Predicting Brent Oil Futures Returns with Machine Learning and Text Data

August Nicolai Meland

Dissertation written under the supervision of Professor Dan Tran

Dissertation submitted in partial fulfilment of requirements for the MSc in Finance, at the Universidade Católica Portuguesa, 04/01/2023

Acknowledgements

I want to thank my supervisor, Professor Dan Tran, for fuelling my interest in machine learning and in the general field of artificial intelligence, and for boosting my motivation throughout this dissertation.

Abstract

In this thesis we develop a multivariate time-series classification model using machine learning techniques and the FinBERT model for extracting sentiment on daily oil news headlines gathered from oilprice.com. The model improves classical methods for predicting brent oil futures by using natural language processing techniques to capture daily market sentiment on oil and by using machine learning techniques and models to capture non-linearity in the data. Using model assessment techniques, we choose an ensemble model to conduct simple trading strategies to show how the developed models can be used in the financial markets. In addition, we explore how news sentiment about oil affects the return of brent oil futures under extreme events by conduction an event study. This thesis finds that by using the developed model in trading brent oil futures with short-sale-constraints, one can outperform a simple buy-and-hold trading strategy in a bull market. As the model shows a clear bias towards predicting positive future trading days, limitations have to be set on how the model would perform in a bear market.

Title: Predicting Brent Oil Futures Returns with Machine Learning and Text Data

Author: August Nicolai Meland

Keywords: Machine Learning, Quantitative Finance, NLP, FinBERT, Oil, News, Sentiment

Resumo

Nesta tese, desenvolvemos um modelo de classificação de séries temporais multivariadas usando técnicas de aprendizado de máquina e o modelo FinBERT para extrair o sentimento em manchetes de notícias diárias sobre petróleo coletadas do oilprice.com. O modelo melhora métodos clássicos para prever os futuros de petróleo Brent usando técnicas de processamento de linguagem natural para capturar o sentimento diário do mercado sobre o petróleo e usando técnicas e modelos de aprendizado de máquina para capturar a não linearidade dos dados. Usando técnicas de avaliação de modelos, escolhemos um modelo de conjunto para realizar estratégias de negociação simples para mostrar como os modelos desenvolvidos podem ser usados nos mercados financeiros. Além disso, exploramos como o sentimento de notícias sobre o petróleo afeta o retorno dos futuros de petróleo Brent em eventos extremos, realizando um estudo de eventos. Esta tese conclui que, ao usar o modelo desenvolvido na negociação de futuros de petróleo Brent, é possível superar uma estratégia de compra e manutenção simples em um mercado de alta.

Title: Predicting Brent Oil Futures Returns with Machine Learning and Text Data

Author: August Nicolai Meland

Keywords: Machine Learning, Quantitativo Financeiro, NLP, FinBERT, Óleo, Notícias, Sentimentio

Table of Contents

List of Tables

List of Figures

1. Introduction

There is no hiding from oil, the black gold has become the world's most important commodity and has been a significant reason for the rise and fall of numerous countries. On a personal note, there is no denying that oil has had an enormous effect on the Norwegian economy, which is one of my motivations for choosing oil as a research target.

With recent events like the coronavirus-pandemic, supply-chain blockades, the war in Ukraine, and an energy shock in Europe, we can clearly see the importance of oil. This resource is vital to every country and will continue to be in decades to come, even as we transition into renewables and green alternatives.

Natural Language Processing (NLP), a subfield within computer science and Artificial Intelligence (AI), is a tool to extract sentiment and make sense of unstructured text data. By gathering text data from sources like newspaper headlines, financial reports, or social media sites, NLP can be used to extract meaningful information and help us make data-driven decisions. NLP is currently an innovative trend in finance with a lot of future potentials.

By using machine learning techniques and the state-of-the-art FinBERT model for news sentiment analysis, we develop a multivariate time-series classification model using machine learning techniques that predicts the future market movement of brent oil futures at time t+1 until t+N. Most previous studies that model oil prices rely primarily on econometric models, historical statistical data, or official macroeconomic data, which used alone might be insensitive to real-time economic issues (Li et al., 2019), or simply does not capture the nonlinearity of the data. This thesis tries to overcome these shortcomings by (1) using daily sentiment scores on oil news headlines as data to capture real-time sentiment on oil and (2) introducing machine learning models, including random forest classifiers, artificial neural networks, logistic regression, support vector machine and XGBoost to capture non-linear relationships in the data.

Event studies investigate how oil sentiment scores before and after oil-related news affects the return of brent oil futures. The return path of extreme positive sentiment events starts, on average, to increase from day t-10 and up until day t+3, from where it seems the news is turning stale. We see a decrease for extreme negative sentiment events from day t-9 until the day of the event. Somewhat surprisingly, extreme negative sentiment events increase from the event day until day t+4, from where it again continues its decline.

We will discuss machine learning (ML), a central subject in this thesis, and natural language processing (NLP), which is used to extract sentiment data, as well as the FinBERT model which is the model we use to extract our sentiment scores. We describe some common pitfalls that can arise when using machine learning techniques, such as overfitting, underfitting, and lack of interpretability. We also introduce the process of extracting, selecting, and transforming raw data into features, called feature engineering, as well as the several different types of ML models, including artificial neural networks (ANNs) and random forest classifiers (RFCs), and how these models work and how they are used in the context of this particular thesis.m

We use F1, accuracy, precision, and recall scores to assess the performance of every classification model, as well as for comparison. We find that, in general, all classification models mentioned above are better at predicting future positive trading days than predicting future negative trading days. On average, our models correctly predict 64% of the future positive trading days and 54% of the future negative trading days when testing on a test set using a rolling window. Adding on this, we find that the years with the highest average F1 scores are 2019, 2021, and 2022, and the years with the lowest average F1 scores are 2016, 2017, and 2020. As 2019, 2021, and 2022 were considered good years for oil, and 2016, 2017, and 2020 were considered bad years, these results also suggest that the model is a good predictor in bull markets and a bad predictor in bear markets.

Further, to put our findings in more financial terms, we create two trading strategies based on an ensemble model that leverages the predictability of future positive trading days by implementing a no-short-sale restriction. All long-only portfolios without leverage significantly outperform a buy-and-hold portfolio; this also holds when considering trading costs of 0.75%. All levered long-only portfolios do not beat the benchmark, except 2x leverage in the first trading strategy.

We successfully develop a multivariate time-series classification model that utilizes sentiment scores obtained from the state-of-the-art FinBERT model. Our model is based on a direct forecasting approach which has the advantage in that it is directly optimized to forecast each step. Some cons of this approach are that we have to build multiple models and assess their individual performance, as well as the computational cost of this approach. The model successfully works as a good predictor in bull markets but has its limits in bear markets. Further research and development could help increase the model's predictability on future negative trading days which will making the model even reliable. Further limitations are discussed in

chapter 8. This study contributes to existing literature by introducing FinBERT as a NLP model to extract sentiment scores from oil news headlines, which is used to develop a multivariate time-series classification model that utilizes ML techniques.

2. Literature Review

Oil has a dominant occupation in the commodity market, and it has become the world's primary energy source and essential commodity (Yang & Zhou, 2020). Researchers have engineered numerous models and intelligent algorithms to forecast crude oil prices. Among them are the integration of intelligent algorithms and econometric models (Wang et al., 2012), machine learning methods (Zhao et al., 2017), probabilistic models (Abramson & Finizza, 1995), and crude oil spot price forecasting model using relative inventories (Ye et al., 2005).

The research methods mentioned by the researchers utilize historical data and official macroeconomic statistics. Moreover, this data alone might be insensitive to real-time economic issues (Li et al., 2019). As we see technologies advance and people increasingly spend more of their life online, the amount of unstructured real-time data is growing daily. As more real-time data becomes readily available, we might see a shift in how we measure economic activity and changes in our predictive models. To further improve models for forecasting crude oil prices, using unstructured data sources like online text has begun to see potential in recent literature. The potential information embedded in unstructured big data, such as textual data, provides a novel data source for oil price forecasting (Li et al., 2019). Online data also cover a more comprehensive range of contents than official statistical data. It may even contain some key predictive features that are difficult to capture for official statistical data (Gong et al., 2022). Highlighting recent events, the invasion of Ukraine (2022) is a geopolitical event that is not captured quickly by official statistical data. Further, the evidence from using an empirical decomposition (EMD) approach suggests that the primary driving force behind crude oil price fluctuations is significant midterm events (Zhang et al., 2019).

The efficient market hypothesis (EMH) (Samuelson, 1965; Fama, 1963, 1965) suggests that market prices are reflected by all information available to market participants. However, humans suffer from psychological and emotional biases that can lead to irrationality when acting as market participants in financial markets. Thus, we have the battle between EMH and behavioral finance. However, there has been enough research on investor rationality to say that

the question is no longer whether investor sentiment affects stock prices but how to measure and quantify it (Baker & Wurgler, 2007).

