Time Series Forecasting

Abstract. Our project focuses on time and demand forecasting. Our objective is to explore how traditional time series models perform against the latest neural networks. Also, our project focuses on how traditional time series forecasting models apply to different types of data and when to select which models based on factors such as seasonality, trend, etc. Also, our focus in this project was to identify datasets that could be used for demand and price forecasting. We manually created the dataset from the OPEC repository by selecting the most relevant features and datasets. This project aims to provide a starting point for predicting oil prices and demand, which can be utilized by countries worldwide for fixing their prices and estimating demand.

Keywords: Time series forecasting, Price forecast, demand forecast, OPEC repository

1 Introduction

The motivation behind this project was to understand time series forecasting and compare traditional models against neural networks. We also wanted to create a dataset that can be used to predict oil prices and demand. This project was aimed at understanding time series forecasting while also building a dataset from scratch for the prediction. Our aim in this project is to build a reasonably good forecasting model that can be utilized by countries for their price and demand estimates.

Currently, predicting oil prices and demand is a major area of research for oil manufacturing companies worldwide. Many multinational companies are utilizing machine learning models to understand oil demand and also predict their prices. Our goal in this project is to identify the best model for price and demand forecasts.

The application of our project is price and demand forecasting for oil prices. Multinational companies can utilize our framework for price and demand forecasts.

The primary challenge for us was dataset creation. Finding a relevant dataset that can be utilized for price and demand forecasts is rare. We found a repository called the OPEC repository, which we utilized for this project. The repository had 50+ datasets that we had to combine by making the relevant joins. After creating the dataset, we then selected the relevant features for our project. Also, an additional challenge was identifying the correct models and selecting the required parameters for our project.

Our dataset is curated from the OPEC repository. Each dataset in the OPEC repository is at a country-year level. Also, the dataset is in matrix format, which we converted to a data frame before making the join.

We approached our problem statement by first doing a thorough Exploratory Data Analysis on our dataset to understand our data. We then identified the relevant features that affect our target feature, which is the spot prices, since we had so many features in our dataset. We utilized the correlation matrix to determine relevant features for our dataset. After identifying the relevant features, we understood the various models that are used for time series data. We selected models that are not utilized for predicting seasonal trends since our data is non-seasonal due to the data being at a yearly level. Once we identified the models, we predicted the price and demand at a country level.

After running our model, we saw that Holt-Winters linear model performed the best with ARIMA in the second place. Both these models are adaptable for non-seasonal datasets, and since our data is at a yearly level, these models were a good fit since our dataset has non-seasonal data. Something interesting we observed was that these models performed better than our neural network and general machine learning models.

2 Problem Statement

Demand and price forecasting are major factors in determining oil prices for multinational companies that sell oil. Our problem statement was to create a dataset with factors affecting oil price and demand and to develop an effective framework while exploring different models for price and demand forecasting. In today's time, many big multinational companies, due to their old methods, have not yet completely adopted machine learning in price and demand forecasting. Our objective is to develop a framework that can effectively factor in features such as demand, production, refinery capacities, and exports in effectively making a forecast. Some of the tasks we wish to achieve through this activity include creating a dataset from the OPEC repository with valid features and identifying the algorithms that are good for price and demand forecasting for our dataset. Some of the experiments we carried out involved testing different models against the selected dataset and using different timelines and timeframes to get the best possible predictions of price and demand.

3 Related Work

The paper "Crude oil price analysis and forecasting: A perspective of the 'new triangle'" discusses the core factors affecting crude oil prices. Also, it utilizes Google Trends and its effect on crude oil prices. It uses the dynamic Bayesian structural time series model (DBSTS) for its prediction.

4 Background Concepts(/Knowledge) (/Preliminary) (Optional)

1. Time Series Data - Time series data are data points recorded at consecutive points in time. Time series data are used to observe historical trends, understand the current conditions, and make future forecasts.

2. Stationary Data - Time series data is said to be stationary if the statistical properties such as mean, variation, and autocorrelation are constant over time. Many time series models, such as ARIMA, assume that time series data is stationary.

3. Autocorrelation - Autocorrelation is a statistical measure that represents the degree of similarity between the given time series and its lagged version for successive time intervals. There can be positive correlation, negative correlation, or no correlation at all. Positive autocorrelation means that, for example, if there are stock prices and the prices were higher than average for one time period, then they will be higher than average for the next time period as well.

4. Autoregressive Component (p): This states the influence of the previous time steps on the current time step. In an AR model, 'p' represents the number of lagged observations.

