Bachelor's thesis

NTNU
Norwegian University of Science and Technology
Faculty of Economics and Management
NTNU Business School

Henrik Bjørnstad, Øyvind Løvig Halvorsen &
Victor Carl Johan Rognås

Forecasting the price of salmon using machine learning algorithms

Bachelor's thesis in Bachelor i Økonomi og Administrasjon
Supervisor: Denis Becker
April 2023

NTNU
Norwegian University of
Science and Technology

Henrik Bjørnstad, Øyvind Løvig Halvorsen &
Victor Carl Johan Rognås

Forecasting the price of salmon using machine learning algorithms

Bachelor's thesis in Bachelor i Økonomi og Administrasjon
Supervisor: Denis Becker
April 2023

Norwegian University of Science and Technology
Faculty of Economics and Management
NTNU Business School

NTNU
Norwegian University of
Science and Technology

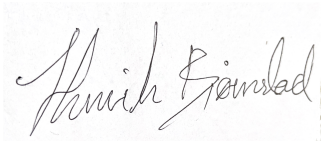
Preface

This bachelor thesis marks the end of our degree in Business Administration with specialization in Business Analytics at NTNU Handelshøyskolen. Throughout the process of writing this thesis we have enhanced our understanding of analytical modelling, as well as seeing how it can be applied in a real-world setting.

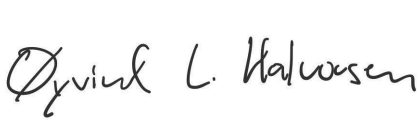
We would like to take the opportunity to thank our supervisor, Denis Becker, for his guidance and support throughout the project. His inputs and engagement regarding the topic has been greatly appreciated. We would also like to thank Gunnar Johnsen at Norges Råfisklag for providing us with the data that our models are built upon.

The contents of the thesis is at the expense of the authors.

Victor Carl Johan Rognås



Henrik Bjørnstad



Øyvind Løvig Halvorsen

Abstract

The purpose of this thesis is to utilize analytical models to predict the price of salmon using similar commodities and macroeconomic factors. To make these predictions we have used the cod price, halibut price, CPI and TWI. We have developed four models; ARIMA, SARIMA, SARIMAX and LSTM. These are all trained on a dataset gathered from Norsk Råfisklag. The models are trained on the data from the timeperiod spring 2013 to winter 2021, and tested on data from 2022.

Our research and testing has shown that the predictions improve when taking seasonality into account. The improvement from the ARIMA to the SARIMA model is very significant, whereas the improvement from SARIMA to the SARIMAX with one exogenous variable is very small. This is also clearly shown in the predictions from the three models, where the SARIMA and SARIMAX roughly follow the same trends, and the ARIMA simply predicts the same average price for all weeks. The best prediction from the LSTM model also comes close to the RMSE of the SARIMAX model, but the results from the LSTM model varies greatly depending on the lookback period. Lookbacks of 52 and 104 weeks are the best perfoming in terms of catching the trends, but the lookback periods of the models with the lowest RMSE is quite random. LSTM with short lookback periods predict the average quite well, meaning that the RMSE is low, however, when looking at the graphs its clear that this does not actually represent a good prediction.

Even though the SARIMAX model performs the best, there are still areas of uncertainty. Tests on the SARIMAX model show that we can't conclude whether or not the residuals are white noise. We can only conclude that we can't reject the null hypothesis that the residuals are white noise.

In conclusion the models taking seasonality into account can to some extent catch the trends in the price of salmon, but they are not able to predict the price of salmon with a high degree of accuracy.

Sammendrag

Formålet med denne oppgaven er å bruke analytiske modeller til å predikere lakesprisen ved hjelp av lignende råvarer og økonomiske faktorer. For å produsere disse prediksjonene har vi brukt torskeprisen, kveiteprisen, KPI og TWI. Vi har utviklet fire modeller; ARIMA, SARIMA, SARIMAX og LSTM. Disse er alle trent på et datasett samlet inn fra Norsk Råfisklag, Norges Bank, Fishpool og SSB. Modellene er trent på data fra tidsperioden vår 2013 til vinter 2021, og testet på data fra 2022.

Vår forskning og testing har vist at prediksjonene forbedres når sesongvariasjon tas i betraktning. Forbedringen fra ARIMA- til SARIMA-modellen er vesentlig, mens forbedringen fra SARIMA til SARIMAX med én eksogen variabel er veldig liten. Dette vises også tydelig i prediksjonene fra de tre modellene, der SARIMA og SARIMAX omtrent følger de samme trendene, mens ARIMA predikerer den samme gjennomsnittsprisen for alle uker. Den beste prediksjonen fra LSTM-modellen er også nærme RMSE'en til SARIMAX-modellen, men resultatene fra LSTM-modellen varierer stort avhengig av "lookback"-perioden. "Lookback" på 52 og 104 uker er best på å fange trendene, men "lookback"-periodene for LSTM med lavest RMSE er tilfeldig. Med kort "lookback"-periode predikeres gjennomsnittet relativt bra. Dette resulterer i lav RMSE, men ved å se på grafene er det tydelig at dette ikke representeterer en god prediksjon.

Selv om SARIMAX-modellen presterer best, er det fortsatt områder med usikkerhet. Tester på SARIMAX-modellen viser at vi ikke kan konkludere med om residualene er støy. Den eneste konklusjonen vi kan trekke er at vi ikke kan forkaste nullhypotesen om at residualene er støy.

Konklusjonen fra denne oppgaven er at modellene som tar høyde for sesongvariasjon til en viss grad klarer å følge trendene, men de klarer ikke å predikere lakseprisen med høy nøyaktighet.

List of Figures

List of Tables

Table of Contents

1 Introduction

In this thesis we would like to answer the question. Can we predict the pricing of salmon using similar commodities and macroeconomic factors? To answer this, we first need to look at fish as a commodity. ”As the largest traded food commodity in the world, seafood provides sustenance to billions of people world-wide. More than 3 billion people in the world rely on wild-caught and farmed seafood.” (WWF, 2019) Seafood has been a traded commodity for hundreds of years, and for most of this time has been one of Norway’s biggest exports. ”Norwegian clipfish has been exported to Southern Europe and beyond since the early 1700s.” (Norway, 2023) Fish world-wide has been an important source of protein. Facing an ever larger growing population and the sustainability issues that comes with it, the seafood industry plays a major role in solving these issues.

Norway’s biggest contribution towards solving these issues is the salmon farming industry. ”The salmon industry is one of the biggest industries in Norway.” (Johansen et al., 2019) The companies in the industry impact the rest of the Norwegian economy and society as a whole through labour and culture. This in turn means that the Norwegian society is indirectly affected by the price of salmon on the open market. Salmon export makes up about 2/3 of the Norwegian seafood export by value, amounting to a total of more than 105 billion NOK in 2022. This made salmon export the third largest exported commodity by value in Norway, below oil and gas. (Meisingset, 2023) (Sjømatråd, 2023)

Most of the salmon that is exported comes from fish farms. Salmon farming is a Norwegian lead global industry with four of the five biggest companies being Norwegian. (Berge, 2020) These companies all rely heavily on the price of salmon. We hope that by creating predictive models on the price of salmon short term we will gain insight into the near future of these major companies. This in turn could give an idea of the impact they might have, and the impact a changing salmon price have on the Norwegian economy and society.

Fishing as a trade is seasonal given that different fish wanders and breads at different times throughout the year. Although farmed fish are not subjected to this seasonality, society adjusted their consumer habits to this seasonality of commodities long before fish farms, and we suspected that this is the case for the salmon too. The quantitative data for sales

also support this notion.

As business analysts this research question is interesting because it utilizes methods and logic that are commonly used in the field. The theme of the research question is also highly relevant in the economy today for multiple reasons. Large actors in the economy are just now starting to implement the major advances in computation such as AI and other Artificial neural networks. This is something we will be using to see if it can help our prediction. The fish farming industry is a cornerstone in the Norwegian economy and has been touted as one of the ways for Norway to prosper after the oil is gone. The market and industry surrounding salmon has also been a subject of debates as of recently. This debate comes as a result of massive growth, profits and old tax-related incentives for fish farming. This has caused a lot of turbulence in the price of the salmon farming companies. In this thesis we will not be tackling the issues of the salmon tax debates and its impact on the industry. But we will be examining the changes in price that happened during the hight of this debate.

In order to answer the research question, we have structured the thesis as follows. In the first section we describe the previous literature and the economic theory that we found relevant to the issue. Then we continue by explaining the theories pertaining to the models we will be using. The next section talks about the gathering of our data, some of its basic properties, and the preparation needed before we can start modelling. Then we present our models and how they function in detail before we discuss the results. In the end we compare the models and scope out the needs for further research.

2 Theory and literature

2.1 Literature review

At the time of writing this thesis there are only a few papers directly forecasting the price of salmon. These papers use a variety of different methods to forecast the price of salmon. These papers also use numerous features, creating a great starting point and giving insight in to what to focus on and what to disregard in future research.

Vukina and Anderson (1994) is a study that forecasts the price of five different species of salmon in the Tokyo wholesale market. Four state-space models are used to predict and compare the prices by modelling non-stationary time series. When measuring the results by use of MSE and MAPE the results were found to be quite good. The results were also surprisingly good at predicting the correct direction of the price-movement.

This study was followed by Gu and Anderson (1995). This study combines OLS used to model the seasonality the seasonality removal with a state-space, time-series forecasting method to predict the price for the US salmon market. The result from this study clearly indicates that accounting for seasonal factor significantly increases the forecasting accuracy of the model.

In the study "Short-term salmon price forecasting" by Daumantas Bloznelis 16 different methods are used to forecast 1-5 weeks Atlantic salmon spot prices. Every method Bloznelis (2018) uses gets the directional movement correct 50% of the time for all forecasting horizons. For one week prediction k-nearest neighbour gives the best prediction. For two to three weeks prediction a vector error correction model using elastic net regulation gives the best results, and for four to five weeks future prices is the best method.

2.2 Economic Theory

2.2.1 Consumer choice theory

"Consumer choice theory is the study of how individuals make choices about what goods and services to consume. The theory is based on the assumption that consumers have

preferences for different goods and services and that they will choose the combination of goods and services that maximizes their utility subject to their budget constraints." (Perloff, 2017) This means that if you have two types of goods that are exchangeable to some degree, the price of one should affect the price/demand of the other, and vice versa. Based on this notion we intend to test whether this effect exists in our data between salmon prices, cod prices and halibut prices.

2.2.2 Norwegian fishermen and the market for wild caught fish

To understand where our price data comes from we need to take a look at how wild caught fish are sold in Norway. Wild caught fish in Norway have a heavily monitored way to the consumer. There are laws concerning quality and handling that secures the fish are safe to eat. There are also laws that dictate the way these fish can be sold and who are liable throughout this process. The laws concerning the sales are governed by the act on first hand purchase of wild marine resource. A sales organisation act like a middleman in the transaction between fisherman and fish buyer. They can be compared in likeness to an auction house or exchange, where the fisherman pays a fee to access the exchange to sell their fish. But the fishermen remain as owner of the fish until the time of the transaction. This means that the sales organisation never actually own any of the fish. Through this sales process the Norwegian receipt for fishermen is created.(Nielsen, 2022)

2.2.3 Exchangeable goods

We want to explore the possibility that one might be able to use prices for similar commodities such as different seafood to help predict the price of salmon. To find the prices of these exchangeable goods we contacted Norsk Råfisklag. They where very helpful and provided us with historic prices data for halibut and cod. The data they provided was gathered using the Norwegian receipt for fishermen also known as a contract note. The receipt is a concept that is unique to Norway. The receipt contains both the quantity of fish caught as well as the sales value of these fish and a plethora of other information. It has a number of purposes such as paying the fishermen, creating the invoice for the buyer, controlling that the quota of fish is upheld, contributing to the Norwegian statistics

for wild caught fish, and more. We use data from these receipts to gather the week by week historic prices for halibut and cod. Norges Råfisklag, who provides the data, is an entity created by the industry to make sure it upholds its responsibility to the rest of the society. It is also responsible for securing a fair and transparent industry for workers and the environment. This makes it a great reliable source of data.(Harland, 2022)

2.2.4 Currency change

As mentioned, the salmon market is an export market. This in turn means that the prices are affected by the changes in between the NOK and the currency of the markets that it is exported to. To account for this, we have added Norges Bank's TWI to the dataset. "The TWI is a nominal effective krone exchange rate calculated on the basis of NOK exchange rates against the currencies of Norway's 25 main trading partners." (Bank, 2020) A rise in the value of the index indicates that the exchange rate of NOK depreciates.

2.3 Exploratory analysis

Exploratory data analysis (EDA) is a method used to analyse datasets and summarize the main characteristics. It helps determine how best to manipulate data sources to get the answers needed. This makes it easier to discover patterns, anomalies, test hypotheses or checking assumptions.

There are 4 steps involved in an exploratory data analysis: Data collection, data cleaning, univariate analysis and bivariate analysis. Data collection simply consists of gathering the data needed for the analysis. Data cleaning is the process of removing or replacing missing values, outliers, and other anomalies. Anomalies can disproportionally skew the data and hence adversely affect the results. (Biswal, 2023) Univariate analysis is analysis the data of only one variable. This is usually done by creating a histogram, boxplot or a frequency table. Bivariate analysis uses the data of two variables and compares them to each other. This establishes the correlation (or lack of) between the two variables. This is usually done by creating a correlation matrix or by plotting the data in a scatterplot.

2.4 Regression

Regression analysis is a reliable method of identifying which variables have impact on a topic of interest. Regression analysis consists of two types of variables: dependent and independent. The dependent variable is the main variable or factor that the model is trying to predict or understand. The independent variables are the factors that is hypothesized to have an impact on the chosen dependent variable.

Linear regression models often use a ”least-squares approach” to determine a line of best fit (regression line) to a given dataset. This line is the yellow line in the figure below. A square is the squared distance between a datapoint and the regression line. These values needs to be squared in order to not counteract each other.