The most widely used and popular method for analyzing and preprocessing unstructured text data is sentiment analysis (Clements & Todorova, 2016; Li et al., 2020). Scholars have been trying to identify and capture investor concern and sentiment in financial markets and their influence on the oil market due to the influence on speculative oil demand and, therefore, oil prices (Ye et al., 2020).

For example, Tetlock (2007) finds that media pessimism has predictive power for stock market prices when he analyzes the interactions between a popular column in the Wall Street Journal (WSJ) with daily content and the stock market. Furthermore, Li et al. (2017) use Henry's Finance-Specific Dictionary to extract sentiment and find evidence to support the robust predictability of obtained sentiment on oil price trends. A common weakness of extracting sentiment using dictionaries is that the sentiment features generated by lexicon-based methods are insensitive to domain-specific terms and negative phrases, resulting in a nonadaptive and incomplete capture of information (Fan et al., 2021). Recently though, Araci (2019) has introduced FinBERT, a pretrained natural language processing (NLP) model that specializes in extracting sentiment from financial text data, which we will utilize in this paper. Another text preprocessing technique called Topic Modeling is popular in recent literature but designed explicitly for preprocessing long texts, like news articles, instead of news headlines. As Shi et al. (2018) described, news headlines can be arbitrary, noisy, and ambiguous, as they only contain a few words. Recent literature from Li et al. (2019) and Gong et al. (2022) also employs convolutional neural networks (CNNs) to analyze text features extracted from online news and to generate a key sentiment indicator. Also in recent literature, Kelly et al. (2019) develops a text-mining method which extract information from news articles to predict asset returns.

So far, news-based techniques and analyses are widely applied in other fields, such as the stock market, exchange market, and cryptocurrency market, but not in energy finance (Gong et al., 2022). With limited research, Bai et al. (2022), Li et al. (2019), and Liu et al. (2018) are among a few who use financial news to conduct analyses on the crude oil market. Wex et al. (2013) and Yu et al. (2005) are also a few who earlier have researched how financial news can improve crude oil forecasting. However, none to our knowledge have researched how news sentiment can be used to improve a brent oil forecasting model, together with the FinBERT model for sentiment analysis and an ensemble ML model for multivariate time-series classification.

3. Data Description

In this research, we utilize four primary datasets, namely a news headline dataset, a financial market dataset, an oil market dataset, and a macroeconomic dataset. Additionally, we calculate and add technical features based on the primary datasets. All the data are collected on daily observations.

Data Collection

To obtain our text data, we created a crawling pipeline in Python and collected 30 375 news headlines from the archive section on oilprice.com, published in the period from February 5th, 2010, to October 14th, 2022. Crawling is a widely used strategy by programmers to download webpages systematically and efficiently (Fan et al., 2021). We use news headlines instead of the full article, following that news headlines contain much less repetition and fewer irrelevant words than the news article itself (Nassirtoussi et al., 2015), as well as the considerable data processing requirement to run sentiment analysis on a big amount of text data with the FinBERT model.

We collect Brent Oil Futures Prices, the Dow Jones Industrial Average (DJIA), the Gold Spot Dollar (XAU/USD), the US Dollar Index Futures, the Goldman Sachs Commodity Index (SPGSCI), 10Y and 3Y treasury yields, CBOE Volatility Index (.VIX) and DAX Futures, all from Investing.com. We collect the Baltic Dry Index (.BADI) from the Refinitiv terminal and the 10-Year Breakeven Inflation Rate (T10YIE) from FRED. All beforementioned data is collected from the same period as the news headlines.

Following Li et al. (2019), we selected a handful of these features because previous literature has shown that futures markets, the foreign exchange market, and stock markets have spillover effects on the crude oil market, Chen and Chen (2007), Cifarelli and Paladino (2010), Lizardo and Mollick (2010), Sadorsky (1999). As we are researching brent oil and not crude oil, we make a reasonable assumption that similar effects are to be found in the brent oil market. In addition, we add the 10Y breakeven inflation rate, which implies what market participants expect the inflation rate to be in the next ten years, on average. An overview of the raw input variables can be found in table 1.

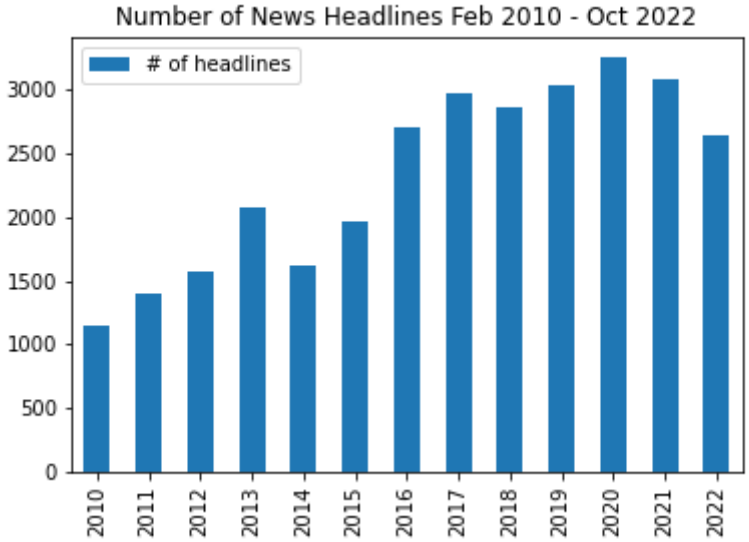
Table 1: Raw Input Variables

Input variables class	Input variables	Data source
Financial Market Data	Goldman Sachs Commodity Index Dow Jones Industrial Average US Dollar Index Gold Spot USD DAX Futures CBOE Volatility Index	Investing.com
Oil Market Data	Baltic Dry Index Brent Oil Futures Price	Refinitv Investing.com
Macroeconomic Market Data	Breakeven Inflation Rate (10Y)	fred.stlouisfed.org
News Headlines	Sentiment Scores	Oilprice.com

News Headlines Preprocessing and Sentiment Extraction

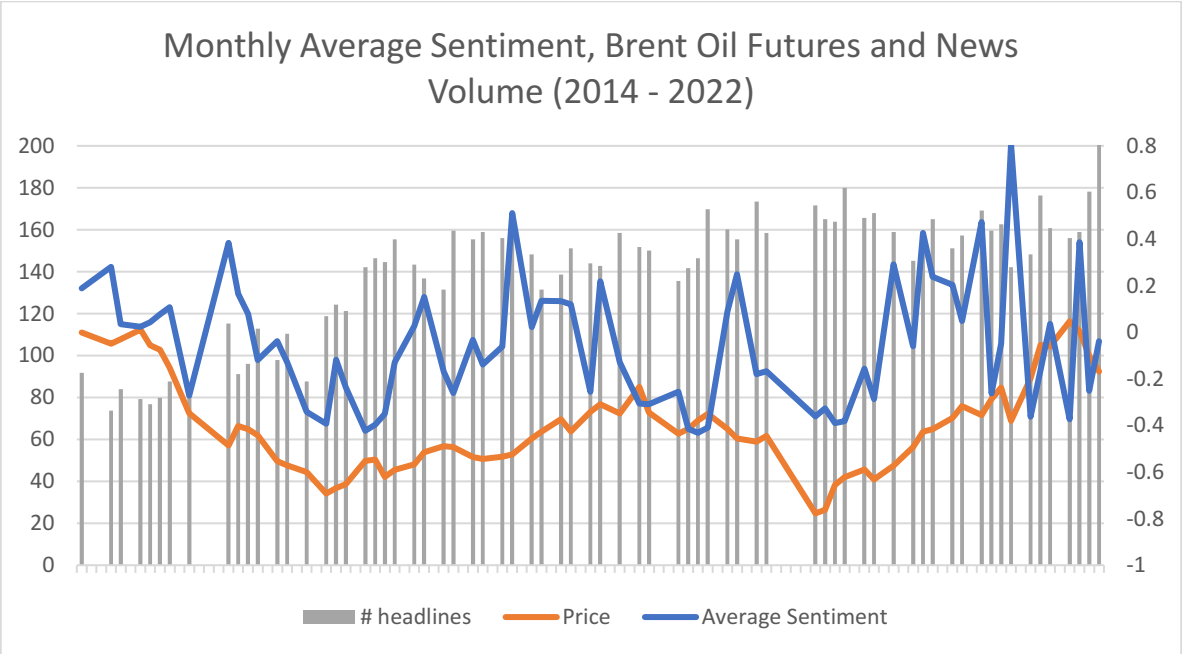
After crawling our text data, we preprocess the news headlines with the goal of removing as much noise in the data as possible. To do this, we eliminate all duplicate news headlines and create a sentence vector of all news headlines published on the same day. The sentence vectors are then processed by removing punctuations like .,!$()*%@[]. Next, we lowercase all of our text. After preprocessing our sentence vectors, we run them through the FinBERT tokenizer and FinBERT model, which returns a positive, negative, and neutral sentiment score for each sentence vector. These sentiment scores represent the probability of a given news headline being positive, negative, or neutral. The sentiment scores are then ready to be analyzed or to be added as inputs to our model.

