | 0.984 | 0.206 |
| | 150,175,200 | 149.90 | 0.873 | 0.965 | 0.918 | 0.305 | 0.988 | 0.118 | 0.501 | 0.982 | 0.177 | 0.379 | 0.985 | 0.141 |
| | 100,125,150,175 | 121.60 | 0.892 | 0.969 | 0.720 | 0.308 | 0.987 | 0.114 | 0.527 | 0.983 | 0.228 | 0.389 | 0.985 | 0.152 |
| | 175,200,250,300 | 172.68 | 0.889 | 0.970 | 0.856 | 0.254 | 0.988 | 0.090 | 0.476 | 0.982 | 0.158 | 0.331 | 0.985 | 0.114 |
| | 175 | 250.67 | 0.840 | 0.940 | 0.815 | 0.514 | 0.982 | 0.179 | 0.722 | 0.968 | 0.250 | 0.600 | 0.975 | 0.209 |
| | 100,125,150,175,200,250,300 | 161.00 | 0.832 | 0.939 | 0.807 | 0.448 | 0.981 | 0.253 | 0.630 | 0.973 | 0.303 | 0.523 | 0.977 | 0.276 |
| t+2 | 125,150,175,200,250 | 167.18 | 0.836 | 0.938 | 0.824 | 0.535 | 0.974 | 0.217 | 0.648 | 0.973 | 0.265 | 0.586 | 0.974 | 0.238 |
| | 150,175,200 | 193.09 | 0.825 | 0.937 | 0.791 | 0.500 | 0.969 | 0.169 | 0.519 | 0.972 | 0.233 | 0.509 | 0.971 | 0.196 |
| | 100,125,150,175 | 171.15 | 0.839 | 0.942 | 0.722 | 0.518 | 0.970 | 0.247 | 0.519 | 0.977 | 0.167 | 0.518 | 0.973 | 0.199 |
| | 175,200,250,300 | 247.68 | 0.836 | 0.939 | 0.820 | 0.521 | 0.971 | 0.187 | 0.519 | 0.970 | 0.185 | 0.520 | 0.971 | 0.186 |
| | 175 | 363.00 | 0.769 | 0.894 | 0.755 | 0.378 | 0.971 | 0.200 | 0.581 | 0.965 | 0.227 | 0.458 | 0.968 | 0.213 |
| | 100,125,150,175,200,250,300 | 242.00 | 0.770 | 0.892 | 0.774 | 0.457 | 0.969 | 0.232 | 0.531 | 0.960 | 0.275 | 0.491 | 0.964 | 0.252 |
| t+3 | 125,150,175,200,250 | 243.05 | 0.771 | 0.891 | 0.748 | 0.480 | 0.968 | 0.253 | 0.580 | 0.964 | 0.246 | 0.525 | 0.966 | 0.250 |
| | 150,175,200 | 281.29 | 0.753 | 0.889 | 0.742 | 0.704 | 0.965 | 0.275 | 0.568 | 0.958 | 0.217 | 0.629 | 0.961 | 0.243 |
| | 100,125,150,175 | 246.87 | 0.771 | 0.898 | 0.668 | 0.741 | 0.969 | 0.314 | 0.556 | 0.965 | 0.275 | 0.635 | 0.967 | 0.293 |
| | 175,200,250,300 | 362.63 | 0.760 | 0.886 | 0.804 | 0.722 | 0.965 | 0.229 | 0.605 | 0.961 | 0.227 | 0.658 | 0.963 | 0.228 |

**Table A.5:** Results for LSTM models with different price groups used for training and validation, no additional features, tested on PG175.