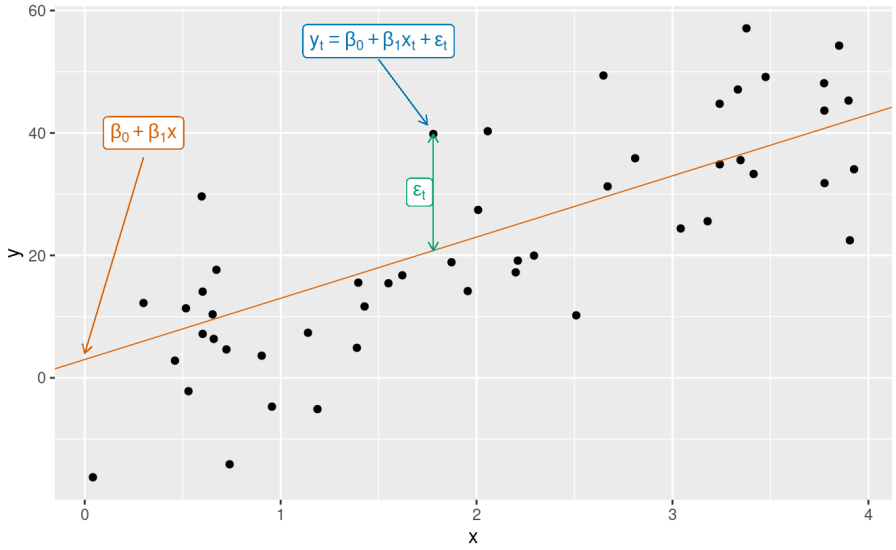


Figure 1: Depiction of a regression analysis. Hyndman and Athanasopoulos, 2021

When the process above has been completed a regression model is constructed. The general form of a multiple linear regression model is:

$$Y = a + b_1X_1 + b_2X_2 + b_3X_3 + ... + b_tX_t + u \tag{1}$$

Where Y is the dependent variable, X is the independent variable(s), a is the y-intercept, b is the slope of the explanatory variable(s) and u is the residual or error term.

2.5 ARIMA — SARIMAX

The ARIMA-model is one of the more popular and useful approaches to time series forecasting. The name is an acronym that stands for AutoRegressive Integrated Moving Average and the model utilizes these in order to predict future values solely on earlier values, it is therefore an univariate model. In order for the model to be able to predict, the data has to be stationary, which means that the mean, variance and covariance are constant over time. Time-series data is especially prone to be non-stationary, which can be solved either by log-transforming the data, convert it to a percentage change or by differencing which is further explained in 2.5.2.

The SARIMAX-model is an extension of the ARIMA-model that also takes external factors and seasonality into account in order to better predict future values. SARIMAX can therefore be a multivariate model. (Hyndman and Athanasopoulos, 2021)

2.5.1 Auto Regressive (AR)

The first part of the ARIMA acronym is the Auto Regressive part. Auto comes from the Greek word autos and mean self, in this context it means that the model is regressing on itself. This part of the model can be written as follows:

$$y_t = c + \phi_1 y_{t-1} + \phi_2 y_{t-2} + \cdots + \phi_p y_{t-p} + \varepsilon_t \tag{2}$$

Where c is a constant, p is the number of lag observations or autoregressive terms, ϕ are the AR coefficients and ε_t is the error term. y_t is the data on which the AR-model is applied on. (Oracle, 2023) This model is a "pure" AR-model and relies therefore solely on its own lags. If p is set to 1, the model looks at the previous value and tries to predict the next value. If p is set to 2, the model looks at the previous two values and tries to predict the next value, and so on. (Artley, 2022)

2.5.2 Integrated (I)

The second part of the ARIMA acronym is the Integrated part. This part of the model is used to make the time series stationary. In the ARIMA-equation it is represented by the letter d and is the number of differencing required to make the time series stationary. Usually, the optimal amount of differencing is the least amount needed to make the data fluctuate around a well-defined mean. (Nau, 2019) It is the only form of converting the data to a stationary form that the ARIMA-model itself can do. Log-transforming and converting to a percentage change can only be done before the model is applied on the data.

2.5.3 Moving Average (MA)

The third and last part of the ARIMA acronym is the Moving Average part. This incorporates the dependency of an observation on the residual errors from a moving average model applied to lagged observations. (Hayes, 2019) This part of the model can be written as follows:

$$\hat{y} = c + \theta_1 \varepsilon_{t-1} + \theta_2 \varepsilon_{t-2} + \cdots + \theta_q \varepsilon_{t-q} \tag{3}$$

Where c is a constant, q is the order of the moving average, i.e the number of lagged forecast errors. θ are the MA coefficients and ε_t is the error term. For example, if q is 1, the output relies solely on the errors from the previous time step. If q is 2, the output relies on the errors from the previous two-time steps. (Hyndman and Athanasopoulos, 2021)

Combining these three parts of the ARIMA-model, we get the following general forecasting equation:

$$\hat{y}_t = c + \phi_1 y_{t-1} + \phi_2 y_{t-2} + \cdots + \phi_p y_{t-p} + \theta_1 \varepsilon_{t-1} + \theta_2 \varepsilon_{t-2} + \cdots + \theta_q \varepsilon_{t-q} + \varepsilon_t \tag{4}$$

2.5.4 Seasonality (S)

For time-series data there can often incur a seasonal trend which is a pattern that repeats itself over a period of time. For example, pollution levels in a city that might increase in the

winter and decrease in the summer. This trend can be accounted for by adding a seasonal component to the model and where the ARIMA-model is written as ARIMA(p,d,q), the SARIMA-model is written as SARIMA$(p,d,q) \times (P,D,Q)s$. Uppercase P, D and Q corresponds to the lowercase p, d and q, namely the AR, I and MA terms, but for the seasonal component. The s is the number of periods in a season, for example if the data is measured daily, s would be 7 if the seasonal trend is weekly. (Chang et al., 2012)

2.5.5 Exogenous variables (X)

In addition to adding seasonality to the ARIMA model, one can also include exogenous variables and turn it into a multivariate model. This is often used when there is a clear correlation between the exogenous variable and the time series. For example, if the ARIMA model is used to predict the price of electricity, one could include weather data as one or several exogenous variables. (Elamin and Fukushige, 2018)

2.6 Neural networks

2.6.1 Recurrent Neural Network

A recurrent neural network (RNN) is a special type of artificial neural network adapted for work with time series data or data that involves sequences. (Saeed, 2021) By adding a ”feedback loop” RNNs are able to step through sequential input data while persisting the state of nodes in the hidden layer between steps. This is called ”working memory”. The hidden node concatenates its current input and the working memory from the previous steps before the result is passed on to an activation function. The output from the activation function is then sent onwards to both the output layer, and forwarded on to the next iteration of the RNN (as the working memory). (Bouvet, 2020)

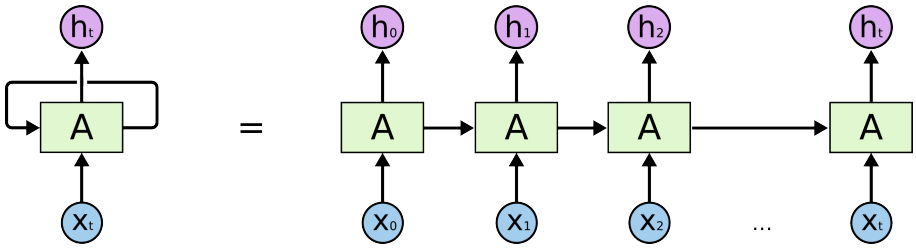


Figure 2: Depiction of an unrolled recurrent neural network. Khalajani, 2023

2.6.2 Long Short-Term Memory

Long Short-Term Memory (LSTM) is a network specifically designed to overcome the long-term dependency problem faced by RNNs. A typical LSTM network consists of three gates; the input gate, the forget gate and the output gate. These gates control how the information in a sequence of data comes into, is stored, and leaves the network.

The forget gate will decide which bits of the cell state are useful, given both the previous hidden state and the new input data. This is done by feeding the previous hidden state and the new input data into a neural network, which uses the sigmoid activation to output a number between 0 and 1, where 0 means ”forget”, and 1 means ”remember”. The output of the forget gate is then multiplied with the previous cell state. An output value of close to 0 means that the cell state will have less influence on the following steps.

The input gate and new memory network will determine what information should be added to the networks cell state, given the previous hidden state and new input data. The inputs in this gate are the same as in the forget gate. Here there is created a neural network with a tanh activation function, to combine the previous hidden state and the new input data and generate a ”new memory update vector”. This vector decides how much to update each component of the cell state of the network, given the data. Since the tanh activation function is used the outputs of this neural network will be between -1 and 1. This new memory vector does not actually take into account whether the input data is worth remembering. Therefore, the input gate is a sigmoid activated network which filters which components of the new memory vector are worth retaining. The output of the new memory vector and the input gate is then multiplied, and the combined vector is added to the cell state.

The output gate uses the newly updated cell state, the previous hidden state and the new input data to decide the new hidden state. The inputs in the output gate are the previous hidden state, and the new input data, and this is fed into a sigmoid activated neural network. This works as a filter, so that the output gate does not simply output everything it knows about something, and instead only outputs the most relevant information. Before this output can be used as the new hidden state, the cell state has to be passed through a tanh activation function, in order to force the values to be between -1 and 1 (so called "squished cell state"). When this is done the output from the output gate and the squished cell state are then multiplied. This outputs the new hidden state.

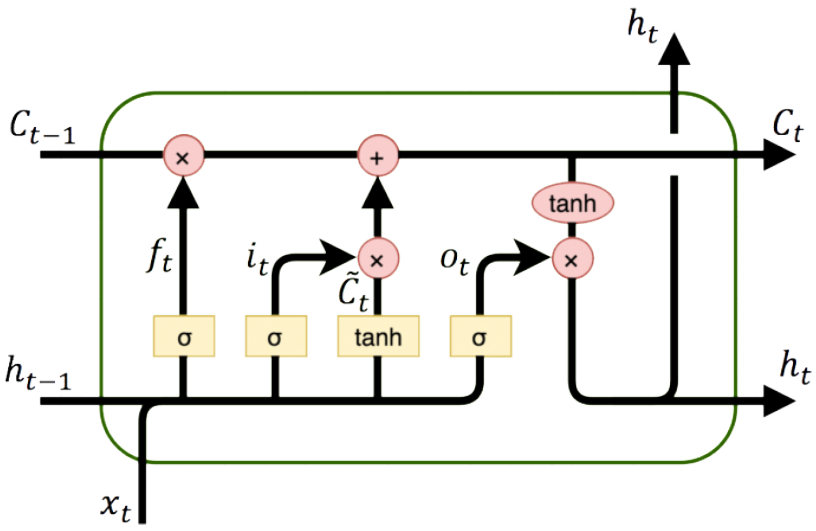


Figure 3: Depiction of a LSTM network. Dolphin, 2021

2.6.3 Tensorflow

Tensorflow is an open source library for numerical computation that makes machine learning and developing neural networks faster and easier. (Yegulalp, 2018) It works by allowing the user to define a graph of computations that are then executed by the library. The graph is defined by the user by creating nodes that represent mathematical operations, and edges that represent the multidimensional data arrays (tensors) communicated between them. The library then optimizes the graph for execution on a CPU or GPU.

2.6.4 Sigmoid function

The Sigmoid function is given as:

$$S(x) = \frac{e^x}{e^x + 1} \tag{5}$$

Since:

$$\lim_{x \to \infty} S(x) = \frac{\infty}{\infty + 1} \approx \frac{\infty}{\infty} = 1 \tag{6}$$

$$\lim_{x \to 0} S(x) = \frac{0}{0 + 1} = \frac{0}{1} = 0 \tag{7}$$

The Sigmoid function will only output values between 0 and 1.

2.6.5 Hyperbolic tangent function

The hyperbolic tangent function is given as:

$$S(x) = \frac{sinh(x)}{cosh(x)} = \frac{e^x - e^{-x}}{e^x + e^{-x}} \tag{8}$$

Since:

$$\lim_{x \to \infty} tanh(x) = \frac{\infty - 0}{\infty + 0} = \frac{\infty}{\infty} = 1 \tag{9}$$

$$\lim_{x \to -\infty} tanh(x) = \frac{0 - \infty}{0 + \infty} = \frac{-\infty}{\infty} = -1 \tag{10}$$

The hyperbolic tangent function will only output values between -1 and 1.

3 Methodology

3.1 Data gathering

In order to analyse the salmon price, we first need to gather this data on the salmon price. The main data point is the price of salmon. There are several sources for this data, but we utilized the data from the NASDAQ salmon exchange. The reason for this being a combination of the accessibility of the data, and the fact that the NASDAQ salmon exchange (NQSALMON) uses a weighted average for the salmon price, gathered from a spectrum of salmon exporters and it is therefore the best source of meaningful data. Another reason for using the NASDAQ salmon exchange is that the data is updated weekly with no missing values for the entire time frame. We downloaded data from March 2013 through December 2022, for a total of 507 data points. This was our base for the independent factors. The next step was to gather data from the other relevant factors for our analysis.

3.2 Descriptive analysis

3.2.1 Salmon Price

By looking at some simple descriptive statistics we will be able to decern if there are any changes needed to be done in the pre-processing stage. We will also examine if there are any trends or correlations that we will have to capture in order to produce the models. Let us begin by looking at our main variable, the salmon price.

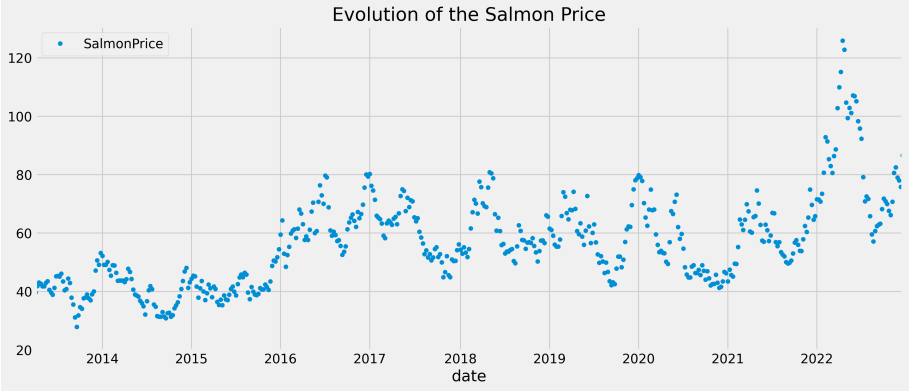


Figure 4: Salmon price evolution.