Figure 1: Amount of Published News on oilprice.com



By looking at figure 1 above, we can see that the amount of news headlines published is following a positive trend, meaning that more and more text data (news headlines about oil) are becoming available for research.

Figure 2: Monthly Average Sentiment, Brent Oil Futures Price and News Volume



In figure 2 above, we find a positive correlation between monthly average sentiment on news headlines and the price of Brent Oil Futures from 2014 – 2022. Something that might seem

surprising at first is that we find a negative correlation between news volume and the price of Brent Oil Futures. However, an explanation for this could be that bad news is a better attention grabber than good news, which means that when the price of oil drops, more news is posted simply because news agencies know that bad news is getting more views than good news.

Technical Features

In addition to the features collected above, we calculate and add technical features based on the historical data of these. We calculate and add the 7-day cumulative sentiment score for all sentiment labels (Positive, Negative, and Neutral). We also calculate and add 2, 3, and 5-day rolling volatility of brent oil futures, the 10-Day vs. 22-Day volatility, as well as the yield spread between 3Y Treasury Yields and 10Y Treasury Yields.

Finally, our dependent feature, the target, is derived from the returns of brent oil futures. We set a binary target where days with a positive return on brent oil futures are equal to 1, and days with a negative return on brent oil futures are equal to -1.

4. Event Study

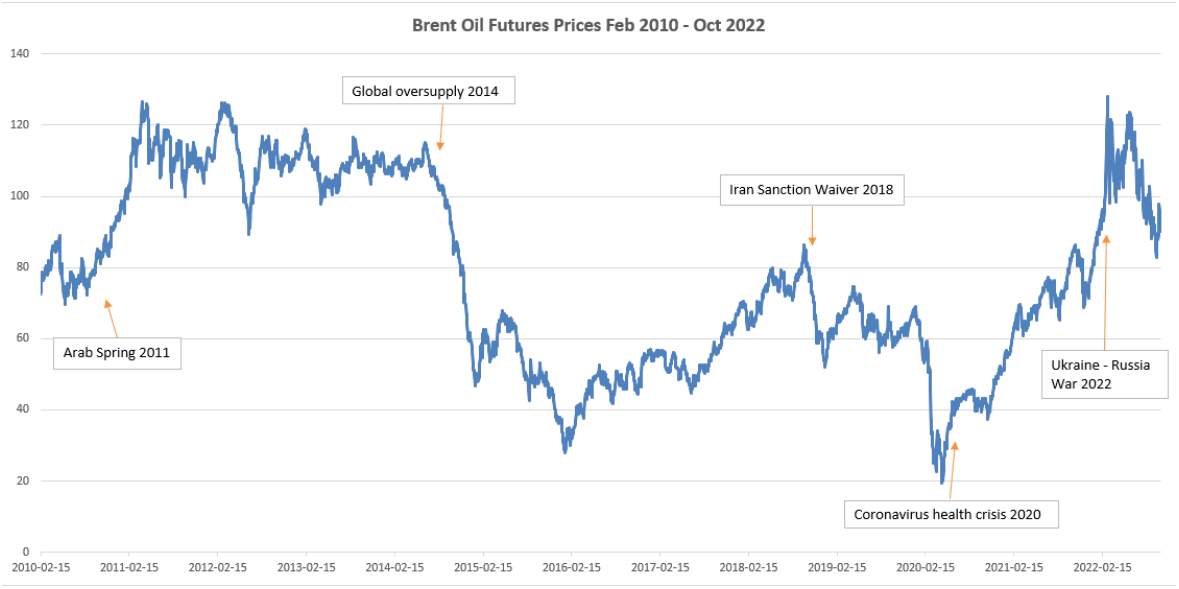
In this section, we study how news headlines with extreme sentiment scores impact the return of brent oil futures returns by using an event study methodology.

An event study is a method where we study how certain events impact a feature that we are interested in, in our case, the returns of brent oil futures. Our research will look at cumulative returns over a set period to understand how news sentiment affects brent oil futures returns.

The price of oil is affected by a variety of factors, including global oil demand, the level of oil production and supply, geopolitical tensions, and the strength of the global economy. As a result, the price of oil can be volatile and fluctuate significantly over time.

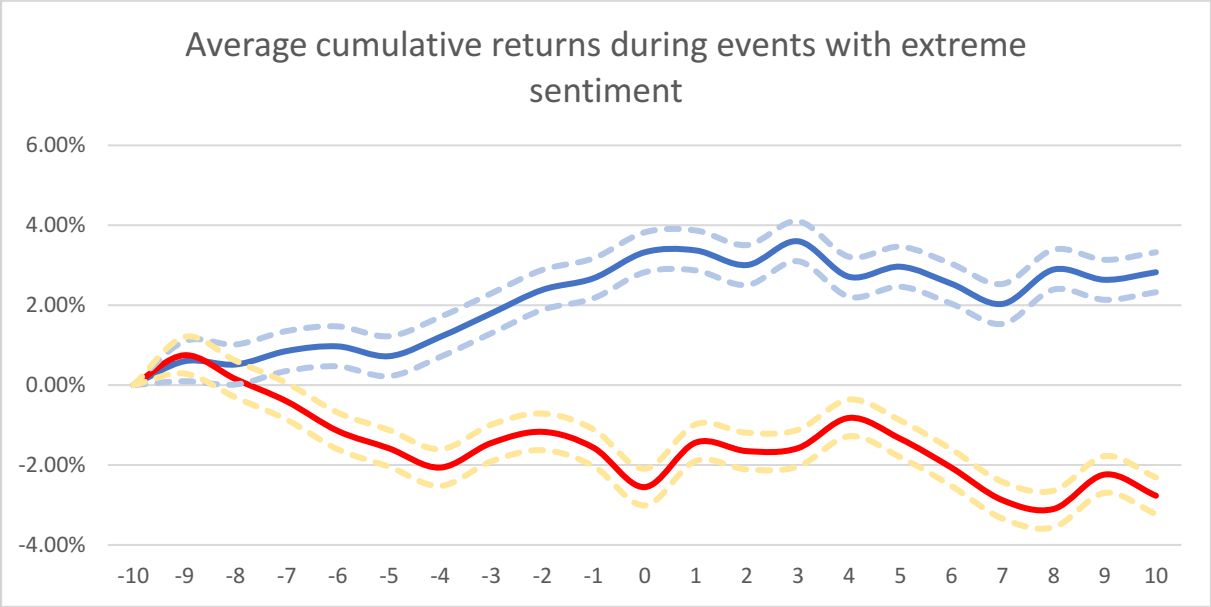
In general, the price of oil reached a peak in the mid-2000s before experiencing a significant drop during the global financial crisis in 2008. It then recovered somewhat before dropping again in 2014 due to increased production and slowing global demand. Since then, the price of oil has generally been range-bound, although it has been affected by various geopolitical events and economic conditions, like in the recent invasion of Ukraine, where we saw a quick spike in the price of oil.

Figure 3: Brent Oil Futures Prices with events



For the event study, we start off by gathering our sentiment scores from the FinBERT model and find the most positive and the most negative event for each year from February 2010 until October 2022. Following the work of Li et al. (2019) and others, we set the day of the published news headline at t=0 and gathered the t-10 days before the event and t+10 days after the event. We then calculate the cumulative return of brent oil futures each day from t-10 until t+10. We repeat this for all extreme events each year and, finally, calculate the average cumulative returns during the event window. We find that the return path of extreme positive sentiment events starts, on average, to increase from day t-10 until day t+3, where it seems the news is turning stale. We see a decrease in extreme negative sentiment events from day t-9 until the day of the event. Surprisingly, extreme negative sentiment events increase from the event day until day t+4, from where it again continues its decline. See figure 4 below.

Figure 4: Average Cumulative Returns of Brent Oil Futures during Extreme Sentiment Events



In figure 4, the solid blue line represents the average cumulative return of events with extremely positive sentiment. The red solid represents the average cumulative return of events with extremely negative sentiment. The dotted lines represent the upper and lower confidence intervals at a level of 95% significance.