| Time | PG | Features | MAE | Score | $\rho$ | Shift | $PRC_{in}$ | $PRC_{stay}$ | $PRC_{out}$ | $RCL_{in}$ | $RCL_{stay}$ | $RCL_{out}$ | $F1_{in}$ | $F1_{stay}$ | $F1_{out}$ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 175 | None | 174.00 | 0.888 | 0.969 | 0.823 | 0.283 | 0.988 | 0.070 | 0.480 | 0.975 | 0.110 | 0.356 | 0.981 | 0.086 |
| | All | None | 111.67 | 0.887 | 0.968 | 0.901 | 0.357 | 0.992 | 0.197 | 0.780 | 0.972 | 0.367 | 0.490 | 0.982 | 0.256 |
| t+1 | All | All | 126.32 | 0.896 | 0.971 | 0.828 | 0.328 | 0.989 | 0.192 | 0.500 | 0.981 | 0.223 | 0.396 | 0.985 | 0.207 |
| | All | Price group | 113.17 | 0.89 | 0.97 | 0.94 | 0.33 | 0.99 | 0.20 | 0.48 | 0.98 | 0.23 | 0.394 | 0.987 | 0.212 |
| | All | Top sales | 135.34 | 0.89 | 0.97 | 0.83 | 0.34 | 0.99 | 0.16 | 0.49 | 0.98 | 0.24 | 0.404 | 0.985 | 0.192 |
| | All | Price group, Top sales | 117.28 | 0.89 | 0.97 | 0.87 | 0.29 | 0.99 | 0.16 | 0.43 | 0.98 | 0.17 | 0.344 | 0.984 | 0.167 |
| | 175 | None | 250.67 | 0.840 | 0.940 | 0.815 | 0.514 | 0.982 | 0.179 | 0.722 | 0.968 | 0.250 | 0.600 | 0.975 | 0.209 |
| | All | None | 161.00 | 0.832 | 0.939 | 0.807 | 0.448 | 0.981 | 0.253 | 0.630 | 0.973 | 0.303 | 0.523 | 0.977 | 0.276 |
| t+2 | All | All | 183.71 | 0.835 | 0.944 | 0.740 | 0.478 | 0.970 | 0.286 | 0.630 | 0.975 | 0.300 | 0.543 | 0.973 | 0.293 |
| | All | Price group | 168.64 | 0.83 | 0.94 | 0.86 | 0.45 | 0.97 | 0.20 | 0.78 | 0.97 | 0.40 | 0.571 | 0.970 | 0.271 |
| | All | Top sales | 201.03 | 0.83 | 0.94 | 0.72 | 0.49 | 0.97 | 0.23 | 0.67 | 0.97 | 0.27 | 0.565 | 0.971 | 0.249 |
| | All | Price group, Top sales | 195.24 | 0.83 | 0.94 | 0.75 | 0.43 | 0.97 | 0.20 | 0.63 | 0.97 | 0.30 | 0.508 | 0.969 | 0.240 |
| | 175 | None | 363.00 | 0.769 | 0.894 | 0.755 | 0.378 | 0.971 | 0.200 | 0.581 | 0.965 | 0.227 | 0.458 | 0.968 | 0.213 |
| | All | None | 242.00 | 0.770 | 0.892 | 0.774 | 0.457 | 0.969 | 0.232 | 0.531 | 0.960 | 0.275 | 0.491 | 0.964 | 0.252 |
| t+3 | All | All | 262.78 | 0.773 | 0.890 | 0.723 | 0.648 | 0.971 | 0.275 | 0.531 | 0.965 | 0.319 | 0.584 | 0.968 | 0.295 |
| | All | Price group | 236.44 | 0.768 | 0.891 | 0.841 | 0.704 | 0.967 | 0.314 | 0.568 | 0.956 | 0.261 | 0.629 | 0.962 | 0.285 |
| | All | Top sales | 270.04 | 0.768 | 0.892 | 0.675 | 0.630 | 0.971 | 0.294 | 0.506 | 0.966 | 0.246 | 0.561 | 0.968 | 0.268 |
| | All | Price group, Top sales | 271.41 | 0.765 | 0.886 | 0.735 | 0.593 | 0.966 | 0.235 | 0.506 | 0.957 | 0.246 | 0.546 | 0.961 | 0.241 |

**Table A.6:** Results for LSTM models combining features and price groups in an attempt to optimize the model, tested on PG175.