Looking at the plot we can see that the price has been increasing over the years. The top price having tripled our starting price, the mean of the price seems to have risen as well. Another increase we notice is the increasing magnitude of the price fluctuation. This is not only causing out highest prices but also some of the lowest prices in recent year. The plot is also showing the seasonal pattern that we expected to find in the price data where it is higher in the winter and lower in the summer. This is a characteristic we want our models to capture in order to make more precise predictions. To understand the seasonality of the price better we plot the prices for each month of the year.

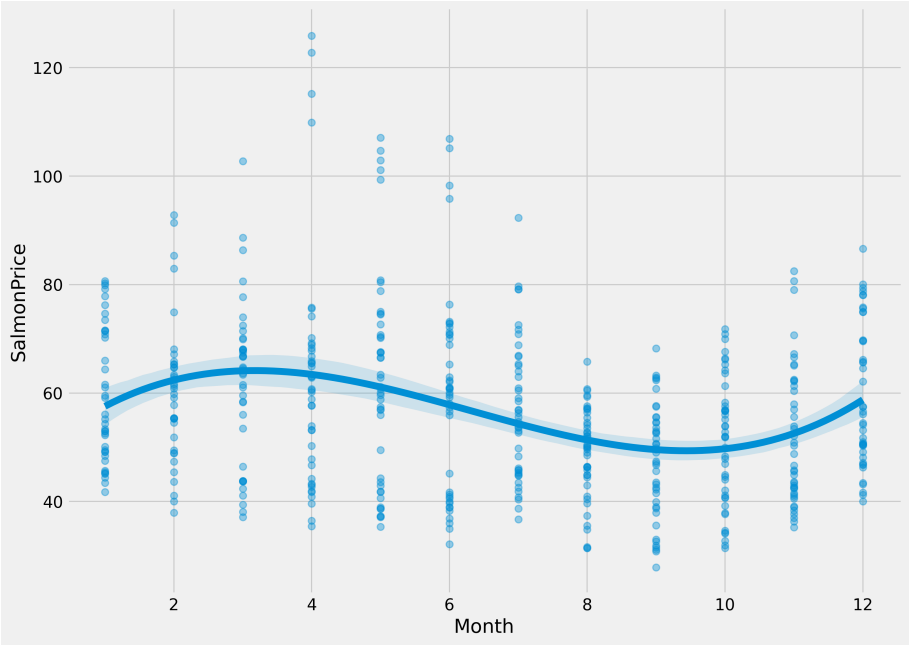


Figure 5: Seasonality of the salmon price

We see that there is a clear peak in the price in April. The price level in general is at

its highest in the spring and early summer and at its lowest in late summer and early fall. Looking at our regression line we see that this is true for most of the data, and not only the very top price points of each month. This line also visualizes the spread in price for the individual months and we can see that the higher spread in price follows the months of the highest prices. This might not be due to higher price differences during the spring. This characteristic is at least in part caused by the fact that the rising price level is amplifying the spring prices making the average of the earlier years a lot lower than the average of the later. This in turn makes it look like there are larger price differences during these months than what is actually the case.

3.2.2 Comparison between the different types of fish

We would also like to gain a little bit of insight as to how our variables compare to each other. We start by having a look at the price data for the different types of fish considering whether the cod and halibut is comparable to the salmon price.

	SalmonPrice	CodPrice	HalibutPrice
count	507.000000	507.000000	507.000000
mean	56.746134	24.553419	58.029858
std	15.455462	8.096777	7.900179
min	27.870000	10.275557	36.507234
25%	44.875000	20.365239	51.971903
50%	55.460000	24.035290	59.300626
75%	65.710000	30.478502	63.418690
max	125.870000	49.076318	79.182888

Table 1: Descriptive statistics for the fish price data

Looking at this we can see that the halibut is very similar to the salmon in price but with a larger minimum and a lower maximum as well as half the standard deviation. In general the halibut price seems to be much more stable. Looking at the cod price we see that its quite a bit lower compared to the two others. Relative to its size it has a lot more similar characteristics to the salmon price in terms of price stability. If we scale it by two, we will see that the mean price becomes 49 compared to the 55 of salmon and the standard deviation becomes 16 compared to the 15,5 of salmon. This indicates that the relative

swing in price of cod is of the same magnitude as the swing in price of salmon. However, the salmon price has a lot higher maximum price then the two others even if scaling the cod price by two, this may mean that we have some outliers at the very top of our salmon price range. To confirm this, we will look at a boxplot of the price data.

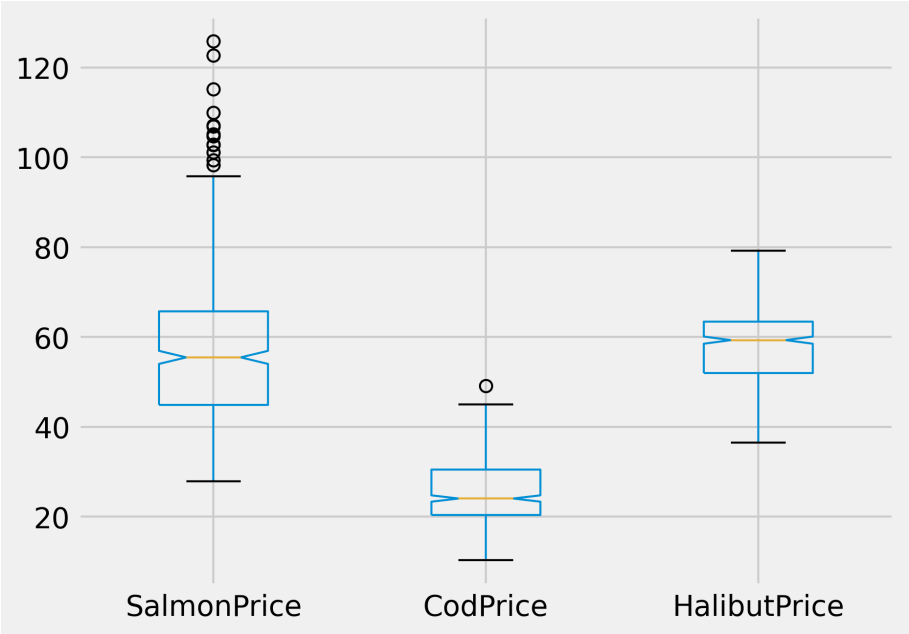


Figure 6: Whisker boxplot of fish price data.

Looking at the boxplot we can see that the salmon does indeed have a lot of outliers at the top of the price range. The cod price seem to have a singular outlier whilst the halibut price has none. The hight of the boxes also visualises the aforementioned price range differences well. Looking at the yellow lines representing the average. We can see that the cod and halibut have an average situated higher in middle of the 50th percentile. This means that the distribution of these two are a little bit skewed to the right.

3.2.3 Covariance and correlation

By exploring the covariance and correlation of our chosen variables we aim to uncover if they can be used as indicators for the salmon price. A good covariance between the salmon price and another variable, means that moving that variable would result in a likelihood of the other variable moving accordingly either positively or negatively. If the covariance is high, it is interesting to look at the correlation for the same variables and see

if we expect to see a change in the same direction for the corresponding variable. This property is what we are looking to exploit in our predictive models and finding variables that possess this property is key to the accuracy of our predictions.

From the matrices in figures 7 and 8 we see that all the variables except months have a pretty good covariance all being above 50. The best ones being the CPI and the Cod Price as we suspected earlier. The corresponding correlations of CPI and cod are also the highest correlations to our main variable which means that these two variables are more likely to be good indicators for the salmon price. The other variables have a lower covariance and correlation, but still have a positive covariance which means that they are still good candidates for our predictive models. The months variable has a negative covariance which means that the salmon price is more likely to go down in the winter months. This is a good indicator for the salmon price as we know that the salmon price has a seasonal tendency. Although a month term might not be the best way to capture it.

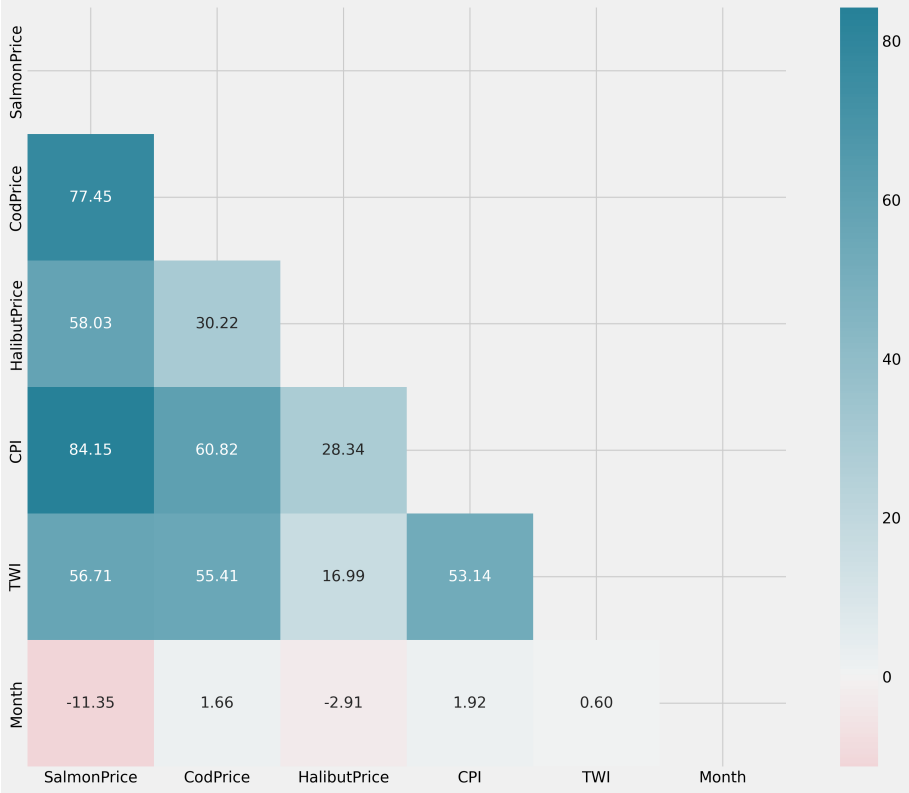


Figure 7: Covariance matrix with all variables

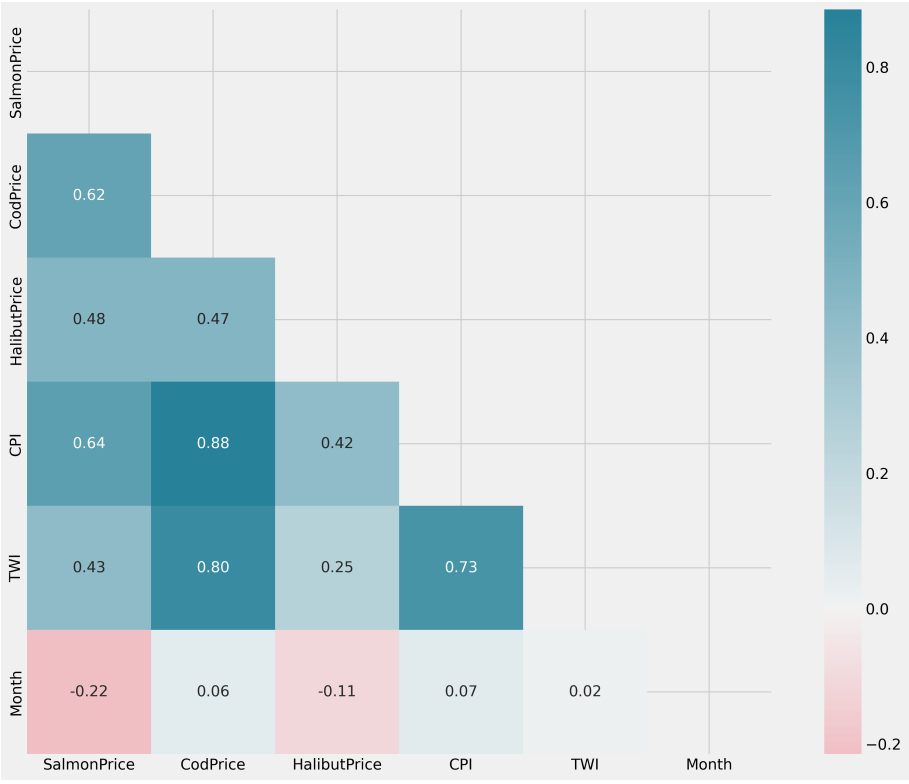


Figure 8: Correlation matrix with all variables

3.3 Data preprocessing

In order to use the data for prediction in our models we need to prepare the data. In our case this means to split the data into a training and test set. We will use the same split for all of our models, so that we they are comparable. The test set will be the last year of the data, 2021. The training set will be the rest of the data, 2013–2020. A very important aspect of forecasting using machine learning algorithms is that the test set should be 'invisible' throughout the training process, otherwise we might get data leakage and consequently, a false sense of how well the model performs. (Brownlee, 2016a)

3.4 ARIMA and SARIMAX

One important prerequisite for the ARIMA model is that the data is stationary. This can be done either by analysing the time-series itself and noticing the variance and trend, or by using the Augmented Dickey-Fuller test. After this is done, the next step will be to find the optimal parameters for the ARIMA model. This is usually done by looking at the ACF and PACF plots. When the optimal parameters are found, the model can be fitted and used to predict future values. (Hyndman and Athanasopoulos, 2021)

3.4.1 Determining stationarity

The Augmented Dickey-Fuller test is a statistical test that can be used to determine stationarity. The null hypothesis of the test is that the time-series is non-stationary. If the p-value is less than the significance level, the null hypothesis is rejected and the time-series is expected to be stationary. Using the Augmented Dickey-Fuller test on the Salmon Price data, we get a p-value of 0.02406, this means that we can reject the null hypothesis at a 95% confidence level according to this test. The data may therefore be stationary. (Dickey and Fuller, 1979)

In order to better understand why the data is not stationary we can plot the data and look at the variance, trend and seasonality. This is done by decomposing the data into its components. We then get the following plot:

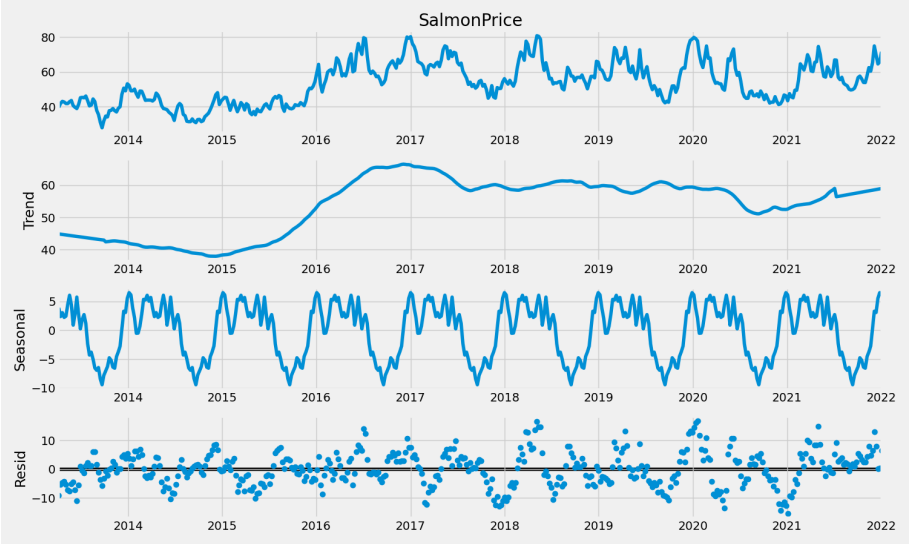


Figure 9: Decomposition of the Salmon Price data.