5. Machine Learning

5.1 Machine Learning

Machine learning (ML) is a subfield of artificial intelligence (AI) that involves designing and developing algorithms and models that can learn from and make predictions or decisions based on data. These algorithms and models can improve their performance over time as they are exposed to new data and can learn from it. Machine learning is often used to analyze large datasets to identify patterns or trends that may not immediately appear to us humans. It has a wide range of applications, including image and speech recognition, and for this thesis: natural language processing and predictive modeling. Generally, an ML workflow consists of several steps: defining the problem, preparing the data at hand, choosing and training models, model assessment and finally, fine-tuning the models.

Some common pitfalls that can arise when we use machine learning techniques is:

1. Overfitting: This occurs when a model is trained too well on the training data, and as a result, it performs poorly on new, unseen data. This can be caused by using a too complex model for the given dataset or not having enough data to train the model.
2. Underfitting: This occurs when a model cannot capture the underlying patterns in the data, and as a result, it performs poorly on both the training and test data. This can be caused by using a model that is too simple for the given dataset or by having too much noise in the data.
3. Lack of interpretability: Some machine learning models, particularly complex ones like deep neural networks, can be difficult to interpret and understand. This can make it challenging to understand how the model makes predictions and identify any potential issues with the model.

In ML, we call our variables for features, and ML techniques require us to prepare these in specific ways via feature engineering, which is the process of extracting, selecting, and transforming raw data into features that can be used to train machine learning models. This is a crucial step in the machine learning workflow, as the quality and relevance of the features can significantly impact the model's performance. To feature engineer, we must apply domain knowledge from finance and known statistical techniques. Some standard techniques for feature engineering include normalization or standardization, one-hot encoding, and feature selection. Our ultimate goal is to create a set of relevant and informative features that we can use for our ML models without falling into underfitting or overfitting.

This thesis uses several ML models, techniques from natural language processing and thereunder, FinBERT, which are explained and outlined below.

5.2 Natural Language Processing

In general, Natural Language Processing (NLP) is a field within machine learning that focuses on enabling computers to understand, interpret, and generate human language. It involves using linguistics, computer science, and machine learning techniques to process and analyze natural language data. NLP is used in a wide range of applications, including language translation, information retrieval, chatbots, and sentiment analysis. Some everyday tasks in NLP include language translation, named entity recognition, and sentiment analysis. NLP has numerous practical applications in fields such as healthcare, customer service and finance.

FinBERT

FinBERT is derived from the BERT (Bidirectional Encoder Representations from Transformers) language model that has been specifically trained on financial texts. BERT is a type of neural network-based language model that has been developed to understand the context and meaning of words in a given sentence or paragraph. It does this by considering the words that come before and after a given the word, as well as the overall structure of the sentence. FinBERT is a version of BERT that has been trained on a large dataset of financial texts, such as news articles, annual reports, and regulatory filings. This allows it to understand financial terminology and context better and be more effective at tasks such as named entity recognition and sentiment analysis in the financial domain. FinBERT is often used in financial applications such as automated financial analysis and risk management.

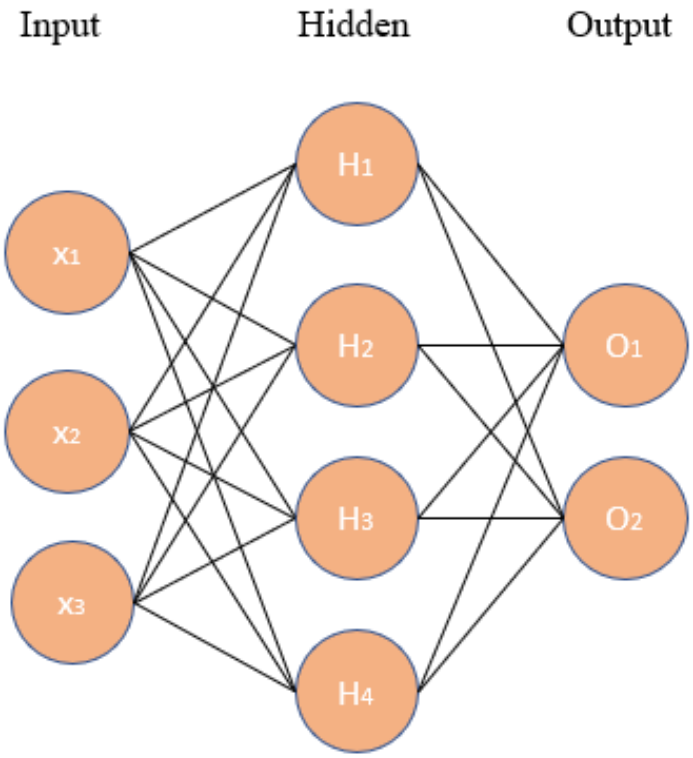
5.3 Machine Learning Models

Artificial Neural Networks

An artificial neural network (ANN) is a machine learning model inspired by the structure and function of the biological neural networks that make up the brain. ANNs are composed of layers of interconnected neurons which process and transmit information. They are trained to recognize patterns and make decisions or predictions based on input data.

There are several different types of ANNs, including feedforward networks, convolutional neural networks, and recurrent neural networks. In this thesis, we are utilizing feedforward networks composed of an input, hidden, and output layer. In short, the input layer receives the input data, and the hidden and output layers process and transmit the information through the network.

Figure 5: Visualization of a Simple Artificial Neural Network



In this thesis, we will primarily utilize two activation functions for our ANNs, the ReLU (or rectified linear unit) activation function and the Tanh activation function. Simply put, the activation function is a key component that determines the output of a neuron given its input. It is a mathematical function that takes in the input to a neuron, applies a transformation to it, and produces an output.

Random Forest Classifier

A random forest classifier (RFC) is an ML algorithm designed for classification tasks. It is an ensemble method, which means that it combines the predictions of multiple decision tree models to make a final prediction. It works by building many decision trees and training them on different subsets of the training data. We get the final prediction by averaging the predictions of all of the individual trees. The random forest classifier takes several hyperparameters that can be adjusted to the problem and data at hand, including the number of trees in the forest, the maximum depth of each tree, the minimum number of samples that are required to split a node, and the minimum number of samples required at a leaf node. A node is simply a point in the

RFC where a decision is made. Overall, an RFC algorithm is better than the average model at reducing overfitting as it combines the predictions of multiple decision trees.

Support Vector Classifier

Support vector classifier (SVC) is an ML algorithm designed for classification tasks. An SVC is a type of support vector machine (SVM), which sorts data into different groups and puts a line (or a hyperplane) between the data groups. The distance between the hyperplane and the nearest data point is what we call the margin. The ultimate goal of the SVC is to maximize our margin between the groups. This gives the SVC an advantage regarding noise or outliers in the data. An SVC is also computationally efficient to train and can rely on a very small subset of training data to make accurate predictions.

XGBoost

eXtreme Gradient Boosting (XGBoost) is an ML algorithm designed for classification tasks. Like the RFC algorithm, it uses an ensemble method that has decision trees as the base model. XGboost is generally known to provide high accuracy, computationally efficiency, and thus, its ability to handle large sets of training data. XGBoost is also good at dealing with missing values. Some hyperparameters of XGBoost that we can tune for performance are, the learning rate, number of trees, maximum depth and regularization.

Logistic Regression

In machine learning we use logistic regression (LR) for classifications tasks, LR uses the logistic function to model the probability of an event occurring. The logistic function is a sigmoid function that maps any real-valued number to a value between 0 and 1. In the context of logistic regression, this probability is used to predict the likelihood of a certain class (e.g., "negative" or "positive") based on our input features. Our goal is to find the best fitting line that separates the data points into their respective classes. This line is found by optimizing a loss function, such as the cross-entropy loss, which is commonly used. The model is then evaluated based on its ability to correctly predict the class of unseen data points in a testing set.

Ensemble Models

In machine learning an ensemble model that is a hybrid model built by combining the predictions of smaller models. The idea behind ensemble modeling is that by aggregating the predictions of multiple models, the resulting model will have improved predictive performance compared to any of the individual models. There are several different ways to build an ensemble model, including bagging, boosting, and voting. Bagging involves training multiple models on different subsets of the training data, and then averaging or voting on the predictions of the individual models. Boosting involves training a series of models in sequence, where each model tries to correct the mistakes made by the previous model. Voting involves training multiple models and then having them vote on the final prediction. Ensemble models can be particularly effective when the individual models make different types of errors, as the ensemble model can correct for these errors and produce more accurate predictions.