| PG | Time | MAE | Score | $\rho$ | Shift | $PRC_{in}$ | $PRC_{stay}$ | $PRC_{out}$ | $RCL_{in}$ | $RCL_{stay}$ | $RCL_{out}$ | $F1_{in}$ | $F1_{stay}$ | $F1_{out}$ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | t+1 | 298.08 | 0.798 | 0.965 | 1.021 | 0.263 | 0.990 | 0.296 | 0.548 | 0.985 | 0.280 | 0.355 | 0.987 | 0.287 |
| 100 | t+2 | 411.88 | 0.695 | 0.927 | 0.890 | 0.324 | 0.979 | 0.371 | 0.430 | 0.981 | 0.262 | 0.370 | 0.980 | 0.307 |
| | t+3 | 615.68 | 0.612 | 0.886 | 0.833 | 0.292 | 0.970 | 0.334 | 0.333 | 0.977 | 0.211 | 0.311 | 0.973 | 0.259 |
| | t+1 | 241.16 | 0.787 | 0.975 | 0.938 | 0.519 | 0.978 | 0.228 | 0.465 | 0.981 | 0.215 | 0.490 | 0.979 | 0.222 |
| 125 | t+2 | 355.05 | 0.689 | 0.945 | 0.775 | 0.684 | 0.969 | 0.352 | 0.565 | 0.975 | 0.340 | 0.619 | 0.972 | 0.346 |
| | t+3 | 512.02 | 0.600 | 0.909 | 0.711 | 0.645 | 0.953 | 0.308 | 0.492 | 0.966 | 0.281 | 0.558 | 0.959 | 0.294 |
| | t+1 | 204.69 | 0.878 | 0.969 | 0.799 | 0.250 | 0.983 | 0.143 | 0.410 | 0.976 | 0.143 | 0.311 | 0.980 | 0.143 |
| 150 | t+2 | 276.48 | 0.805 | 0.939 | 0.738 | 0.329 | 0.974 | 0.192 | 0.476 | 0.965 | 0.203 | 0.389 | 0.970 | 0.197 |
| | t+3 | 410.07 | 0.742 | 0.905 | 0.716 | 0.317 | 0.967 | 0.220 | 0.452 | 0.953 | 0.253 | 0.373 | 0.960 | 0.235 |
| | t+1 | 111.67 | 0.887 | 0.968 | 0.901 | 0.357 | 0.992 | 0.197 | 0.780 | 0.972 | 0.367 | 0.490 | 0.982 | 0.256 |
| 175 | t+2 | 161.00 | 0.832 | 0.939 | 0.807 | 0.448 | 0.981 | 0.253 | 0.630 | 0.973 | 0.303 | 0.523 | 0.977 | 0.276 |
| | t+3 | 242.00 | 0.770 | 0.892 | 0.774 | 0.457 | 0.969 | 0.232 | 0.531 | 0.960 | 0.275 | 0.491 | 0.964 | 0.252 |
| | t+1 | 65.85 | 0.901 | 0.943 | 0.922 | 0.230 | 0.979 | 0.556 | 0.444 | 0.981 | 0.148 | 0.303 | 0.980 | 0.234 |
| 200 | t+2 | 109.26 | 0.839 | 0.881 | 0.871 | 0.201 | 0.964 | 0.581 | 0.375 | 0.971 | 0.216 | 0.261 | 0.967 | 0.315 |
| | t+3 | 150.99 | 0.788 | 0.842 | 0.778 | 0.196 | 0.945 | 0.667 | 0.300 | 0.971 | 0.160 | 0.237 | 0.958 | 0.258 |
| | t+1 | 127.55 | 0.879 | 0.922 | 0.941 | 0.240 | 0.979 | 0.668 | 0.284 | 0.970 | 0.518 | 0.260 | 0.975 | 0.583 |
| 250 | t+2 | 167.13 | 0.822 | 0.857 | 0.937 | 0.274 | 0.953 | 0.633 | 0.444 | 0.980 | 0.190 | 0.339 | 0.966 | 0.292 |
| | t+3 | 221.92 | 0.767 | 0.807 | 0.863 | 0.348 | 0.970 | 0.283 | 0.394 | 0.974 | 0.130 | 0.369 | 0.972 | 0.178 |
| | t+1 | 57.16 | 0.884 | 0.919 | 0.849 | 0.421 | 0.977 | 0.116 | 0.394 | 0.976 | 0.143 | 0.407 | 0.976 | 0.128 |
| 300 | t+2 | 83.84 | 0.832 | 0.875 | 0.766 | 0.639 | 0.956 | 0.163 | 0.333 | 0.977 | 0.139 | 0.438 | 0.967 | 0.150 |
| | t+3 | 112.41 | 0.780 | 0.829 | 0.750 | 0.605 | 0.944 | 0.277 | 0.310 | 0.971 | 0.244 | 0.409 | 0.957 | 0.260 |

**Table A.7:** Results for all price groups using final LSTM model. The model is trained on all price groups with no additional features.