Examining the decomposed data in Figure 9 there are especially two things that stand out. The first is the trend, which is clearly increasing. The second is the seasonality. There is a clear yearly seasonal trend where the price is higher in the summer months before decreasing during the autumn and reaching a low in the winter. From this we draw a different conclusion than the Augmented Dickey-Fuller test. According to the plot, the data is not stationary and needs differencing. In the ARIMA model, this will be done by setting d to 1 or more.

The exact number of differencing needed can be found either by using the ADF-test on the differenced data and looking for when the p-value is less than the critical value, or by looking at the autocorrelation plot and using rules set out by Nau (2019) to determine the number of differencing needed.

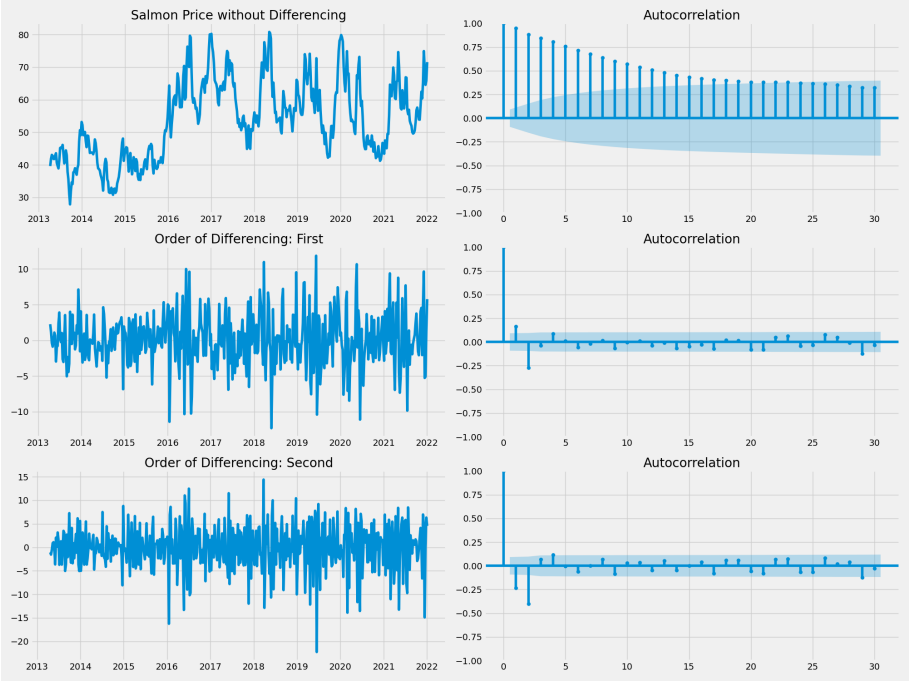


Figure 10: Different orders of differencing.

Examining the trend and autocorrelation in Figure 10 we can see that the original data without differencing has both a clear trend and a slow decay in the autocorrelation plot with a high number of positive lags. Following the first rule from Nau (2019) we can conclude that the data needs at least one order of differencing.

After just a single order of differencing, the trends start to flatten out and fluctuate around 0. The autocorrelation plot also drops sharply after the first lag, and is then quite small and patternless, this follows the second rule from Nau (2019) and will often be a sign that higher differencing is not needed. Running the Augmented Dickey-Fuller test on the differenced data, we get a p-value of 3.86388e-24, much lower than the critical value of 0.05. We can therefore reject the null hypothesis and conclude that the differenced data is stationary.

With a second order of differencing, the trend seems to flatten even more, and the autocorrelation plot shows a sharp negative decline for the first and second lag, this may be a sign that the data is over-differenced. This is not optimal as over-differencing will lead to a loss of historical information and trends. It is therefore of utmost importance to find the order of differencing that both makes the data stationary and keeps the historical memory intact. Over-differencing is a common mistake when fitting non-stationary data to ma-

chine learning models, and can lead to the model not being able to capture the underlying trend. (Prado, 2018)

A third way to determine the optimal number of differencing is, according to Nau (2019), to look at the standard deviation of the plot at different orders of differencing. Following his rule number 3, the optimal number of differencing is the one where the standard deviation of the plot is the lowest. Examining Table 2 we can see that the standard deviation of the data is lowest at the first order of differencing. After this the standard deviation increases, and there is no reason to believe any higher number of differencing will reduce the standard deviation.

	Standard deviation
No differencing	11.989141
First differencing	3.708461
Second differencing	5.661965
Third differencing	6.548135

Table 2: Standard deviation of the differenced data.

We can therefore conclude that the optimal number of differencing should be either 0 or 1.

3.4.2 Autoregression and Moving Average

The next step in a regular ARIMA model is to identify the optimal AR and MA terms needed to fit the model. This can be done by comparing different models by looking at the AIC and BIC values, but with larger models this would constitute an unnecessary use of computational power. A more efficient way to find the optimal terms is to look at the ACF and PACF plots of the data and use the rules set out by Nau (2019) to determine the optimal p and q.

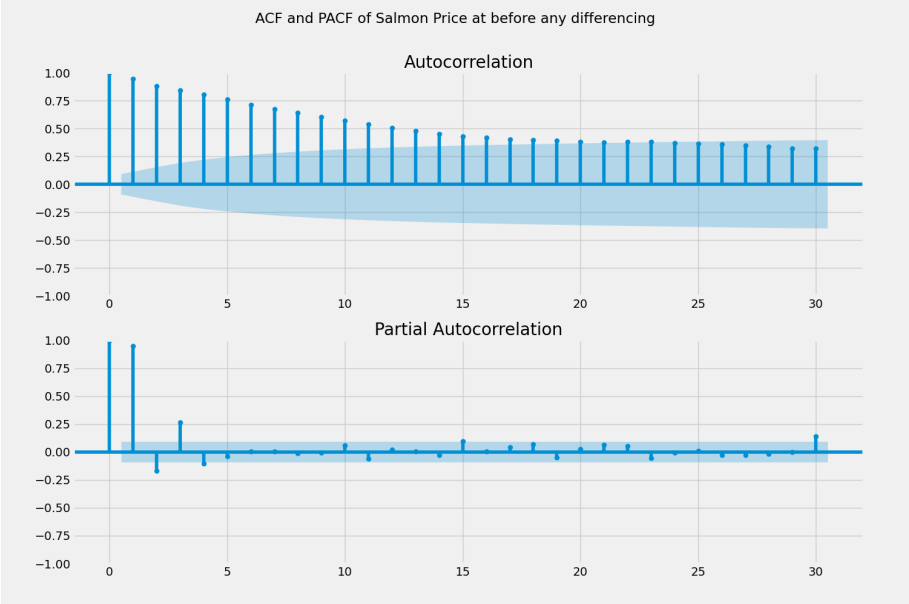


Figure 11: ACF and PACF plots of Salmon price.

The most noticeable part of the ACF plot in Figure 11 is the slow decline of the autocorrelation after the first lag. This is a sign that the data is not white noise, and that there is a correlation between the data and its lags. According to Nau (2019), this is called an 'AR signature' and is a clear sign that we need to add an AR term to the model, rather than adding an MA term. To find the optimal amount of AR terms needed we can look at the PACF plot. As a rule of thumb, the optimal number is where the PACF plot exhibits a clear drop. In Figure 11 we can see that the PACF plot drops sharply after the lag 1, and then fluctuates around 0, mostly within the confidence interval. Therefore, without any differencing, the optimal number of AR terms is 1. But, as we concluded in 3.4.1, the data needs at least one order of differencing. We therefore need to look at the ACF and PACF plots of the differenced data.

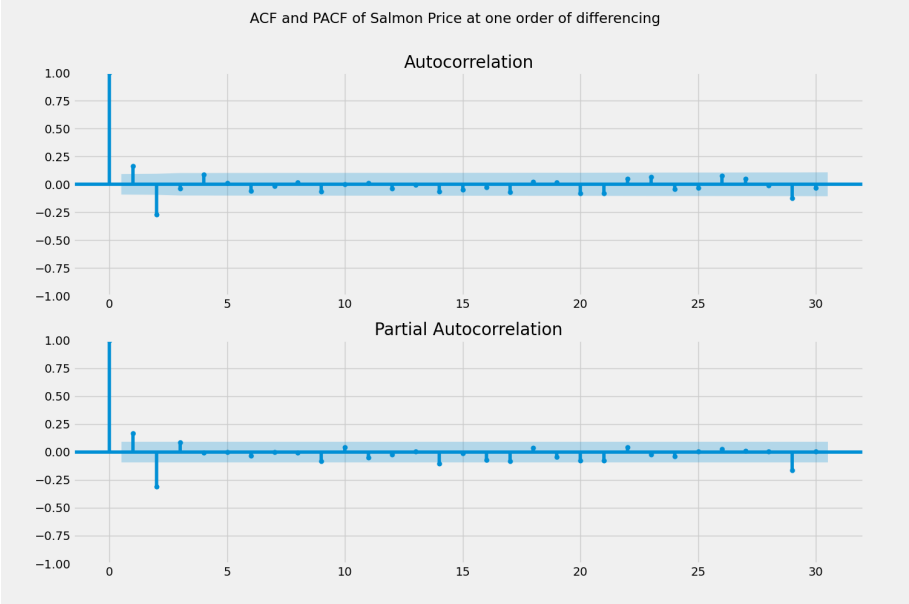


Figure 12: ACF and PACF plots of differenced Salmon price.

Examining the ACF plot in Figure 12 we can see that the autocorrelation now has a much more rapid decline compared to the original data. The PACF plot exhibits some of the same characteristics as the original data, but with a much sharper decline for the first lag. Nevertheless, as the first lag of both the ACF and PACF plot is positive and significant outside the 95% confidence interval, there should be at least one AR term. At the same time, there is also some negative lags in both the ACF and PACF plots which may indicate that there should be an MA term, but according to Nau (2019), we should be careful mixing AR and MA terms in ARIMA models, as they may cancel each other out. The best model will, in most cases, consist solely of either AR or MA terms.

As the interpretation of visual plots and clues from these can be a bit subjective, we should employ some objective test to determine the optimal number of p and q. The most straight forward way to do this is to perform an iterative search over the possible values of p and q and then compare the AIC and BIC values of the different models, as well as comparing the mean squared error of the predictions from the models.

3.4.3 Seasonality

Hitherto we have only considered the non-seasonal ARIMA model, but as with many time series, the Salmon price might also exhibit some seasonal patterns and trends. It may

therefore be better to add a seasonal part to the model and consequently call it a SARIMA model. One way to determine this is to look at the seasonal part of the decomposed data. The decomposed data shows us what part of the variation in the data is due to the trend, the seasonal part, and the residual part.

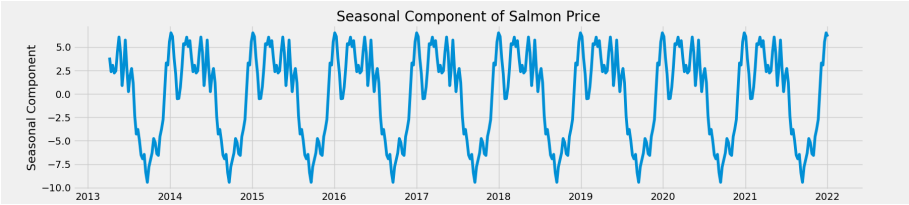


Figure 13: Seasonal part of the decomposed data

Examining Figure 13 we can see a clear seasonal trend which seems to be repeating itself every 12 months. This could indicate that we should add a seasonal part to the model. As the data is recorded weekly, the s should be 52. Hyndman and Athanasopoulos (2021) mentions that working with weekly data can prove difficult for the ARIMA model to handle as there are, on average, 52.18 weeks in a year, which is not an integer. Nevertheless, we will try to fit a SARIMA model with $s = 52$ and compare the AIC and mean squared error of the predictions to the non-seasonal ARIMA model.

The seasonal part of the ARIMA model also contains PDQ which corresponds to the pdq in the non-seasonal ARIMA model in that they describe the number of seasonal autoregressive terms, the number of seasonal differences, and the number of seasonal moving average terms. In order to determine these terms, we can use the same method as we did for the non-seasonal ARIMA model, namely examine the ACF and PACF plots, but this time for the seasonally differenced data. We can also perform a grid search for the different combinations of AR and MA terms, and in that way obtain the optimal model.