5.4 Dealing with Time-Series Data in ML

Training and Test Data

In machine learning we use training and test sets to evaluate the performance of our models. The goal of training a model is to create a function that can accurately predict the output for new inputs. However, simply training a model on a dataset and evaluating its performance on the same dataset can be misleading, as the model may simply memorize the training data and not generalize well to new, unseen data. This is known as overfitting (as mentioned above). To accurately assess the model's performance and ability to generalize to new data, we use a separate test set. The model is trained on the training set and then evaluated on the test set (Out-of-sample). This allows us to see how well the model generalizes to new data and how well it will perform on unseen data in the real world.

Rolling Window for Time-Series Data

Rolling window is a technique we use to analyze time-series data. It involves dividing the time series data into a series of smaller, overlapping chunks or windows, and analyzing each chunk separately. For example, if we have a time series data set with 1000 data points, we might use a rolling window of size 100 to analyze the data. This would result in 10 separate chunks of data, each containing 100 data points. We can then apply machine learning algorithms to each

of these chunks independently to identify patterns or trends in the data. The advantage of using a rolling window is that it allows us to analyze the data in a more granular way, as we are looking at smaller chunks of data rather than the entire time series. This can be very useful when we need to test our models over a longer test period, as the window chunks allow us a bigger test data set.

Data Leakage

As we are dealing with time-series data. data leakage is something we have to be aware of. In machine learning data leakage refers to the presence of information in the training data that should not be used to make predictions. This can occur when the data used to train a model includes information that would not be available at the time a prediction needs to be made. For example, if a model is trained to predict the likelihood that a trading day will be positive or negative, and the training data includes the outcome of this (whether the trading day was positive or negative), this would be considered data leakage. This is because the outcome would not be known at the time the model is being used to make a prediction. Data leakage can lead to overly optimistic model performance and inaccurate predictions when the model is used in the real world. To prevent data leakage, it is important to carefully evaluate the data used to train a model and ensure that it only includes information that would be available at the time a prediction needs to be made.

Data Purging

In machine learning, data purging is the process of removing data from a dataset that is not relevant or useful for the task at hand. This can be done for a variety of reasons, such as reducing the size of the dataset to make it easier to work with or removing data that may introduce bias or cause overfitting. Data purging is typically done as a preprocessing step before training a model. It is important to carefully consider which data to include or exclude when purging a dataset, as removing too much data can lead to a loss of important information, while keeping too much data can result in noise or redundancy that can negatively impact model performance.

5.5 Model Assessment

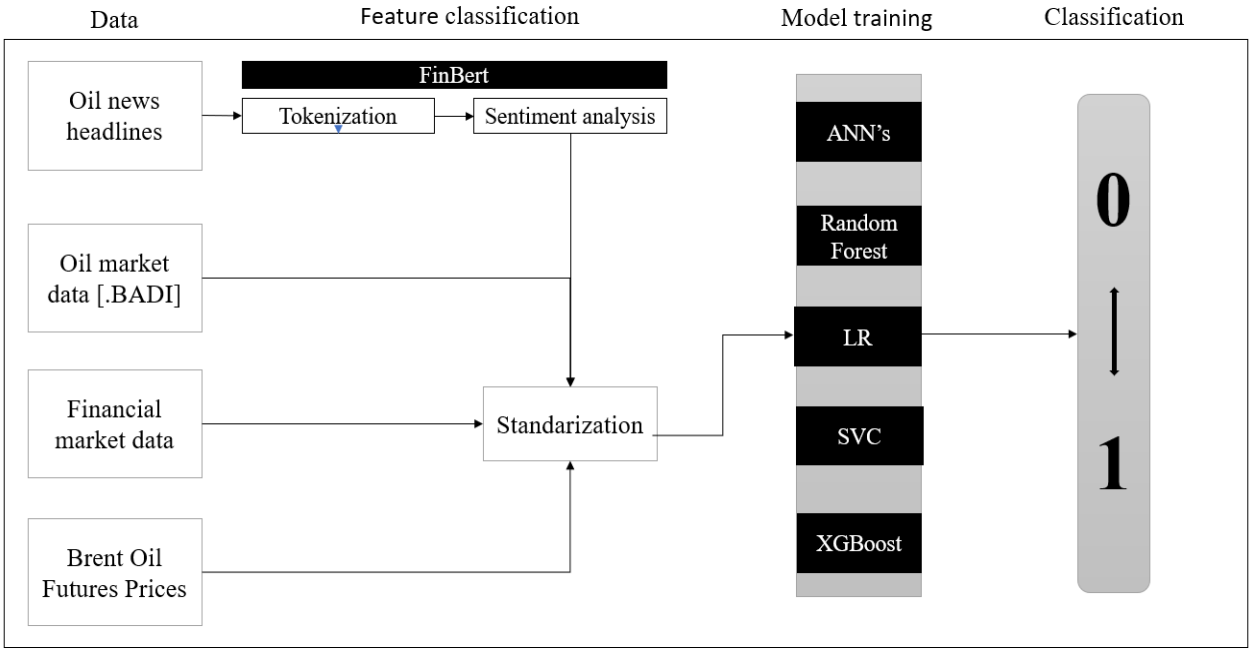
There are many ways to assess models in machine learning, depending on the problem you are trying to solve. One common approach for classification tasks is to use a confusion matrix, which is a table that compares the predicted class labels with the true class labels. The confusion matrix allows us to calculate various metrics, such as accuracy, precision, and recall, which can help understanding how well the model is performing. In short, accuracy is calculated by dividing the number of correct predictions made by our model by the total number of predictions made. Recall is defined as the number of true positive predictions made by the model, divided by the total number of actual positive cases in the dataset, we usually also use recall with precision, which measures the accuracy of positive predictions made by our model.

In our methodology chapter, we will implement the various models discussed above, as well defining training and test sets, and preprocessing with data leakage measures and data purging.

6. Methodology

The methodology in this study consists mainly of two parts. (1) sentiment extraction from oil news headlines via the FinBERT model and (2) applying machine learning models to predict the future movement of brent oil futures. In the end, we will conduct a couple of simple trading strategies to show how our models could be implemented in the financial market. For an overview of our primary methodology, see the methodology pipeline in figure 6 below.

Figure 6: Methodology pipeline



6.1 Data Preparation

Sentiment Extraction

From the news headlines collected through our crawling pipeline, we aim to extract sentiment scores which we will use for analysis and for our forecasting model. To achieve this, the pre-trained FinBERT model and the FinBERT tokenizer (Araci, D. 2019) are utilized. See FinBERT in chapter 5.

Before extracting the sentiment scores, we need to preprocess the news headlines by removing noise in the text. This is done using the NLTK library in Python. We remove all duplicates, lowercase all words and finally remove all news that is published on Saturdays and Sundays. Take notice that we do not remove stop-words (which is a common step when preprocessing text data), the reason for this is simply that FinBERT, unlike other sentiment models, looks at the syntax of the text and how the text is built structured. Further, instead of using full news articles in the analysis, we use news headlines due to several advantages: first, news headlines can provide a sufficient summary of the key news information; second, news headlines contain much less repetition and fewer irrelevant words than the news article itself (Nassirtoussi et al., 2015).

We run the processed news headlines through the FinBERT tokenizer and model to obtain three probability-based sentiment scores. The three probability scores (between 0-1) obtained tell us how probable it is that a given news headline fits the label of positive, negative, and neutral.

Table 2: Example of Sentiment Extraction on a News Headline about Oil

News headline	Positive	Negative	Neutral
Oil Prices Set for A Weekly Loss as Inflation Fears Return	0.04	0.90	0.06

Financial Market Data and Oil Data

Before being used in our machine learning model, the financial market data and oil data had to be preprocessed. After collecting data from our sources, we convert daily prices into daily logarithmic returns, following equation (1):

(1) $$r_t = \ln\left(\frac{p_t}{p_{(t-1)}}\right)$$

where

r_t = return of feature at time (t)

p_t = price of feature at time (t)

$p_{(t-1)}$ = price of feature at time (t - 1)

Final Data Preparation

Thereafter, we explore the stationarity of our features. When we say that our data is stationary, we mean that the statistical properties of our data are stationary over time. Using the Dickey-Fuller test, we find that all of our features are stationary and can be rejected at a 95% significance.

Finally, we engineer polynomial features to increase the complexity of the model and standardize the data using the sci-kit learns StandardScaler to make the features closer to standard normally distributed data. Standardization follows equation (2):

(2) $$z_i = \frac{(x_i - u)}{s}$$

where

z_i = standardized feature at observation (i)

x_i = feature at observation (i)

u = mean of traning sample

s = standard deviation of training sample

To finalize our data, we split the data into two sets, predictor features (x) and our target feature (y). Finally, we shift our predictor features one day into the past, such that we have predictors (X's) at t-1 and the target (y) at t=0. This is important, such that we do not use data from the future to predict the future, which would obviously not be possible in a real-world scenario.