5. Differencing Component (d) - This involves subtracting the previous observation from the current observation to help make the time series stationary. 'd' represents the number of differencings required to obtain stationarity.

6. Moving Average (q) - This represents the size of the moving average window.

7. Holt-Winters Model - This model is used to forecast time series data that exhibits trends and seasonality.

8. Holt-Winters Linear Model - Since our dataset does not have seasonality, we use a variation of the Holt-Winters model called the Holt-Winters linear model, which is used for modeling data with no seasonality.

9. Augmented Dickey Fuller Test (ADF Test) - This is a statistical test used to determine if the time series data is stationary.

10. Autocorrelation Function (ACF) and Partial Autocorrelation Function (PACF) - The ACF shows the correlation between the AR and MA components of the ARMA model. The PACF shows the correlation between the time series and lagged versions of itself after controlling for the values of the time series at all shorter lags.

5. Data

Data Source

The OPEC repository provides data that is relevant to the petroleum industry. It has data tables with facts and figures on member countries such as GDP, population, exports, etc. It also has data on oil trade like exports, imports, spot prices, taxes on oil, and natural gas data. There are approximately 53 tables pertaining to OPEC countries and their oil prices. This repository can be used for various tasks such as analyzing market trends, forecasting, and policy making. The data is typically in years as columns and countries as rows with the respective feature populated. We obtained this data source from the following link: https://asb.opec.org/data/ASB_Data.php

Data Size:

The data has 560 rows with years from 1983 to 2022 and there are 14 regions for which we are making the forecasts. Also our dataset has 27 features.

Attributes and description:

1. Country: The name of the country to which the data row pertains.
2. Year: The year in which the data was recorded.
3. Spot_Price_72: The spot price crude oil
4. Crude_oil_exports_52: The total volume of crude oil exported by the country in the given year.
5. Demand: The total demand for energy or oil within the country for the specified year.
6. Exports: The total value or volume of all exports from the country in the given year.
7. Gas_Reserve: The proven reserves of natural gas in the country for the year specified.
8. Imports: The total value or volume of all imports to the country in the given year.

9. Production: The total production of energy or oil within the country for the specified year.
10. Oil_demand_46: The specific demand for oil in the country for the given year.
11. Oil_exports_dest_51: The volume of oil exports destined for specific regions or countries.
12. Refinery_capacity_41: The total capacity of oil refineries in the country for the year specified.
13. Refinery_throughput_43: The volume of crude oil processed by refineries in the country during the given year.
14. Tax: The tax revenue generated from the oil and gas sector in the country for the specified year.
15. Tax_To_CIF_Ratio: The ratio of tax to the cost, insurance, and freight (CIF) value of imported goods.
16. World_Production: The global production of oil or energy for the given year.
17. ActiveRigs-table3.2: The number of active drilling rigs in the country for the year specified.
18. AccountBalance-table2.7: The country's account balance related to the oil and gas sector for the given year.
19. WellsCompleted-table3.3: The number of oil and gas wells completed in the country during the specified year.
20. Population-table2.1: The total population of the country in the given year.
21. petroleumExport-table2.5: The value or volume of petroleum exports from the country for the specified year.
22. GDP-table2.2: The Gross Domestic Product of the country for the given year.
23. GDPGrowthRate-table2.3: The growth rate of the country's GDP for the specified year.
24. ExportValue-table2.4: The total value of all exports from the country for the given year.
25. crudeOilReserve-table3.1: The proven crude oil reserves in the country for the year specified.
26. CrudeOilProduction-table3.5: The total crude oil production in the country for the given year.
27. bioFuelProduction-table3.7: The production volume of biofuels in the country for the specified year.

Characteristics of the data:

The dataset has global oil-related features for the following countries:

Based on the information you provided, here is a list of countries:

1. Algeria
2. Angola
3. Congo
4. Equatorial Guinea
5. Gabon
6. IR Iran (Iran)
7. Iraq
8. Kuwait
9. Libya
10. Nigeria
11. Saudi Arabia
12. United Arab Emirates
13. Venezuela
14. OPEC (Organization of the Petroleum Exporting Countries)

Time Period:

The period for this dataset ranges from 1983 to 2022 and the dataset is a time series dataset

Data Types:

The dataset contains mostly numerical data depicting features such as prices, quantity, trends, etc.

Data Quality:

The dataset contains NA values after the joins were made. Data was missing for certain countries for certain years. We replaced the NA values using backward fill and forward fill.