# Appendix B

# Metadata Classes

This table is similar to a table presented in my specialization project. It presents a list of all of the columns of data and metadata available in the three data sets, *Sales* (S), *Products* (P), and *Rankings* (R). The first column is the original name in Norwegian, the second column is which file contained that name, S, P, or R, and the third column is the English translation and explanation for certain names. Many of these features were at some point analyzed or evaluated, but the ones presented in this thesis are marked by * after the name. Some of the columns are scarce, i.e. sugar, seal type, taste, etc., making them difficult to use for this type of analysis. Validity of the data is not guaranteed, the column "Liters this month last year" proves to be full of flaws when compared to "Liters this month this year" shifted one year.

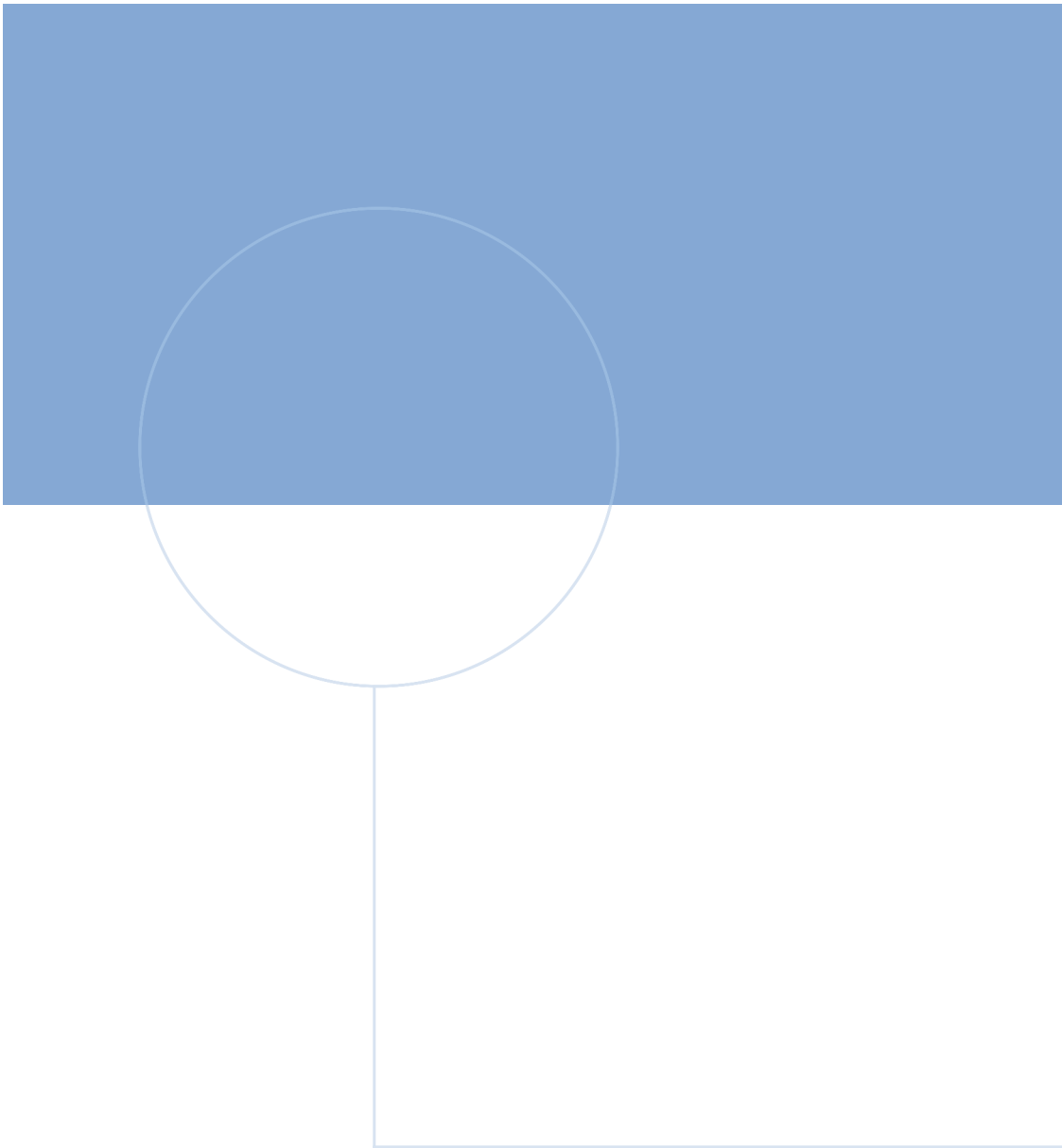
| Names | Data Set | Definition |
|---|---|---|
| År* | S,R | Year of sale, parsed with month |
| Måned* | S,R | Month of sale, parsed with year |
| Land | S,P | Country |
| Distrikt | S,P | District |
| Hovedvaretype | S,P | Main product group |
| Varetype* | S,P | Product group |
| Subvaretype | S,P | Sub product group |
| Artikkelnr* | S,R | Article number |
| Artikkelnavn | S,P | Article name |
| Årgang | S,P | Vintage |
| Volum | S,P | Volume |
| Alkoholprosent | S,P | Percentage alcohol |
| Emballasjetype | S,P | Type of packaging |
| Miljøsmart | S | Environmentally friendly packaging |
| Økologisk | S | Organic |
| Utvalg* | S | Selection |
| Kategori | S,P | Category |
| Grossist | S,R | Wholesaler |
| Distributør | S,P | Distributor |
| Liter denne måned i år* | S | Liters this month this year (Liter) |
| Liter denne måned i fjor | S | Liters this month last year |
| Liter hittil i år | S | Liters so far this year |
| Liter hittil i fjor | S | Liters so far last year |
| Liter siste 12 måneder | S | Liters last 12 months |
| Salgspris | S | Sales price |
| Kvalitet | P | Quality |
| Produsent | P | Producer |
| Importør | P | Importer |
| Vårt varenr | P | Our article number |
| VMP ID* | P | Vinmonopolet ID/Article number |
| VMP lanseringsdato | P | VMP launch date |
| VMP utgåttdato | P | VMP expired date |
| Status | P,R | Status |
| Innhold Sukker | P | Sugar contents |
| Syre | P | Acid |
| Tilsatt sulfitt | P | Added sulfite |
| Sertifikater | P | Certificates |
| Produksjonsvolum | P | Production volume |
| FPAK | P | Consumer packaging |
| DPAK | P | Distribution packaging |
| Produktutvalg | P | Form of availability |
| Korktype | P | Seal type |
| Enheter i forpakning | P | Units in packaging |
| Emballasjevekt (g) | P | Packaging weight |