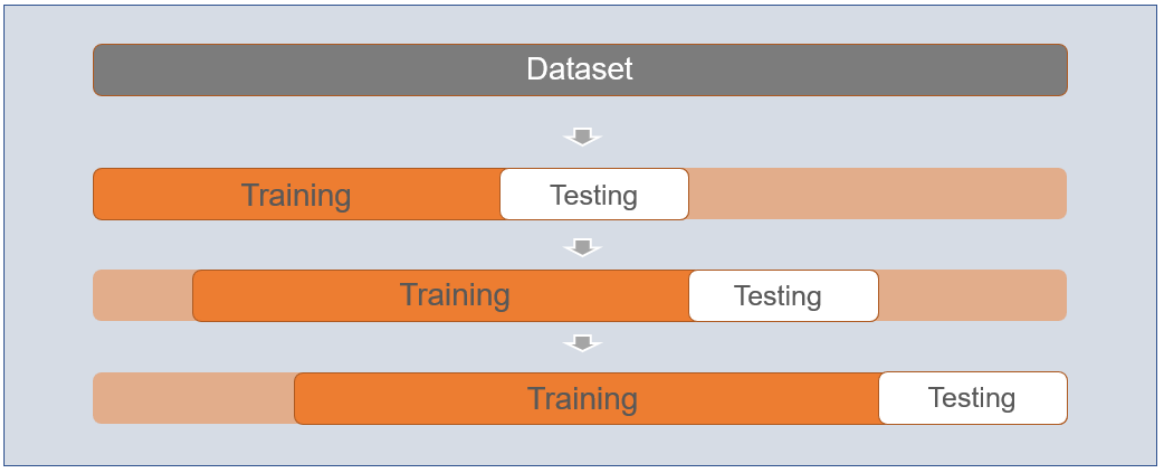
Training and Testing

After our data is prepared, we define our training and test sets. We implement a rolling window approach for training to evaluate a classifier model on our dataset. The rolling window moves across the time series in fixed steps, and at each step, a classifier is trained on the data in the current window and used to make a prediction for the next time point. We use a grid search method to find the optimal windows for our model and define a training set of 85% and a test of 15%.

To capture a reasonable amount of various market environments, the models are trained on windows of 1400 observations which roughly equals to five trading years, this also means that the first five years will not be available for testing out-of-sample. The window moves one step at a time, which is optimal when we want to predict t+1. We also delete 20 days between each training and test set to prevent data leakage and to delete data that might cause overfitting, this

process is what is known as data purging. The training is then repeated daily with a one-step rolling window and predicts a negative or positive trading day on the next day (t+1).

Figure 7: Example of a Rolling Window Workflow



6.2 Modeling

To predict negative or positive trading days at t+1 until t+n, we run the data through several different machine learning classifier models that are well known to be good at finding patterns in non-linear data, including:

- Artificial Neural Network (ANN) with tanh activation function.
- Random Forest Classifier (RFC)
- Support Vector Classifier (SVC) with rbf kernel and poly kernel.
- Logistic Regression (LR)
- XGBoost Classifier

To find the optimal hyperparameters we employ a grid search method which tests multiple hyperparameters for the given data set and outputs the best hyperparameters.

For our ANN we find the optimal four hidden layers of size 31, 120, 15 and 2, and with tanh as activation function.

As some models perform better than others at classifying positive or negative trading days, we implement ensemble learning, which combines models to create one hybrid model. We test different hybrid models, but the best performing one is a hybrid model consisting of an SVC

model with Radial Basis Function (RBF) kernel and a LR model which results in F1-score of 60%.

6.3 Robustness Check

To see the impact of our news sentiment features on the outcome of our models' predictions we repeat the same workflow as outlined above, but we remove all news sentiment features from the dataset. We find that the average F1-score as well as accuracy score is 4% higher on average with the sentiment features added to the model.

6.4 Trading Strategies

Based on the hybrid model (ensemble model) that consists of LR and SVC, we create two long-only trading strategies that shifts capital between Brent Oil Futures and cash. The strategies are based on the predictions made on a test set (out-of-sample) from 14.12.2020 until 14.10.2022, which equals 458 trading days. For our first trading strategy, if our model predicts that t+1 is positive 100% of our capital is invested in Brent Oil Futures, if our model predicts that t+1 is negative 100% of our capital is invested in cash. This process is repeated each day until the last trading day in our test sample.

For our second trading strategy, if the model predicts that t+1 and t+2 will be positive trading days, we invest 100% of our capital in Brent Oil Futures, if the model predicts that only t+1 will be positive we invest 60% in Brent Oil Futures and 40% in cash, and at last, if the model predicts that t+1 is negative, we put 100% of our capital in cash.

These two trading strategies leverages the fact that our model is good at predicting positive future trading days. Further we replicate the same strategies, but with leverage at the 2x and 5x level.

Other trading strategies were tested, such as time-weighted-averaging by increasing the amount invested when positive predictions were made, as well as higher leverage strategies, but all were eventually discarded.

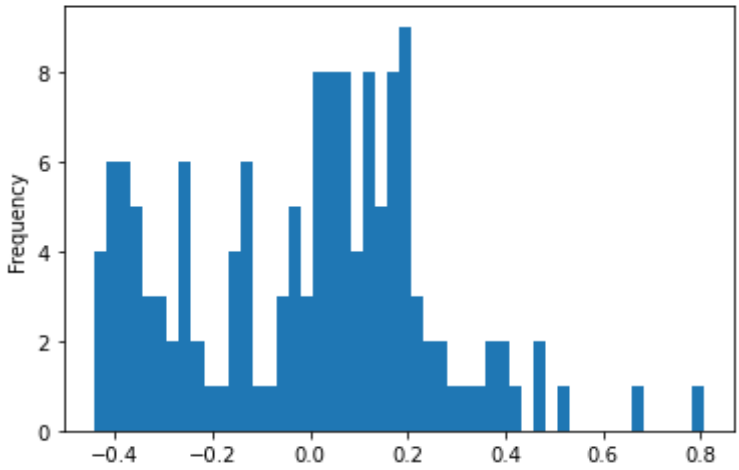
7. Results and Discussion

In this chapter we will summarize the finding in this study with a main focus on the forecasting ability of our machine learning algorithms and our trading strategies.

7.1 News Analysis and Event Study

In our collected data about oil news headlines, we found that an average sentiment score has a positive correlation to Brent Oil Futures prices of roughly 0.25, we also found that the price of Brent Oil Futures has a negative correlation of about -0.14 to news volume, which could be explained by that news agencies might have an incentive to publish more negative news than positive news as negative news is more attention grabbing. This suggested explanation is also supported by the fact that the sentiment scores of news is heavily left skewed as seen when we plot the monthly frequency of the average sentiment score.

Figure 8: Average Frequency of Sentiment Scores, Monthly



Further we find that the average number of news headlines is growing, with most news headlines between the months of March and September. The average number of news headlines published each year is 2336, the average number of news headlines published each month is 195 and the average number of news headlines published each day is about 6.

In our event study we find that the return path of extreme positive sentiment events starts, on average, to increase from day t-10 and up until day t+3, from where it seems the news is turning

stale. For extreme negative sentiment events, we see a decrease from day t-9 until the day of the event. Somewhat surprisingly, extreme negative sentiment events seem to increase from the event day until day t+4, from where it again continues its decline. But as seen in figure 4 from chapter 4, extreme positive sentiment events seem to be incorporated into prices rather quickly.

7.2 Machine Learning Models

We assess the results from our machine learning classifying models/algorithms by the key metrics accuracy, recall, precision and F1-score. The results are summarized in the three tables below.