Data Granularity:

The data is at a country/year level or the granularity of the data is at country and year

Design of analytics base table:

Features with data type:

1. Country: object
2. Year: int
3. Spot_Price_72: float
4. Crude_oil_exports_52: float
5. Demand: float
6. Exports: float
7. Gas_Reserve: float
8. Imports: float
9. Production: float
10. Oil_demand_46: float
11. Oil_exports_dest_51: float
12. Refinery_capacity_41: float
13. Refinery_throughput_43: float
14. Tax: float
15. Tax_To_CIF_Ratio: float
16. World_Production: float
17. ActiveRigs-table3.2: float
18. AccountBalance-table2.7: float
19. WellsCompleted-table3.3: float
20. Population-table2.1: int
21. petroleumExport-table2.5: float
22. GDP-table2.2: float
23. GDPGrowthRate-table2.3: float
24. ExportValue-table2.4: float
25. crudeOilReserve-table3.1: float

26. CrudeOilProduction-table3.5: float
27. bioFuelProduction-table3.7: float

Target:

Price : float

Demand: float

6. Data exploration

Exploratory Data Analysis (EDA)

The following details describe an Exploratory Data Analysis (EDA) of a dataset that covers important metrics in the oil and gas sector. The data spans from 1983 to 2022. The analysis includes several steps such as data cleaning, statistical summaries, time series analysis, and visual representations. The purpose of this analysis is to gain insights into trends, patterns, and anomalies within the data.

1. Introduction

   This section introduces the scope and significance of the EDA in understanding the dynamics of the oil and gas sector. It outlines the dataset's span and the variables analyzed.

2. Data Overview

   The dataset comprises 560 entries across 25 columns, including numerical and categorical data. Key columns include 'Country', 'Year', 'Spot_Price', 'Production', 'Exports', and 'Imports'. Notably, several columns initially contained non-numeric characters and missing values.

3. Methodology:

   The EDA involved several key steps:
   - Data Cleaning: Conversion of numeric information from string to appropriate numeric types. Management of non-numeric characters and missing values.
   - Statistical Summary: Analysis of the data's central tendency, dispersion, and distribution shape.
   - Time Series Analysis: Exploration of trends and patterns over time, leveraging 'Year' as the primary axis.
   - Visualizations: Generation of plots to visualize overall and country-specific trends.

4. Results and Discussion

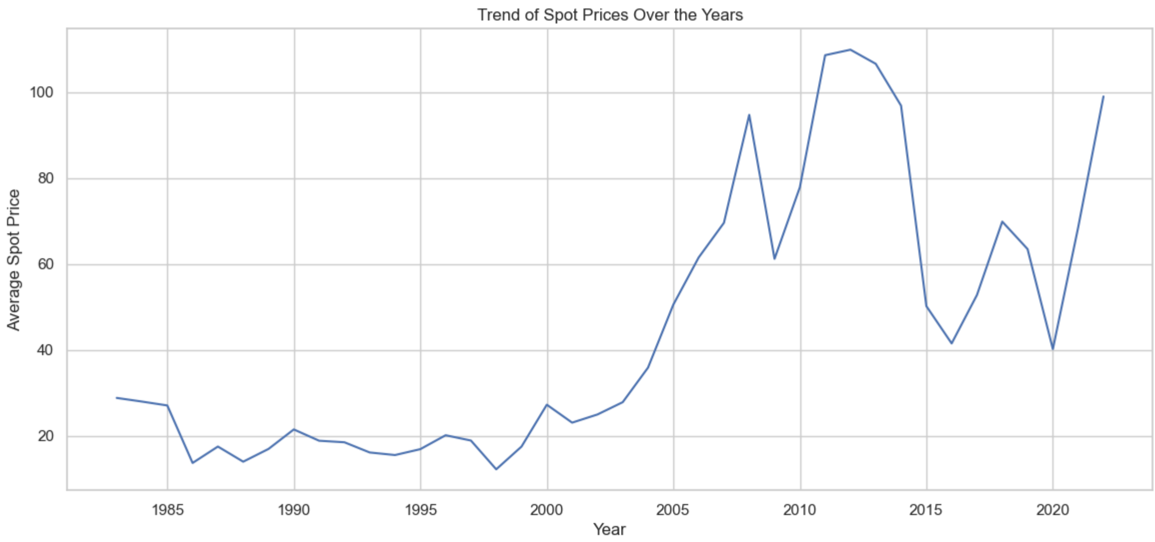
   The analysis yielded several findings:

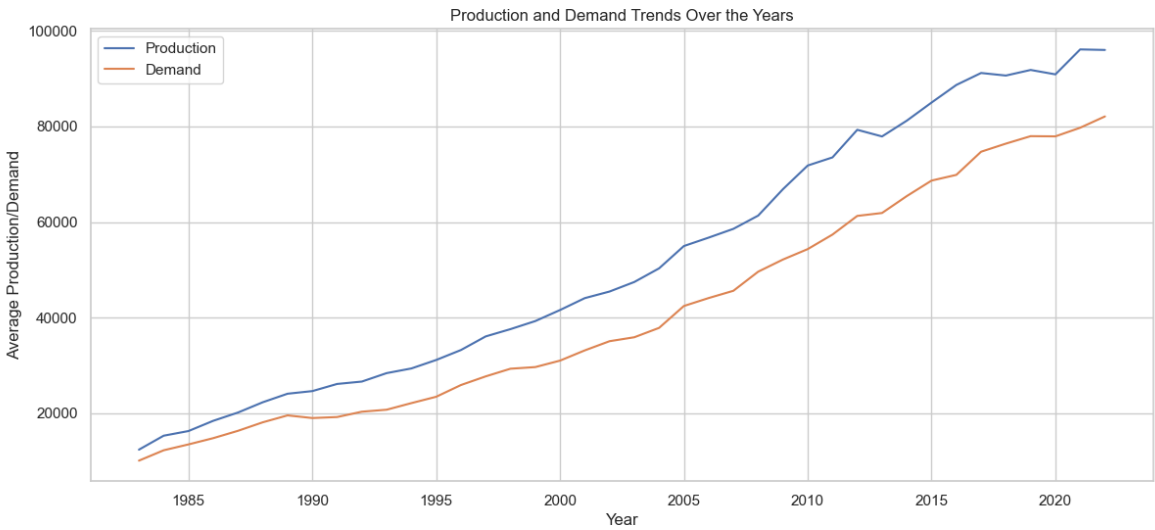
   - Trends Over Time: Fluctuations in spot prices, production, and demand were observed, reflecting global economic and political events.
   - Country-Specific Analysis: Comparative trends across different countries revealed diverse economic and resource management strategies.
   - Correlation Analysis: Potential relationships between different variables were identified.
   - Outlier Detection: Identification of anomalies that could influence future models and analyses.

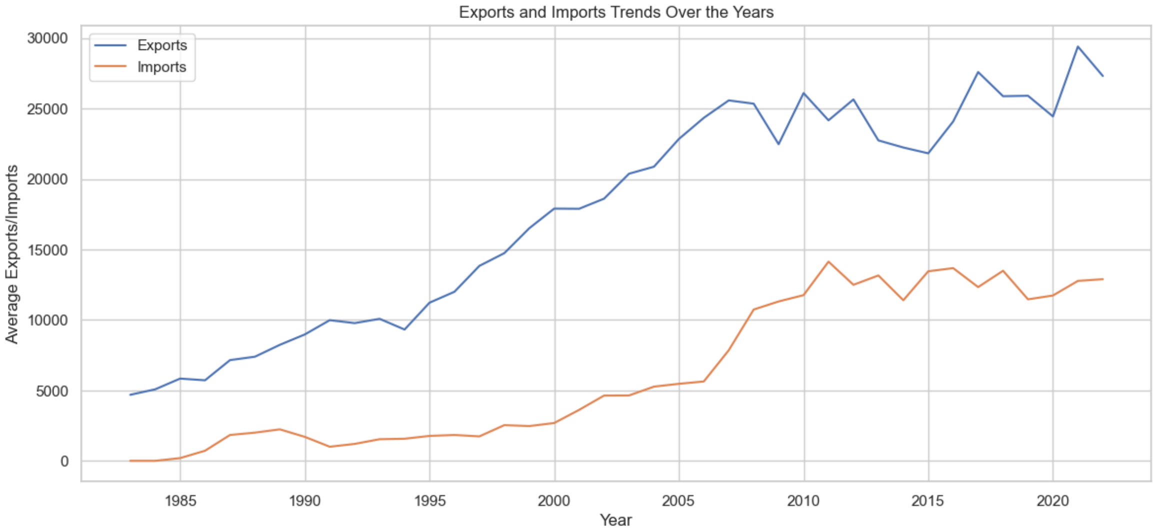
5. Visualizations

   Several plots were generated to illustrate the trends. These include:

- Overall trends in spot prices, production, and trade metrics.
- Country-specific trends highlight the unique dynamics of each country in the context of oil and gas production, trade, and pricing.