3.4.4 Exogenous Variables

One last way to expand on the ARIMA model in order to make it more accurate is to include one or more exogenous variables. This is especially useful if there is some external data that might affect the time series. In our case we suspected that the price of Salmon might be affected by the price of other seafood, as well as the Norwegian Krone. We

therefore decided to add these to our dataset and see if it would improve the accuracy of the model.

In order to measure the accuracy of these exogenous variables, we can follow the same procedure as before and compare both the AIC of different models as well as the root mean squared error of the predictions. Comparing the RMSE will probably prove to be a more accurate way of examining the accuracy of the models as AIC cannot handle different levels of differencing and will not work well to compare ARIMA with Tensorflow. (Brownlee, 2017)

3.5 LSTM — Tensorflow

In order to apply the data to a LSTM model, we need to define the different parameters of the model. As a LSTM model consists of an input layer, a hidden layer, and an output layer, we need to define the number of neurons in each layer. And as can be seen in Figure 14, all neurons between the layers has to be connected, which makes the complexity of the model grow exponentially. Therefore, we need to be careful not to make the model too complex, as it will use too much unnecessary computing power, as well as increasing the risk of overfitting.

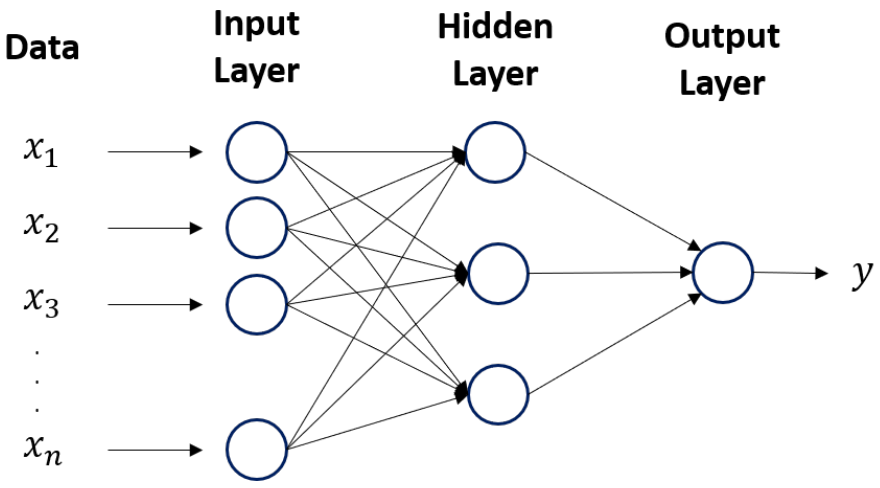


Figure 14: Example of the different layers in an LSTM model. TowardsAI, 2020.

3.5.1 The input layer

The input layer is the first layer in the model, and as the name suggests, it is where the input data is fed into the model. The number of input nodes corresponds to the number of features in our model. Therefore, it will depend on the number of exogenous variables we choose to include in the model. We can, for example, choose to only train the model on the previous price of Salmon, and have only one input neuron, or we can include such variables as the price of other seafood and the Norwegian Krone, and have three input neurons.

In order to prepare the data we will need to scale it between 0 and 1, technically this is not necessary for the LSTM model to work, but it will make the training process both faster and more accurate. (Brownlee, 2019) Secondly, for time series data, it can be helpful to reshape the data into several timesteps, for example, we can use the previous 10 weeks of data to predict the next week. This is expected to make the model more accurate, as it will be able to learn the patterns in the data more efficiently. The reshaped data will be a 3-dimensional array, where the first dimension is the number of samples, the second dimension is the number of timesteps, and the third dimension is the number of features. In mathematics, this sort of array is called a tensor; hence the name Tensorflow. (TensorFlow, 2023)

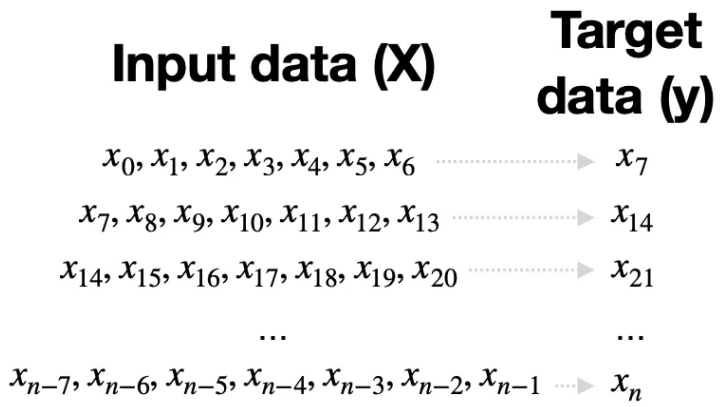


Figure 15: Example of sequential data in timesteps. Dobilas, 2022.

3.5.2 The hidden layer

The hidden layer is the layer in the model that is responsible for the actual learning, and while the input layer is quite straightforward as the only variables are the number of timesteps and features, the hidden layer is more complicated. This layer consists primarily of multiple interconnected neurons, these neurons can themselves be arranged in layers and result in several hidden layers. The neurons are assigned weights and biases which are used to calculate the output, the weights are used to determine the importance of the input while biases are used to regulate at what point the weight should be added to the output. The weights and biases are updated during the training process in order to make the model as accurate as possible, it achieves this by reducing the loss function, such as mean squared error. In addition to specifying the number of neurons in the hidden layer, we also need to specify the activation function, which is used to determine when the neuron should be activated. The most common activation functions are the sigmoid function, the hyperbolic tangent function, and the rectified linear unit function. (Sharma, 2019)

Consequently, there are a great number of hyperparameters that needs to be chosen before the model can be fitted, and while the goal is to find the optimal model, we also need to avoid overfitting. One way to achieve this is to solely experiment with different hyperparameters and see which combination gives the best results, but this can be a time-consuming process. A more efficient way is to utilize grid search, similarly to the ARIMA model, which will test for different models automatically and return the model with the best results according to our chosen metric. (Brownlee, 2016b) To avoid overfitting the model it can be useful to implement a regularization technique, such as a dropout layer which will randomly break some synapsis between the neurons, or an early stopping criterium which will stop the training process when the model stops improving. (Srivastava et al., 2014)

3.5.3 The output layer

The output layer is the last layer in the model, and as the name suggests, it is where the output is generated. This layer consists of a single neuron which will output the predicted

value of the Salmon price.

In conclusion, we will choose an appropriate number of timesteps for the input tensor, and then use grid search to find the optimal model. The hyperparameters we will evaluate are the number of neurons in the hidden layer, the activation function, the number of epochs, the batch size, the dropout rate, and the optimization algorithm. In order to evaluate the model, we will use the root mean squared error, which is the same metric we used for the ARIMA model.

3.5.4 Implementation

Finding the optimal neural network can be a complex and time-consuming task, we therefore constructed a nested for-loop with different hyperparameters. A network like this can be connected with infinite amount of params, depending on the amount of layers and neurons, to reduce the complexity and run-time, we opted for a simpler model with only 97 params and three layers. During the research period, we did some limited testing with more complex models, but did not find any improvements over the simpler ones.

At the first layer we experimented with different options, but ended up with a flatten layer, this is used to reduce the dimensionality of the input data and make it easier for the model to learn. In order to keep the model as simple, we used a single hidden dense layer with 32 neurons and a relu activation function. The output layer is a dense layer with a single neuron.

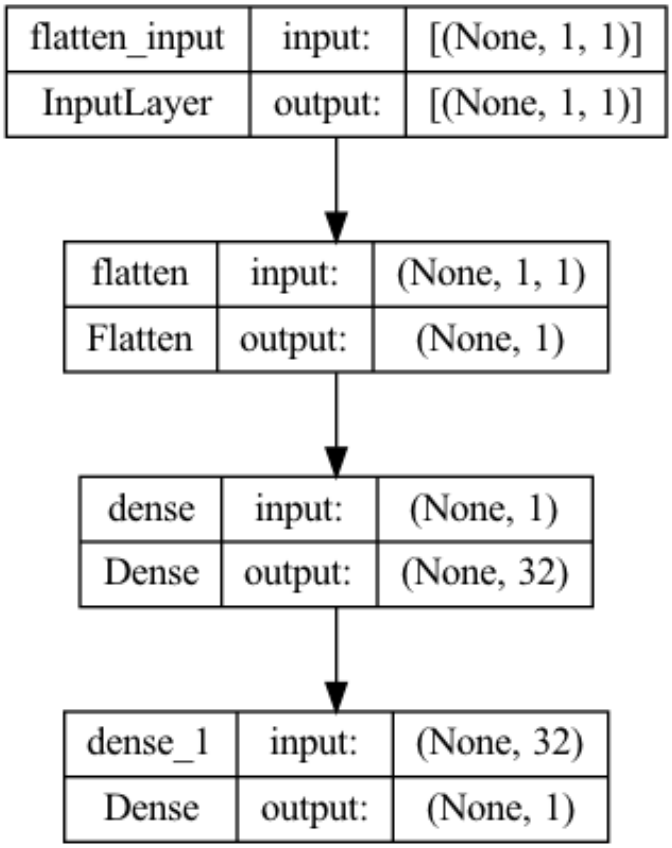


Figure 16: Model architecture.

In order to test for the different possible models, we decided to have the batch size, the number of epochs, the number of timesteps and different optimizers as variables in our loop. In addition we tested for both univariate and multivariate models. The batch size is the number of samples that will be propagated through the network, we choose to test for batch sizes between 1 and 104. The number of epochs is the amount of times the model will be trained for each model, we here chose epochs between 10 and 100. The number of time steps we chose to use is a week, 4 weeks, 52 weeks and 104 week. The main reason for this is the hypothesis that the model will perform better if it is able to take the seasonality into account. The optimizers we tested for are Adam and nadam, two of the more accurate optimizers. (Zaman et al., 2021) In conclusion, this resulted in 288 different models.

4 Results and discussion

The following chapter will discuss the results of the models presented in section 3 before comparing the different models and discussing the results. We will start by examining the different ARIMA models, from the simplest ARIMA to the more complex SARIMAX. We will then compare this to the LSTM model.

4.1 ARIMA

As explained in section 3.4, determining the order of the ARIMA model can be a difficult and complex task, examining the ACF and PACF plots can be a good starting point, but will also be a bit subjective. The only objective way to optimize the model is to use a loss function. In our case we have chosen root mean squared error as this is a commonly used way to measure the accuracy of a model.

In order to find the model with the least RMSE, we performed a grid search on the different parameters using a nested for-loop and outputting the results in the following table:

	RMSE
ARIMA(0,1,1)	20.113051
ARIMA(1,1,0)	20.451239
ARIMA(1,1,1)	20.459960
ARIMA(0,1,0)	21.098688
ARIMA(0,1,2)	21.199198
ARIMA(1,1,4)	21.335281
ARIMA(1,1,2)	21.343223
ARIMA(0,1,4)	21.345766
ARIMA(1,1,3)	21.359142
ARIMA(0,1,3)	21.410871

Table 3: Results of the grid search for the ARIMA model.

Examining table 3 we can see that the ARIMA(0,1,1) has the lowest RMSE and is therefore expected to be the best model, but there is no significant difference between the models. This may indicate that the model is not able to capture the trends. An ARIMA model with p of 0, d of 1 and q of 1 is, according to Nau (2019), a simple exponential

smoothing model which indicates that it may capture the moving average trend, but no other trends. Fitting the model to the data and plotting the results against the actual values we get the following plot:

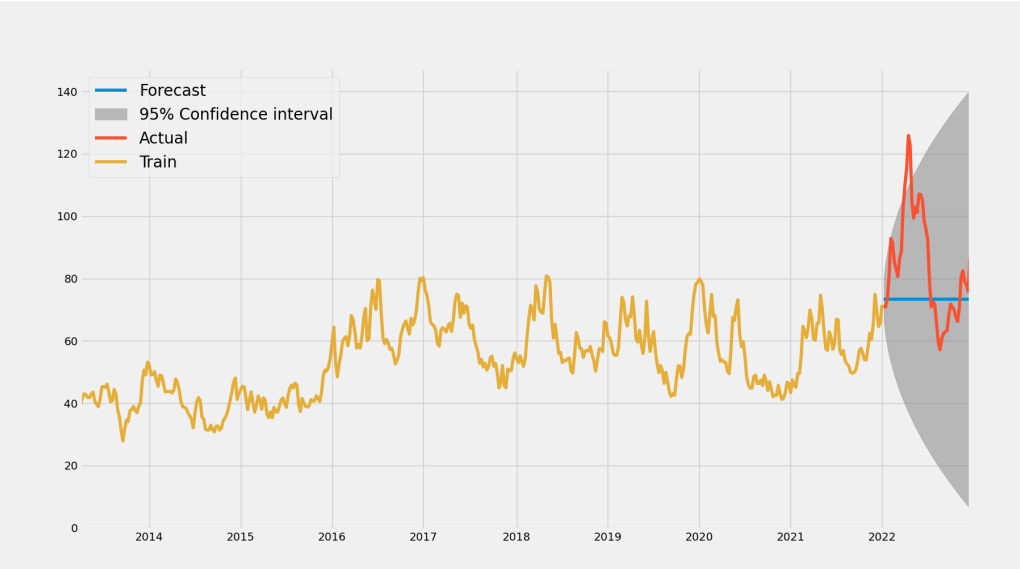


Figure 17: ARIMA(0,1,1) model fitted to the data.

As we can see, the model was not able to capture the trend, it seems that it was only able to take the average and draw it further out. In addition, the 95% confidence interval is very wide, indicating that the model is less than accurate.

Since this was the ARIMA model with the lowest RMSE, there is no reason to believe that the other ARIMA models will be any more accurate. We will therefore not examine the other ARIMA models, but instead move on to the SARIMA models.