Table 3: Weighted average of positive prediction scores

Model	Recall	Precision	F1-Score
SVC			
RBF kernel	**0.59**	**0.63**	**0.61**
Poly kernel	0.58	0.61	0.60
XGBoost	0.57	0.64	0.60
LR	**0.64**	**0.62**	**0.63**
RFC	0.55	0.64	0.59
ANNs			
Tanh	0.55	0.58	0.56
ReLU	0.43	0.59	0.50
Hybrid 1 - LR & SVC	0.53	0.66	0.59

Table 4: Weighted average of negative prediction scores

Model	Recall	Precision	F1-Score
SVC			
RBF kernel	0.59	0.54	0.57
Poly kernel	0.55	0.53	0.54
XGBoost	0.62	0.54	0.58
LR	0.58	0.56	0.57
RFC	**0.63**	**0.54**	**0.58**
ANNs			
Tanh	0.53	0.50	0.51
ReLU	0.65	0.49	0.56
Hybrid 1 - LR & SVC	**0.68**	**0.53**	**0.60**

Table 5: Weighted average of both predicted classes scores

Model	Accuracy	Recall	Precision	F1-Score
SVC				
RBF kernel	0.59	0.59	0.59	0.59
Poly kernel	0.57	0.57	0.57	0.57
XGBoost	0.59	0.59	0.60	0.59
LR	**0.60**	**0.60**	**0.60**	**0.60**
RFC	0.59	0.59	0.59	0.59
ANNs				
Tanh	0.55	0.54	0.54	0.54
ReLU	0.53	0.54	0.54	0.53
Hybrid 1 - LR & SVC	**0.59**	**0.60**	**0.61**	**0.60**

Accuracy = (True Positives + True Negatives) / (True Positives + False Positives + True Negatives + False Negatives); Recall = True Positives / (True Positives + False Negatives); Precision = True Positives / (True Positives + False Positives); F1-Score = 2(Recall * Precision) / (Recall + Precision)*

From the results in table 3 and 4, we can see that all the models are on average significantly better at predicting future positive trading days, compared to negative trading days, an exception is the ANN with ReLU activation function which has a 6% higher F1-score for negative predictions. A reason for the bias in prediction could be that the there are fewer examples in the training data of patterns between the features that result in negative trading days than for positive trading days, which makes it harder for the models to correctly classify negative trading days. The model has simply been trained more on classifying positive trading days.

We also found with our robustness check that our collected news sentiment on oil news headlines do affect the model's ability to predict future positive and negative trading days, with an average increase of 4% in F1-score and accuracy.

7.3 Trading Strategies

To put the results of our ML models into more financial terms, we conducted a couple of simple trading strategies. For both trading strategies, the ensemble model consisting of LR and SVM w/rbf kernel is utilized.

In the first trading strategy we buy Brent Oil Futures if the model predicts that t+1 will be a positive trading day, else we put all of our capital in cash. This results in a daily average return of 0.20% and a Sharpe ratio of 0.11, which beats a buy-and-hold strategy which gives a daily average return of 0.11% and a Sharpe ratio of 0.04. With a position of 2x leverage, this same trading strategy gives a daily average return of 0.38% and a Sharpe ratio of 0.10. For trading strategy 1, transactions costs of 0.75% were implemented and resulted in a lower Sharpe ratio of 0.10.

Figure 9: Trading Strategy 1

In our second trading strategy we buy Brent Oil Futures if the model predicts that t+1 and t+2 will be positive trading days, else we put all of our capital in cash. This results in a daily average return of 0.03% and a Sharpe ratio of 0.02. With a position of 2x leverage, this same trading strategy gives a daily average return of 0.04% and a Sharpe ratio of 0.01. The reason this trading strategy fails to beat any of the others is that it missed a lot of positive trading days, as the two next trading days has to be labeled as positive for it to initiate a buy command.

Figure 10: Trading Strategy 2

Note that we do not include extra costs of using leverage, which does not reflect a real-world application. But as we find that the trading strategies without leverage yield better results, one could possibly discard the idea of using leverage at all, as this would also inflict extra costs, and especially in day-to-day trading and intraday trading.

Table 6: Trading Strategy Results

	Buy-and-Hold Benchmark	**Trading Strategy 1**	**2x Leverage**	**Trading Strategy 2**	**2x Leverage**
Average Daily Return	0.11%	0.20%	0.38%	0.03%	0.04%
Standard Deviation	2.57%	1.77%	3.59%	1.46%	3.0%
Maximum Daily Loss	-14.11%	-15.21%	-33.15%	-15.21%	-33.15%
Sharpe Ratio	0.04	**0.11**	**0.10**	0.02	0.01

Table 6 above summarizes the performance of the trading strategies over the out-of-sample testing period of two trading years.

8. Conclusion and Limitations

In this study, we have analyzed how sentiment on oil news headlines affects the returns of brent oil futures and used sentiment features to create several multivariate time-series classification models utilizing machine learning techniques. Finally, we implemented an ensemble model on two simple trading strategies to show how the model could be used in a real-world financial application. A trading strategy based on a one-day forecast beats a benchmark of buy-and-hold and offers a higher return as well as a higher Sharpe ratio.

In our event study we found that sentiment from extreme events is incorporated rather quicky, which would make it hard to day trade on sentiment alone. However, the event study does not consider longer time periods, which could potentially lead to other conclusions.

Our model's prediction ability is heavily affected by the overall health of the market, and thus provides little protection against systematic risk. However, for traders dealing with brent oil futures, the models researched in this thesis can help achieve a higher average return and risk-

adjusted returns. Limitations in regard to the ML models come in the form of computational requirement as performing grid search and other hyperparameter tuning of the ANN models shows to require a lot of processing time, optimally one would optimize the hyperparameters of the ANNs to achieve better results.

Some limitations should be discussed around the trading strategies, as the test period from the end of 2020 until the end of 2022 is considered very good years for oil. The trading strategies should be backtested in bad years as well to ensure the robustness of the model. The fact that the model is better at predicting future positive trading days than negative trading days means that we can expect a worse performance when considering test periods in bear markets.

This study successfully shows that we can use the state-of-the-art FinBERT model to gather sentiment scores from news headlines related to oil that captures the current market sentiment. Further, we show that these sentiment scores can be used as features in various ML models for classification tasks. Some limitations come to the computational requirement of using FinBERT, in this study we use news headlines, but by using the full articles one could take advantage of the extra information in the text. And even though FinBERT is viewed as the best sentiment model to this date, this might not hold for sentiment extracted from commodity news. Thus, other sentiment extraction methods could prove for better results. One could also train the FinBERT model further on financial text specifically related to oil.

This study has contributed to the research on how we can utilize NLP and ML models to improve standard econometric forecasting models for predicting the future returns of brent oil futures. An ensemble model was used to conduct trading strategies; according to our knowledge, this has not been attempted in recent literature.

Further research on additional input features of the model could help create a more reliable model that can more confidently make predictions in both bull and bear markets. Further research could also be conducted on backtesting more complex trading strategies, as this study only showed a simple approach.

Bibliography

Abramson, B., & Finizza, A. (1995). Probabilistic forecasts from probabilistic models: A case study in the oil market. *International Journal of Forecasting*, *11*(1), 63–72.

Ammann, M., Frey, R., & Verhofen, M. (2014). Do Newspaper Articles Predict Aggregate Stock Returns? *Journal of Behavioral Finance*, *15*(3).

Bai, Y., Li, X., Yu, H., & Jia, S. (2022). Crude oil price forecasting incorporating news text. *International Journal of Forecasting*, *38*(1), 367–383.

Baker, M., & Wurgler, J. (2006). Investor Sentiment and the Cross-Section of Stock Returns. *The Journal of Finance*, *61*(4), 1645–1680.

Baker, M., & Wurgler, J. (2007). Investor sentiment in the stock market. *Journal of Economic Perspectives*, *21*(2), 129–152.

Black, F. (1986). Noise. *The Journal of Finance*, *41*(3), 528–543.

Borovkova, S., & Mahakena, D. (2015). News, volatility and jumps: The case of natural gas futures. *Quantitative Finance*, *15*(7), 1217–1242.

Calomiris, C. W., Melek, N. Ç., & Mamaysky, H. (2021). Predicting the Oil Market. *Working Paper 29379*.

Chan, W. S. (2003). Stock price reaction to news and no-news: Drift and reversal after headlines. *Journal of Financial Economics*, *70*(2), 223–260.

Chen, S.-S., & Chen, H.-C. (2007). Oil prices and real exchange rates. *Energy Economics*, *29*(3), 390–404.

Cifarelli, G., & Paladino, G. (2010). Oil price dynamics and speculation: A multivariate financial approach. *Energy Economics*, *32*(2), 363–372.

Clements, A. E., & Todorova, N. (2016). Information flow, trading activity and commodity futures volatility. *Journal of Futures Markets*, *36*(1), 88–104.

Daneshvar, A., Ebrahimi, M., Salahi, F., Rahmaty, M., & Homayounfar, M. (2022). Brent Crude Oil Price Forecast Utilizing Deep Neural Network Architectures. *Computational Intelligence and Neuroscience*. /

Das, S. R., & Chen, M. Y. (2007). Yahoo! For Amazon: Sentiment Extraction from Small Talk on the Web. *Management Science*, *53*(9), 1375–1388.

Dellavigna, S., & Pollet, J. M. (2009). Investor Inattention and Friday Earnings Announcements. *The Journal of Finance*, *64*(2), 709–749.

Dogu, A. (2019). FinBERT: Financial Sentiment Analysis with Pre-trained Language Models. *ArXiv Preprint ArXiv:1908.10063*.

Ellwanger, R., Alquist, R., & Jin, J. (2020). The effect of oil price shocks on asset markets: Evidence from oil inventory news. *The Journal of Futures Markets*, *40*(8), 1212–1230.