Exports and Imports Trends Over the Years
Exports
Imports
Average Exports/Imports
30000
25000
20000
15000
10000
5000
0
1985
1990
1995
2000
2005
2010
2015
2020
Year

7. Data preprocess

In the data preprocessing, we did the following:

1. We first converted each of the Excel workbooks to CSVs so that they could be imported as data frames for further preprocessing.
2. We also removed unwanted text from the CSV files like descriptions and explanations. We did this manually.
3. Once the CSV files were clean, we imported them into our Jupyter notebooks.
4. Once imported, the data was in the format of the year as a column and the country as the row with the respective feature as the populated value.
5. We then melted the tables so that we got year, country, and feature as the respective columns so that these would be in the correct format to input into our time-series model.
6. We then joined the 27 tables starting with spot prices, which is our target, and our left table using a left join. We used this logic since we wanted the feature values to be populated only for countries which were present in the spot prices table and not all the countries' data which we had.
7. Once all the joins were made, we replaced the NA values using forward fill and backward fill. Time-series data cannot handle NA values, and therefore we had to use this methodology.
8. We created a preprocessing feature where it does the following:
8. Removes commas from numeric strings.
9. Converts the cleaned strings to numeric values.
10. Fills any remaining missing values using both forward and backward filling, ensuring that the column is devoid of NaN values.
9. Then we created a correlation matrix for the features. Since time-series data is sensitive to the feature inputs, we selected features which had a correlation of 0.4 or above with the target feature, which was spot price. After doing this, we had the following features in our dataset:

    'Spot_Price_72'
    'Demand'
    'Imports'
    'Production'
    'Oil_demand_46'
    'Oil_exports_dest_51'
    'Refinery_capacity_41'
    'Refinery_throughput_43'
    'World_Production'

8. Methodology (/Proposed Methods/ Approach/ Procedure)

We tested 5 models for our time series analysis. We tested a traditional machine learning model (SVR), a BILSTM, CNN, and 2 traditional time series forecasting models which are ARIMA and Holt Winters.

8.1. ARIMA model

1. We chose the ARIMA model since our data doesn't have seasonality. Our data contains granularity at a yearly level; therefore, seasonality is missing in our data. ARIMA is designed to fit non-seasonal data.
2. ARIMA only works on stationary data, and our data is non-stationary.
3. We determined our data is non-stationary by using the augmented Dickey-Fuller test (ADF).

ADF Statistic: -1.444517373831189
p-value: 0.5607578133100624
Critical Values:
1%: -3.442405720052601

5%: -2.866857804790472
10%: -2.5696022094099824

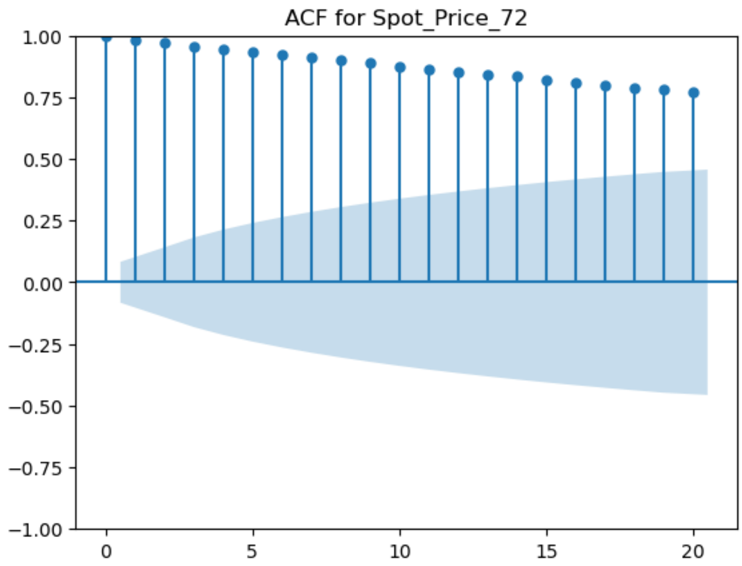
4. The ADF statistic is a test statistic. The more negative values indicate a stronger rejection of our null hypothesis. Our null hypothesis is that the data is not stationary.
5. p-value - the p-value tells the probability of observing the data that the null hypothesis is true. A lower p-value less than 0.05 suggests we can reject the null hypothesis.
6. Critical values – These values are the confidence levels. If the ADF statistic is more negative than the critical value, the null hypothesis can be rejected at that confidence level.
7. When we apply the function to Spot prices for the first time based on the above results, the data is non-stationary, which means the null hypothesis is true based on the above values. The null hypothesis states that the data is not stationary.
8. Therefore, we apply differencing to the time series data to check if the data becomes stationary.
9. The differencing is between time periods t and t-1.
10. Once we apply the differencing, we recheck the ADF test. The second time we obtained the null hypothesis to be false based on the below values.