4.2 SARIMA

One of the more prominent features of the Salmon data is the clear seasonality that is exhibited on a yearly basis. We should therefore expect a seasonal model to better be able to capture this trend. As we did with the ARIMA models, we will perform a grid search on the different parameters using a nested for-loop, the problem with this is that the SARIMA model has 6 parameters instead of 3. The search will therefore grow exponentially. Consequently, we decided to solely use a P of 0, D of 1 and Q of 0 as a larger P

and Q seemed to have a negative effect on the RMSE. As the data has a seasonality of 52 weeks, this is the seasonal parameter we will use. The ten best results of the grid search are presented in the following table:

	RMSE
SARIMA(2,1,0)(0,1,0,52)	14.829437
SARIMA(3,1,4)(0,1,0,52)	14.833213
SARIMA(3,1,0)(0,1,0,52)	14.833392
SARIMA(3,1,2)(0,1,0,52)	14.837780
SARIMA(3,1,1)(0,1,0,52)	14.839969
SARIMA(4,1,0)(0,1,0,52)	14.840067
SARIMA(4,1,2)(0,1,0,52)	14.840717
SARIMA(2,1,2)(0,1,0,52)	14.840721
SARIMA(3,1,3)(0,1,0,52)	14.840871
SARIMA(4,1,1)(0,1,0,52)	14.840890

Table 4: Results of the grid search for the SARIMA model.

Similarly, to the ARIMA model, there is not a significant difference between the models, but the RMSE is clearly lower for the SARIMA than the ARIMA, this indicates the importance of capturing the seasonal trend in the dataset. The SARIMA(2,1,0)(0,1,0)[52] model has the lowest RMSE and should therefore be the most accurate SARIMA model. After fitting the model on the train data and comparing the predictions against the actual data we get the following plot:

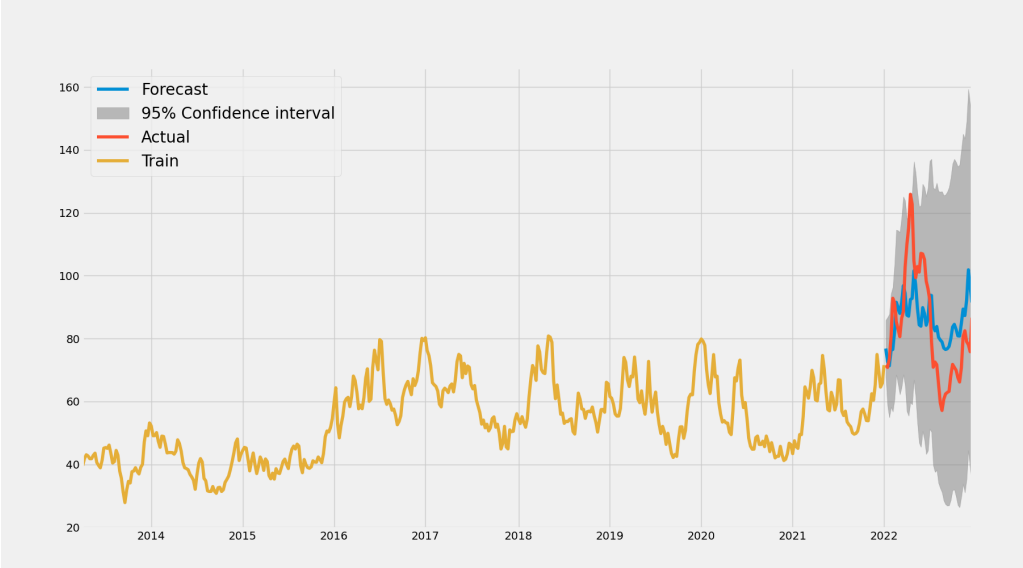


Figure 18: SARIMA(2,1,0)(0,1,0,52) model fitted to the data.

As we can see in figure 18, and as predicted from the RMSE, the SARIMA model is able predict the data much closer to the actual data than the ARIMA model. The 95% confidence interval is still quite large, especially when we reach the end of 2022, but as we can see from the large spike in the spring of 2022, this interval is necessary to be accurate. We can also take a closer look at the twenty first predictions for the year 2022:

	Actual	Predicted	Difference
2022-04-17	125.870000	92.412135	-33.457865
2022-04-24	122.750000	92.791669	-29.958331
2022-04-10	115.170000	87.122430	-28.047570
2022-12-04	78.050000	101.861789	23.811789
2022-05-29	107.090000	83.941779	-23.148221
2022-04-03	109.910000	87.461039	-22.448961
2022-08-21	57.150000	78.941789	21.791789
2022-12-11	75.830000	96.591789	20.761789
2022-08-14	59.560000	79.561789	20.001789
2022-06-12	105.150000	87.961792	-17.188208
2022-06-05	106.900000	89.841786	-17.058214
2022-05-22	101.130000	84.361794	-16.768206
2022-02-06	92.820000	76.549877	-16.270123
2022-08-28	60.740000	76.991789	16.251789
2022-08-07	65.780000	80.431789	14.651789
2022-10-30	66.200000	80.791789	14.591789
2022-07-10	79.140000	93.701789	14.561789
2022-09-18	63.230000	77.501789	14.271789
2022-11-06	70.700000	84.841789	14.141789
2022-09-04	62.400000	76.531789	14.131789

Table 5: SARIMA(2,1,0)(0,1,0,52) model predictions for the year 2022.

Sorting the difference between actual and predicted values in ascending order we can see that the model does have some large errors, especially in the spring of 2022 with the largest difference being -33.5. We can also take a look at the actual model with the different parameters:

We can in table 6 see our two autoregressive terms, ar.L1 and ar.L2, and the error term sigma2. The important thing to note from the table is that both the autoregressive terms have a p-value of 0.000, which means that they are both statistically significant. Another important conclusion to draw from the SARIMA results is whether or not the residuals are independent, or white noise. Examining the Ljung-Box test, we see that it produces

Dep. Variable:	SalmonPrice	**No. Observations:**	457
Model:	SARIMA(2, 1, 0)x(0, 1, 0, 52)	**Log Likelihood**	-1190.4
Date:	Fri, 21 Apr 2023	**AIC**	2386.72
Time:	14:19:24	**BIC**	2398.73
Sample:	04-07-2013 - 01-02-2022	**HQIC**	2391.48
Covariance Type:	opg		

	coef	**std err**	**z**	**P> |z|**	**[0.025**	**0.975]**
ar.L1	0.1953	0.044	4.486	0.000	0.110	0.281
ar.L2	-0.2932	0.043	-6.895	0.000	-0.376	-0.210
sigma2	21.2106	1.328	15.966	0.000	18.607	23.814

Ljung-Box (L1) (Q):	0.14	**Jarque-Bera (JB):**	6.15
Prob(Q):	0.71	**Prob(JB):**	0.05
Heteroskedasticity (H):	2.14	**Skew:**	-0.14
Prob(H) (two-sided):	0.00	**Kurtosis:**	3.54

Table 6: SARIMA(2, 1, 0)x(0, 1, 0, 52) Results

a result with a p-value of 0.71, this is far greater than the critical value of 0.05. This means that we cannot reject the null hypothesis that the residuals are independent, and there could therefore be more information in the residuals that the model was not able to capture.

While the AIC and BIC values both can be important to when comparing models, the change in differencing and seasonality makes it difficult to compare the models purely based on this criterion. This is part of the reason why we chose to use the RMSE as our main criterion for comparing the models.

4.3 SARIMAX

The final way to improve upon our ARIMA model is to include exogenous variables. We will also here utilize a grid search on the different parameters in order to optimize the model. To reduce the number of iterations we will assume that the optimal parameters from the SARIMA model are still the optimal parameters for the SARIMAX model. Further, we have the choice between having all exogenous variables in the model, only a few, or just a single variable. We will start by including all exogenous variables in the model, and then gradually reduce the number of exogenous variables. As there is only one pos-

sible model when including all models, we will not use a grid search, but instead just fit the model, we then get the following results:

	RMSE
SARIMAX(2,1,0)(0,1,0,52)(CodPrice+HalibutPrice+CPI+TWI)	15.809896

Table 7: Result from SARIMAX model with all exogenous variables.

As we can see from table 7 the RMSE did increase from table 4, this probably means that the exogenous variables do not increase the accuracy of the model. This might be because there are conflicting trends in the exogenous variables that cancel each other out. A better way to increase accuracy might therefore be to drop some of the exogenous variables and instead include the ones that reduces the RMSE the most. Iterating over all possible combinations of two exogenous variables we get the following results:

	RMSE
SARIMAX(2,1,0)(0,1,0,52)(HalibutPrice+TWI)	14.575270
SARIMAX(2,1,0)(0,1,0,52)(TWI+HalibutPrice)	14.575270
SARIMAX(2,1,0)(0,1,0,52)(TWI+CodPrice)	14.705717
SARIMAX(2,1,0)(0,1,0,52)(CodPrice+TWI)	14.705717
SARIMAX(2,1,0)(0,1,0,52)(HalibutPrice+CodPrice)	14.928625
SARIMAX(2,1,0)(0,1,0,52)(CodPrice+HalibutPrice)	14.928625
SARIMAX(2,1,0)(0,1,0,52)(TWI+CPI)	15.962664
SARIMAX(2,1,0)(0,1,0,52)(CPI+TWI)	15.962665
SARIMAX(2,1,0)(0,1,0,52)(CPI+CodPrice)	16.593182
SARIMAX(2,1,0)(0,1,0,52)(CodPrice+CPI)	16.593207
SARIMAX(2,1,0)(0,1,0,52)(CPI+HalibutPrice)	16.607369
SARIMAX(2,1,0)(0,1,0,52)(HalibutPrice+CPI)	16.607373

Table 8: Result from SARIMAX model with two exogenous variables.

Examining the RMSE we can now see that reducing the number of variables had a positive effect on the accuracy of the model. We archived the best results when including HalibutPrice and TWI as exogenous variables. This might indicate that the CodPrice has a negative effect on the model. In addition, this SARIMAX model is more accurate than the SARIMA model, which indicates that the inclusion of exogenous variables can improve the accuracy.

Still, we can reduce the number of exogenous variables to just one, in order possibly improve the accuracy even further. Utilizing a grid search we get the following results:

	RMSE
SARIMAX(2,1,0)(0,1,0,52)(TWI)	14.529888
SARIMAX(2,1,0)(0,1,0,52)(HalibutPrice)	14.857380
SARIMAX(2,1,0)(0,1,0,52)(CodPrice)	14.877221
SARIMAX(2,1,0)(0,1,0,52)(CPI)	16.654635

Table 9: Result from SARIMAX model with one exogenous variable.

As we can see, the RMSE did improve. More interestingly, it seems that the only exogenous variable that improves the accuracy of the model is the TWI, as both HalibutPrice, CodPrice and CPI has a negative effect on the model. To examine the results further we can take a look at the summary of the model:

Dep. Variable:	SalmonPrice	**No. Observations:**	457
Model:	SARIMAX(2, 1, 0)x(0, 1, 0, 52)	**Log Likelihood**	-1185.0
Date:	Sun, 23 Apr 2023	**AIC**	2378.09
Time:	17:10:51	**BIC**	2394.09
Sample:	04-07-2013 - 01-02-2022	**HQIC**	2384.42
Covariance Type:	opg		

	coef	**std err**	**z**	**P> |z|**	**[0.025**	**0.975]**
TWI	-0.5192	0.146	-3.561	0.000	-0.805	-0.233
ar.L1	0.1774	0.043	4.087	0.000	0.092	0.263
ar.L2	-0.3044	0.044	-6.850	0.000	-0.391	-0.217
sigma2	20.6612	1.320	15.650	0.000	18.074	23.249

Ljung-Box (L1) (Q):	0.05	**Jarque-Bera (JB):**	5.29
Prob(Q):	0.82	**Prob(JB):**	0.07
Heteroskedasticity (H):	2.04	**Skew:**	-0.17
Prob(H) (two-sided):	0.00	**Kurtosis:**	3.45

Table 10: SARIMAX(2, 1, 0)x(0, 1, 0, 52) Results

Compared to table 6, we can see that both the AIC and BIC have been reduced. This does indicate that the model is more accurate. However, the p-value of the Ljung-Box test is still above the critical value of 0.05 which indicates that the residuals are not white noise and that there might be some trends left in the residuals that the model has not captured.

Finally, we can plot the results from the best SARIMA and SARIMAX model against the actual data:

Figure 19: SARIMAX model with TWI as exogenous variable compared with SARIMA.

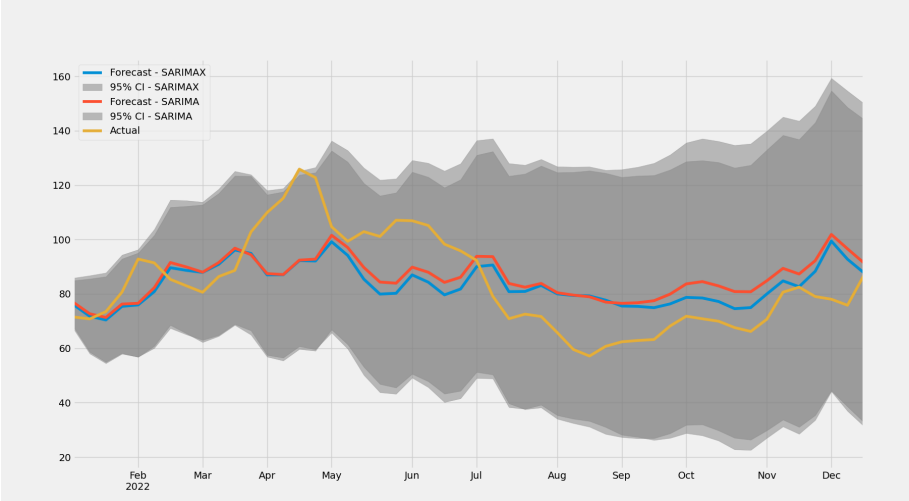


Figure 20: Predictions from figure 19.

Comparing these predictions we find no discernible difference between the two models. This is not surprising with a difference in RMSE of only 0.3. The reason why the SARIMAX model is somewhat more accurate may be that it throughout predicts a lower price than the SARIMA model.