| Names | Data Set | Definition |
|---|---|---|
| Vinmarker | P | Wine field |
| Vinifikasjon | P | Vinification |
| Farge | P | Color |
| Lukt | P | Smell |
| Smak | P | Taste |
| Annet | P | Other |
| Utsalgspris | P | Sales price |
| DDP pris | P | Delivered Duty Paid price |
| Horeca pris | P | Hotel, restaurant and catering price |
| Grossistpris | P | Wholesaler price |
| Innkjøpskostnad | P | Purchase cost |
| Innfraktkostnad | P | Freight cost |
| Vårt segment | R | Our segment |
| Produktgruppe | R | Product group |
| VMP segment | R | VMP segment |
| Rangering* | R | Rank |
| Styringstall* | R | Control number |
| Fredet | R | Protected |
| Segmentpris* | R | Segment price |
| Minimum | R | Minimum price |
| Maksimum | R | Maximum price |
| Netto Salg* | R | Net sales |
| Prosentandel sør | R | Percentage of sales south |
| Prosentandel øst | R | Percentage of sales east |
| Prosentandel vest | R | Percentage of sales west |
| Prosentandel nord | R | Percentage of sales north |

# Appendix C

# Code

A repository with the code for this project is found at:
`https://github.com/SteffiJF/ForecastingRedWineRankings`



NTNU

Norwegian University of
Science and Technology