Engle, R. F., Giglio, S., Kelly, B., Lee, H., & Stroebel, J. (2020). Hedging Climate Change News. *The Review of Financial Studies*, *33*(3), 1184–1216.

Engle, R. F., & Ng, V. K. (1993). Measuring and Testing the Impact of News on Volatility. *The Journal of Finance*, *48*(5), 1749–1778.

Fama, E. F. (1970). Efficient capital markets: A review of theory and empirical work. *The Journal of Finance*, *25*(2), 383–417.

Fan, J., Xue, L., & Zhou, Y. (2021). How Much Can Machines Learn Finance From Chinese Text Data?

Feuerriegel, S., & Prendinger, H. (2016). News-based trading strategies. *Decision Support Systems*, *90*, 65–74.

Garcia, D. (2013). Sentiment During Recessions. *Journal of Finance*, *68*, 1267–1300.

Gentzkow, M., Kelly, B., & Taddy, M. (2019). Text as Data. *Journal of Economic Literature*, *57*(3)

Ghoddusi, H., Creamer, G. G., & Rafizadeh, N. (2019). Machine Learning in Energy Economics and Finance: A Review. *Energy Economics*, *81*, 709–727.

Gong, X., Guan, K., & Chen, Q. (2022). The role of textual analysis in oil futures price forecasting based on machine learning approach. *Journal of Futures Markets*, *42*(10), 1987–2017.

Haidar, I., Kulkarni, S., & Pan, H. (2008). Forecasting Model for Crude Oil Prices Based on Artificial Neural Networks. *2008 International Conference on Intelligent Sensors, Sensor Networks and Information Processing*, 103–108.

Hautsch, N., & Groß-Klußmann, A. (2011). When machines read the news: Using automated text analytics to quantify high frequency news-implied market reactions. *Journal of Empirical Finance*, *18*(2), 321–340.

Heston, S. L., & Sinha, N. R. (2017). News vs. Sentiment: Predicting Stock Returns from News Stories. *Financial Analysts Journal*, *73*.

Hillert, A., Jacobs, H., & Müller, S. (2014). Media Makes Momentum. *The Review of Financial Studies*, *27*(12), 3467–3501.

Jegadeesh, N., & Wu, D. (2013). Word power: A new approach for content analysis. *Journal of Financial Economics*, *110*(3), 712–729.

Ke, Z., Kelly, B., & Xiu, D. (2019). Predicting Returns with Text Data. *Working Paper 26186*.

Kearney, C., & Liu, S. (2014). Textual sentiment in finance: A survey of methods and models. *International Review of Financial Analysis*, *33*, 171–185.

Kelly, B., Manela, A., & Moreira, A. (2019). Text Selection. *Journal of Business and Economic Statistics*, 859–879.

Kulkarni, S., & Haidar, I. (2009). Forecasting Model for Crude Oil Price Using Artificial Neural Networks and Commodity Futures Prices. *International Journal of Computer Science and Information Security*, *2*(1).

Labonte, M. (2004). The effects of oil shocks on the economy: A review of the empirical evidence. *Congressional Research Service, Library of Congress*.

Lean, Y., Wang, S., & Lai, K. K. (2005). A rough-set-refined text mining approach for crude oil market tendency forecasting. *International Journal of Knowledge and Systems Sciences 2.1*, *2*(1), 33–46.

Lei, Y., & Gao, S. (2017). A new approach for crude oil price prediction based on stream learning. *Geoscience Frontiers*, *8*(1), 183–187.

Li, J., Xu, Z., Yu, L., & Tang, L. (2016). Forecasting Oil Price Trends with Sentiment of Online News Articles. *Procedia Computer Science*, *91*, 1081–1087.

Li, X., Shang, W., & Wang, S. (2019). Text-based crude oil price forecasting: A deep learning approach. *International Journal of Forecasting*, *35*(4), 1548–1560.

Li, Y., Jiang, S., Li, X., & Wang, S. (2021). The role of news sentiment in oil futures returns and volatility forecasting: Data-decomposition based deep learning approach. *Energy Economics*, *95*, 105140.

Lizardo, R. A., & Mollick, A. V. (2010). Oil price fluctuations and US dollar exchange rates. *Energy Economics*, *32*(2), 399–408.

Marty, T., Vanstone, B., & Hahn, T. (2020). News media analytics in finance: A survey. *Accounting & Finance*, *60*(2), 1385–1434.

Mozetič, I., Grčar, M., Caldarelli, G., Aleksovski, D., & Ranco, G. (2015). The Effects of Twitter Sentiment on Stock Price Returns. *PloS One*, *10*(9).

Noble, W. S. (2006). What is a support vector machine? *Nature Biotechnology*, *24*(12), 1565–1567.

Sadorsky, P. (1999). Oil price shocks and stock market activity. *Energy Economics*, *21*(5), 449–469.

Shi, T., Kang, K., Choo, J., & Reddy, C. K. (2018). Short-text topic modeling via non-negative matrix factorization enriched with local word- context correlations. *Proceedings of the 2018 World Wide Web Conference*, 1105–1114.

Sisk, J., & Leinweber, D. (2011). Event-Driven Trading and the "New News." *The Journal of Portfolio Management*, *38*(1), 110–124.

Smales, L. A. (2014). News sentiment in the gold futures market. *Journal of Banking & Finance*, *49*, 275–286.

Song, Q., Liu, A., & Yang, S. Y. (2017). Stock portfolio selection using learning-to-rank algorithms with news sentiment. *Neurocomputing*, *264*, 20–28.

Souma, W., Vodenska, I., & Aoyama, H. (2019). Enhanced news sentiment analysis using deep learning methods. *Journal of Computational Social Science*, *2*(1), 33–46.

Sun, L., Najand, M., & Shen, J. (2016). Stock return predictability and investor sentiment: A high-frequency perspective. *Journal of Banking & Finance*, *73*, 147–164.

Tetlock, P. C. (2007). Giving Content to Investor Sentiment: The Role of Media in the Stock Market. *The Journal of Finance*, *62*(3), 1139–1168.

Toutanova, K., Lee, K., Chang, M.-W., & Devlin, J. (2019). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *ArXiv Preprint ArXiv:1810.04805*.

Tumarkin, R., & Whitelaw, R. F. (2001). News or Noise? Internet Postings and Stock Prices. *Financial Analysts Journal*, *57*(3), 41–51.

Uhl, M. W., Pedersen, M., & Malitius, O. (2015). What's in the News? Using News Sentiment Momentum for Tactical Asset Allocation. *The Journal of Portfolio Management*, *41*(2), 100–112.

Wang, C. (2001). Investor Sentiment and Return Predictability in Agricultural Futures Markets. *Journal of Futures Markets*, *21*, 929–952.

Wang, C. (2003). Investor Sentiment, Market Timing, and Futures Returns. *Applied Financial Economics*, *13*(12), 891–898.

Wang, C., Wang, T., Yuan, C., & Rong, J. Y. (2022). Learning to trade on sentiment. *Journal of Economics and Finance*, *46*(2), 308–323.

Wang, J.-J., Wang, J.-Z., Zhang, Z.-G., & Guo, S.-P. (2012). Stock index forecasting based on a hybrid model. *Omega*, *40*(6), 758–766.

Wex, F., Widder, N., Liebmann, M., & Neumann, D. (2013). Early warning of impending oil crises using the predictive power of online news stories. *2013 46th Hawaii International Conference on System Sciences*, 1512–1521.

Yang, L., Han, L., & Yin, L. (2018). Does news uncertainty matter for commodity futures markets? Heterogeneity in energy and non-energy sectors. *Journal of Futures Markets*, *38*(10), 1246–1261.

Yazdani, S. F., Murad, M. A. A., Sharef, N. M., Singh, Y. P., & Latiff, A. R. A. (2017). Sentiment Classification of Financial News Using Statistical Features. *International Journal of Pattern Recognition and Artificial Intelligence*, *31*(3), 1750006.

Ye, M., Zyren, J., & Shore, J. (2005). A monthly crude oil spot price forecasting model using relative inventories. *International Journal of Forecasting*, *21*(3), 491–501.

Zhang, Y.-J., & Wei, Y.-M. (2010). The crude oil market and the gold market: Evidence for cointegration, causality and price discovery. *Resources Policy*, *35*(3), 168–177.

Zhao, Y., Li, J., & Yu, L. (2017). A deep learning ensemble approach for crude oil price forecasting. *Energy Economics*, *66*, 9–16.

Zhou, Y., & Yang, J. (2020). Return and volatility transmission between China's and international crude oil futures markets: A first look. *Journal of Futures Markets*, *40*(6), 860–884.