4.4 LSTM

In order to get a meaningful result from the different models in the grid search, the RMSE was calculated for each model, and returned to the following dataframe:

Run	Batch Size	Epochs	Model Name	Optimizer	Time steps	Uni/Multi	RMSE
1	1	10	Model 1	adam	1	Univariate	37.942711
2	1	10	Model 2	adam	1	Multivariate	31.080595
3	1	40	Model 3	adam	1	Univariate	31.504920
4	1	40	Model 4	adam	1	Multivariate	27.286080
5	1	100	Model 5	adam	1	Univariate	33.270839
6	1	100	Model 6	adam	1	Multivariate	30.467997
7	1	10	Model 7	adam	4	Univariate	33.727742
8	1	10	Model 8	adam	4	Multivariate	27.184055
9	1	40	Model 9	adam	4	Univariate	30.977471
10	1	40	Model 10	adam	4	Multivariate	20.755111

Table 11: RMSE for each model in the grid search.

These first runs are not particularly accurate, at least compared to the best SARIMAX models. There are also no clear difference between the different parameters. A more efficient way to find the best models is sorting after the RMSE. This is done in the following table:

Run	Batch Size	Epochs	Model Name	Optimizer	Time steps	Uni/Multi	RMSE
186	26	100	Model 186	nadam	52	Multivariate	14.870351
72	2	100	Model 72	adam	104	Multivariate	16.107470
286	104	40	Model 286	nadam	104	Multivariate	17.215180
46	1	40	Model 46	nadam	104	Multivariate	17.250147
202	52	40	Model 202	adam	4	Multivariate	18.637282
124	13	40	Model 124	nadam	1	Multivariate	18.896268
130	13	40	Model 130	nadam	4	Multivariate	18.949729
52	2	40	Model 52	adam	1	Multivariate	19.147302

Table 12: RMSE for each model in the grid search sorted after RMSE.

By examining table 12 we can clearly see that the best performing LSTM model is run number 186 with a batch size of 26, 100 epochs, 52 timesteps and with the nadam optimizer. Still, the RMSE is somewhat worse than the best SARIMAX model. This may indicate that the neural network was not able to capture all the trends in the dataset.

An interesting observation is that while the SARIMA and SARIMAX were quite similar measured by RMSE, every single of the best LSTM models are multivariate. The LSTM may therefore be better suited for multivariate time series than univariate, and may need a larger amount of data to be able to capture the trends.

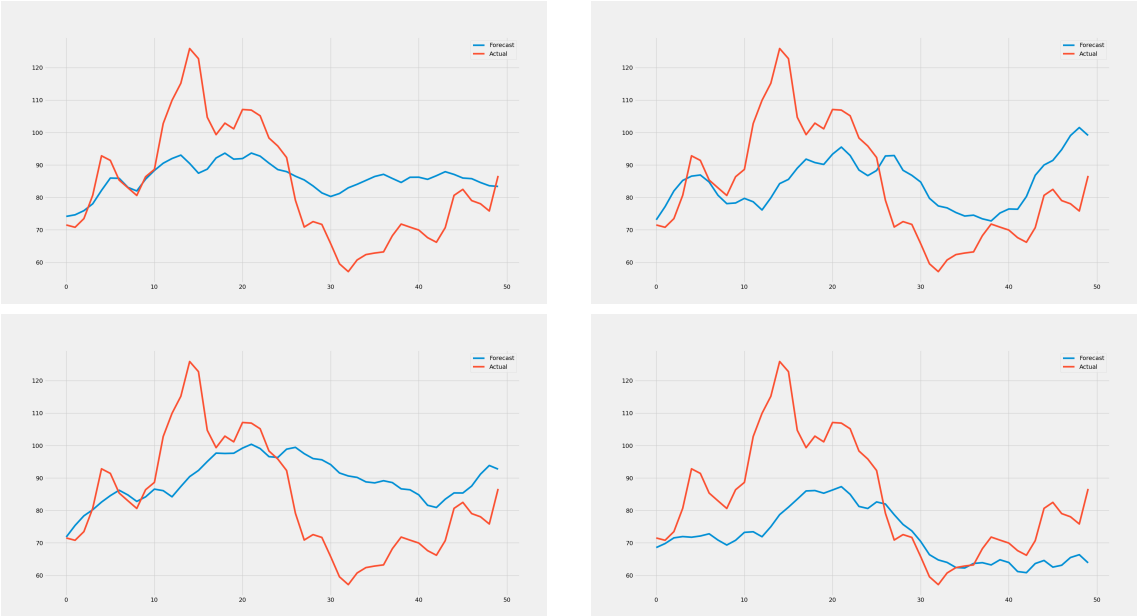


Figure 21: The eight best models measured by RMSE, 1–4, actual data in red, predicted data in blue

Figure 22: The eight best models measured by RMSE, 5–8, actual data in red, predicted data in blue

By plotting the models from table 12 we can see that the four best models are able to quite accurately predict the next year, while the next four models does not seem to capture any other trend than a rolling mean. The RMSE from these models may therefore be somewhat misleading as they are not able to capture any complex trends.

In order to determine the effect of the different hyperparameters we can group by these and calculate the mean RMSE for each group. This is done in the following tables:

Batch Size	RMSE
1	30.772256
2	30.464227
13	31.391215
26	31.309650
52	31.555332
104	30.439968

Table 13: Mean RMSE for each batch size.

Time steps	RMSE
1	30.994684
4	29.726808
52	30.672561
104	32.561046

Table 14: Mean RMSE for each timestep.

Epochs	RMSE
10	31.923834
40	31.494624
100	29.547866

Table 15: Mean RMSE for each epoch.

Optimizer	RMSE
adam	30.869023
nadam	31.108526

Table 16: Mean RMSE for each optimizer.

Uni/Multi	RMSE
Multivariate	29.319889
Univariate	32.657660

Table 17: Mean RMSE for each univariate or multivariate model.

From the different tables we can discern that there are not any large difference in RMSE between the different hyperparameters. With the exception of univariate to multivariate where the multivariate models do have a clear lower RMSE. An interesting conclusion to draw is that the batch size seem to have a small effect on the RMSE, and it may therefore be more efficient with a large batch size, as this should decrease the computational time

needed. Another is that the number of time steps does not have a large effect on the RMSE, even though we hypothesised that it may see more seasonality when introducing more time steps. This may still be attributed to the models ability to capture the rolling trend of the data in such a way that the RMSE becomes somewhat misleading.

The number of epochs in table 15 does not seem to vary from 10 to 40, but seems to have some effect when increasing to 100 epochs. This may either be due to the need for more epochs to draw out the information from the data, especially with such a simple model, or it may be because of overfitting of the data. Taking a look at the loss plot from the same models we examined in figure 21 and 22 we see that for the models with an actual variation in the prediction there is a logarithmic decrease, converging towards 0 loss. Comparatively, the models where the predictions are more similar to a rolling mean, the loss exhibits a large amount of variation. This may indicate that the model is not able to learn anything from the data, and is therefore not able to improve the prediction.

Interestingly, several of the models that perform well in terms of RMSE, have an optimal epoch of 40. This might indicate that the loss will increase with more epochs.

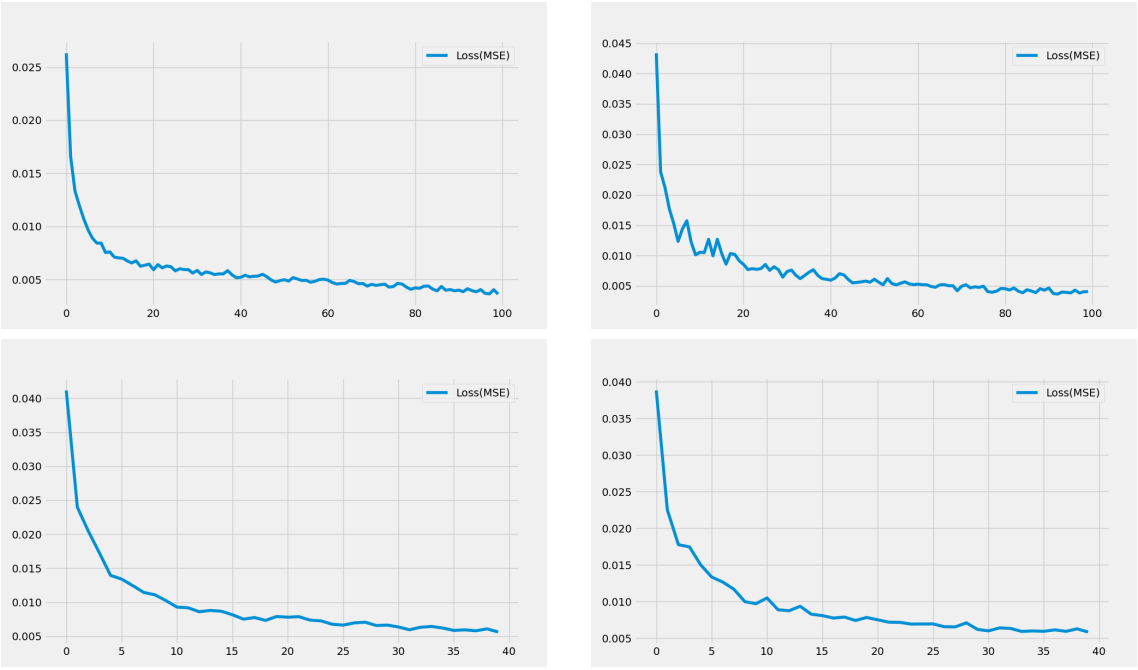


Figure 23: Loss plot for the eight best models measured by RMSE, 1–4

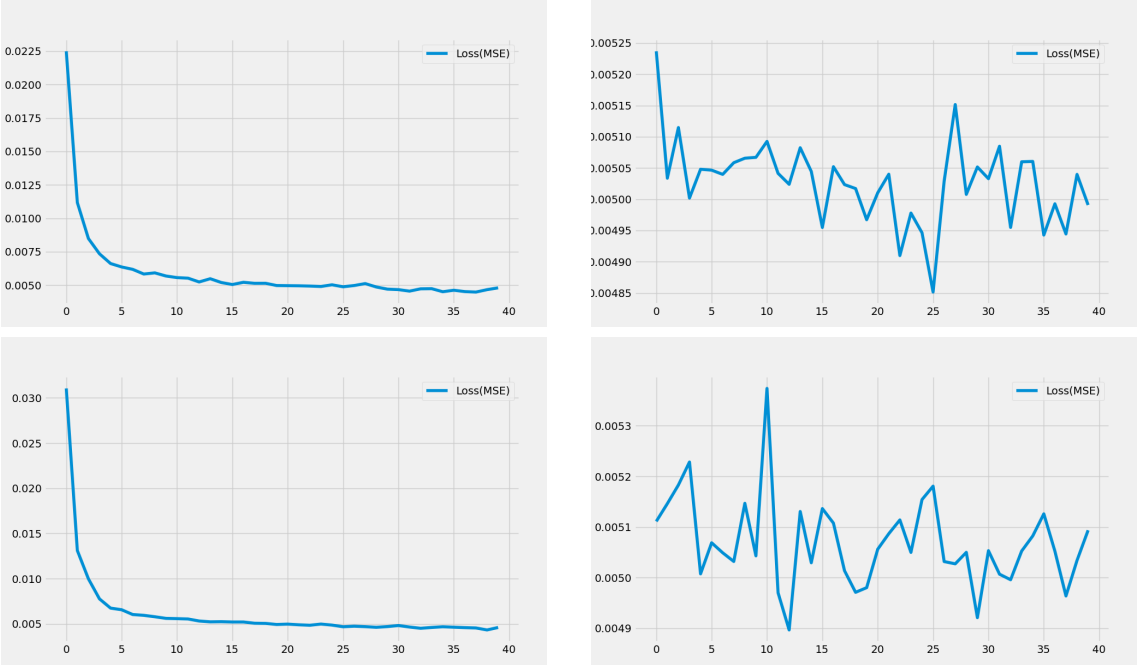


Figure 24: Loss plot for the eight best models measured by RMSE, 5–8

4.5 Comparison and Discussion

The biggest difference in RMSE is from ARIMA to SARIMA, indicating the importance of seasonality. Adding exogenous variable to the SARIMA, making it a SARIMAX model, improves the RMSE very slightly, but the SARIMAX model still seems to be suboptimal. Comparing the RMSE from the SARIMAX model and the LSTM model we see that the best prediction from the LSTM is still slightly worse than the SARIMAX model. This being said there is a little drop off in the RMSE from the best prediction from the LSTM to the second best, meaning that cherry picking the best prediction from the LSTM gives a wrong impression of how well the LSTM actually performs. At the same time its also worth noting that without the lack of the necessary computing power, as well as the other limitations listed below, the LSTM model has the potential to be the best model.

One limitation in our data is the massive swings of the salmon price in 2022. Compared to earlier years we see that the trends are still the same in 2022, but the extent of the price-spikes are much larger. This means that actually producing a model that predicts the price in 2022 relatively well would mean that the model is in fact not very good. Good models would have to produce lower highs and higher lows than what was actually the case in 2022. This is the case for our test set as well, and is very clearly illustrated in figure

20 where the actual price is peaking outside of the 95% confidence interval. In order to succeed in predicting such a steep climb in price with certainty the model would have to capture a much larger portion of the factors that go into determining the price.

Given that the NASDAQ gathers it data though sampling form a large range of sales venues each week and by doing so aggregating a good average with little bias over time. This might cause the LSTM models with a low amount of lookbacks to be biased in some of its predictions because it looks at this small sample isolated. This may be one the reasons why the model is not able to catch the trends as well when utilizing low amounts of lookbacks.

Even though our best model, the SARIMAX, seems to be somewhat able to catch the trends, we cant say that there is not more information in the residuals. The results from the Ljung-Box test means that we have to reject the null hypothesis. However, rejecting the null hypothesis only tells us that we cant claim that there is not white noise. It does not mean that we can claim that there is white noise. This uncertainty means that we cant be sure how much of the price fluctuations are explained by the model and how much is left in the residuals.

When suggesting further research we note that when examining the Ljung-Box test on the SARIMAX model. It shows that the residuals might not be white noise, and that there might be some trends left to capture. We therefore suggest that future research focuses on improving the models and perhaps lengthen the dataset in order to get better odds of picking up these more illusive trends. One possibility for improving the models is simply to use more computing power as this would allow for more complex models. Another possibility is to use different variables in the multivariate analysis in LSTM as we saw the multivariate was by far the best version of LSTM. This might also apply to the SARIMAX model. It is possible to try different models entirely, but we suggest focusing on multivariate.

5 Conclusion

In this thesis we have predicted the salmon price using different analytical models. In doing so we have tested how well these different models perform, when trying to predict the price of salmon using similar commodities and macroeconomic factors. The approach we had to solving this task were to first perform a simple exploratory analysis to see if there was any changes needed to be done in the pre-processing stage. This was not the case. We then built four different models: ARIMA, SARIMA, SARIMAX and LSTM.

The general findings are that the SARIMA and SARIMAX models produces quite similar predictions, and both outperform the ARIMA model. This clearly indicates that seasonality must be taken into account. When examining the SARIMAX predictions we see that the confidence interval is rather large. However, the salmon price spiked massively in the spring of 2022, so this is necessary. Therefore, simply examining whether or not the model predicts the correct movement of the price is of relevance. Although the model is not great, it is somewhat capable of catching the trends in the short term. The LSTM model seems not to be able to catch the seasonality before it reaches 52 Time steps. This makes sense as we saw from the exploratory analysis that the seasonality is a yearly pattern. But when looking back at 52 weeks or more then the models get better at catching the seasonality as well as other trends.

Overall, the SARIMAX model seems to be the best model for predicting the salmon price. However, the LSTM model has the potential to be the best model, but it needs more work. The struggles of the best models were mainly predicting the big magnitude in price change as it is following the price change in direction in many cases, but not as steep as the actual change.

References

Artley, B., 2022. *Time series forecasting with arima , sarima and sarimax*, May. Available from: https://towardsdatascience.com/time-series-forecasting-with-arima-sarima-and-sarimax-ee61099e78f6 [Accessed 22 March 2023].

Bank, N., 2020. *Twi, trade weighted exchange rate*. Norges Bank, September. Available from: https://www.norges-bank.no/en/topics/Statistics/twi/ [Accessed 14 April 2023].

Berge, A., 2020. *Dette er verdens 20 største lakseoppdrettere*, July. Available from: https://ilaks.no/dette-er-verdens-20-storste-lakseoppdrettere-2/ [Accessed 21 March 2023].

Biswal, A., 2023. *What is exploratory data analysis? steps and market analysis — simplilearn*, February. Available from: https://www.simplilearn.com/tutorials/data-analytics-tutorial/exploratory-data-analysis [Accessed 30 March 2023].

Bloznelis, D., 2018. Short-term salmon price forecasting. *Journal of forecasting* [Online], 37(2), pp.151–169. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/for.2482. Available from: https://doi.org/10.1002/for.2482.

Bouvet, 2020. *Explaining recurrent neural networks*. Available from: https://www.bouvet.no/bouvet-deler/explaining-recurrent-neural-networks [Accessed 30 March 2023].

Brownlee, J., 2016a. *Data leakage in machine learning*, August. Available from: https://machinelearningmastery.com/data-leakage-machine-learning/ [Accessed 29 March 2023].

Brownlee, J., 2016b. *How to grid search hyperparameters for deep learning models in python with keras*, August. Available from: https://machinelearningmastery.com/grid-search-hyperparameters-deep-learning-models-python-keras/ [Accessed 20 April 2023].

Brownlee, J., 2017. *Time series forecasting performance measures with python*, January. Available from: https://machinelearningmastery.com/time-series-forecasting-performance-measures-with-python/ [Accessed 13 April 2023].

Brownlee, J., 2019. *Tensorflow 2 tutorial: get started in deep learning with tf.keras*, December. Available from: https://machinelearningmastery.com/tensorflow-tutorial-deep-learning-with-tf-keras/ [Accessed 19 April 2023].

Chang, X., Gao, M., Wang, Y. and Hou, X., 2012. Seasonal autoregressive integrated moving average model for precipitation time series. *Journal of mathematics and statistics* [Online], 8(4), April, pp.500–505. Available from: https://doi.org/10.3844/jmssp.2012.500.505.

Dickey, D.A. and Fuller, W.A., 1979. Distribution of the estimators for autoregressive time series with a unit root. *Journal of the american statistical association* [Online], 74(366a), pp.427–431. eprint: https://doi.org/10.1080/01621459.1979.10482531. Available from: https://doi.org/10.1080/01621459.1979.10482531.

Dobilas, S., 2022. *Rnn: recurrent neural networks — how to successfully model sequential data in python*, February. Available from: https://towardsdatascience.com/rnn-recurrent-neural-networks-how-to-successfully-model-sequential-data-in-python-5a0b9e494f92 [Accessed 19 April 2023].

Dolphin, R., 2021. *Lstm networks — a detailed explanation*, March. Available from: https://towardsdatascience.com/lstm-networks-a-detailed-explanation-8fae6aefc7f9 [Accessed 30 March 2023].

Elamin, N. and Fukushige, M., 2018. Modeling and forecasting hourly electricity demand by sarimax with interactions. *Energy* [Online], 165(0360-5442), December, pp.257–268. Available from: https://doi.org/10.1016/j.energy.2018.09.157.

Gu, G. and Anderson, J.L., 1995. Deseasonalized state-space time series forecasting with application to the us salmon market. *Marine resource economics* [Online], 10(2), pp.171–185. eprint: https://doi.org/10.1086/mre.10.2.42629109. Available from: https://doi.org/10.1086/mre.10.2.42629109.

Harland, A.O., 2022. *Åpenhetsloven*. Available from: https://www.rafisklaget.no/rundskriv-list/aapenhetsloven [Accessed 13 April 2023].

Hayes, A., 2019. *Autoregressive integrated moving average (arima)*. Available from: https://www.investopedia.com/terms/a/autoregressive-integrated-moving-average-arima.asp [Accessed 23 March 2023].

Hyndman, R.J. and Athanasopoulos, G., 2021. *Forecasting: principles and practice*. Melbourne: Otexts.

Johansen, U., Bull-Berg, H., Vik, L.H., Stokka, A.M., Richardsen, R. and Winther, U., 2019. The norwegian seafood industry – importance for the national economy. *Marine policy* [Online], 110, p.103561. Available from: https://doi.org/10.1016/j.marpol.2019.103561.

Khalajani, 2023. *Gradient background - unlimited download. cleanpng.com.* Available from: https://www.cleanpng.com/png-recurrent-neural-network-artificial-neural-network-1374468 [Accessed 30 March 2023].

Meisingset, S., 2023. *Skyhøye gasspriser ga historisk høy eksport i 2022*, January. Available from: https://e24.no/norsk-oekonomi/i/xgOd7X/skyhoeye-gasspriser-ga-historisk-hoey-eksport-i-2022 [Accessed 21 March 2023].

Nau, R., 2019. *Introduction to arima: nonseasonal models*. Duke University. Available from: https://people.duke.edu/~rnau/411arim.htm [Accessed 22 March 2023].

Nielsen, B., 2022. *Hva er en sluttseddel?* Available from: https://www.rafisklaget.no/fiskipedia/sluttseddel [Accessed 13 April 2023].

Norway, S. from, 2023. *Clipfish*. Available from: https://fromnorway.com/seafood-from-norway/clipfish/ [Accessed 21 March 2023].

Oracle, 2023. *Oracle® crystal ball reference and examples guide*. Available from: https://docs.oracle.com/cd/E57185_01/CBREG/ch06s03s04s01.html [Accessed 22 March 2023].

Perloff, J., 2017. *Microeconomis: theory and applications with calculus*. Boston: Pearson Education.

Prado, M.L. de, 2018. The 10 reasons most machine learning funds fail. *SSRN electronic journal* [Online]. Available from: https://doi.org/10.2139/ssrn.3104816.

Saeed, M., 2021. *An introduction to recurrent neural networks and the math that powers them*, September. Available from: https://machinelearningmastery.com/an-introduction-to-recurrent-neural-networks-and-the-math-that-powers-them/ [Accessed 30 March 2023].

Sharma, S., 2019. *Explained: deep learning in tensorflow — chapter 0*, December. Available from: https://towardsdatascience.com/explained-deep-learning-in-tensorflow-chapter-0-acae8112a98 [Accessed 20 April 2023].

Sjømatråd, N., 2023. *Nøkkeltall*. Available from: https://nokkeltall.seafood.no/ [Accessed 21 March 2023].

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I. and Salakhutdinov, R., 2014. Dropout: a simple way to prevent neural networks from overfitting. *Journal of machine learning research* [Online], 15(56), pp.1929–1958. Available from: http://jmlr.org/papers/v15/srivastava14a.html.

TensorFlow, 2023. *Introduction to tensors — tensorflow core*. Available from: https://www.tensorflow.org/guide/tensor [Accessed 20 April 2023].

TowardsAI, 2020. *Introduction to deep learning with tensorflow – towards ai*, August. Available from: https://towardsai.net/p/deep-learning/introduction-to-deep-learning-with-tensorflow [Accessed 17 April 2023].

Vukina, T. and Anderson, J.L., 1994. Price forecasting with state-space models of non-stationary time series: case of the japanese salmon market. *Computers & mathematics with applications* [Online], 27(5), pp.45–62. Available from: https://doi.org/10.1016/0898-1221(94)90075-2.

WWF, 2019. *Sustainable seafood — industries — wwf*. Available from: https://www.worldwildlife.org/industries/sustainable-seafood [Accessed 21 March 2023].

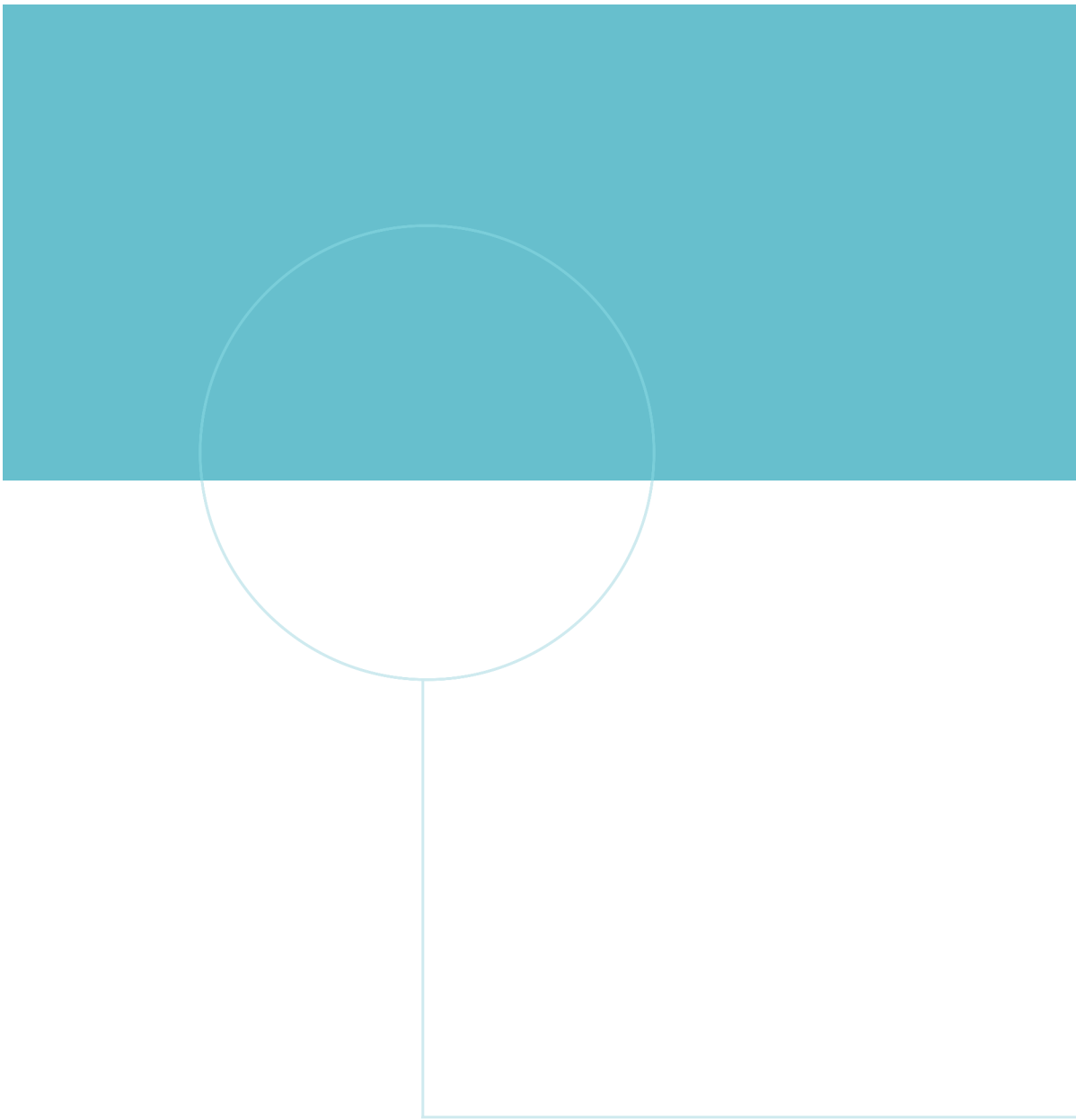
Yegulalp, S., 2018. *What is tensorflow? the machine learning library explained*, June. Available from: https://www.infoworld.com/article/3278008/what-is-tensorflow-the-machine-learning-library-explained.html [Accessed 30 March 2023].

Zaman, S.M., Hasan, M.M., Sakline, R.I., Das, D. and Alam, M.A., 2021. A comparative analysis of optimizers in recurrent neural networks for text classification. *2021 ieee asia-pacific conference on computer science and data engineering (csde)* [Online], pp.1–6. Available from: https://doi.org/10.1109/CSDE53843.2021.9718394.

Appendix

GitHub Repository

All data, code and figures used in the thesis can be found in the GitHub repository https://github.com/BachelorLaks/bachelor2023 which will be made public after the thesis is graded.



NTNU

Norwegian University of
Science and Technology