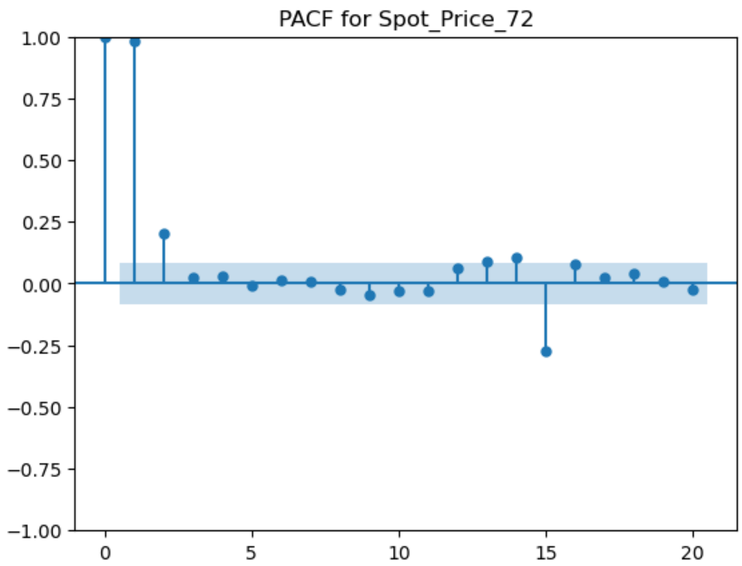
ADF Statistic: -4.847698062846962
p-value: 4.4040982765033434e-05
Critical Values:
1%: -3.442405720052601
5%: -2.866857804790472
10%: -2.5696022094099824

11. Since the ADF statistic is significantly less than the critical values and the p-value is very low, we can reject the null hypothesis. This means that the differenced time series is stationary.
12. Differencing has to be done until the ADF test proves that the dataset is stationary.
13. After that, we select the p, d, and q values.
14. To determine P, D, and Q values:
    - P - Current value dependent on how many prior intervals
    - D – How many times differencing should be applied to make the data stationary
    - Q – Determines the dependency of the current value on past forecast errors
15. We determined P, D, and Q values by using ACF, PACF plots, and trial & error. From these methodologies, we determined P = 2, D = 0, and Q = 2

16. From the PACF plot, we determine the value of p. This is done by checking where the lag observation mostly becomes insignificant. We select either 1 or 2 for p values since it is after this order that the lag observation becomes insignificant.

17. From the ACF plot, it's currently a little hard to determine the value of q since the order of significant lags does not cut off, i.e., it remains constant. Therefore, we determine the value of q by trial and error.
18. The value of d will be 0 since we have already done the differencing and verified with the ADF test that the data has become stationary.
19. We split the model into an 80% train and 20% test split.
20. We evaluated the model using MSE.





8.2. Holt Winters

Holt Winters linear model is an adaptation of Holt Winters designed to fit non-seasonal data.
It has a hyperparameter where the seasonal component can be selected as none.
We chose the time period from 2000 since this was giving the best average MSE score.
Holt Winters has the following parameters:

- 'Demand' is the target variable for forecasting.
- 'trend='add'' indicates an additive trend component.
- 'seasonal=None' indicates no seasonal component.
- ‘damped_trend=False' specifies that the trend is not damped.

1. *Data Acquisition and Preprocessing:*
    - Dataset Loading: The dataset, named 'merged_dataset.csv', is loaded into a pandas DataFrame. This dataset presumably contains various economic indicators including 'Spot_Price_72'.

- Conversion to Numeric: Non-numeric columns, except 'Country' and 'Year', are converted to numeric format. This ensures that all relevant columns are in a format suitable for analysis.
- Handling Missing Values: Missing values in numeric columns are handled using forward fill and backward fill methods to maintain data continuity.
-

2. *Feature Selection and Correlation Analysis:*
    - Grouping by Year: The dataset is grouped by 'Year' and the mean of each column is calculated. This step simplifies the data, focusing on annual trends.
    - Correlation Matrix Computation: A correlation matrix of yearly means is computed to understand the relationships between different variables, especially with respect to 'Spot_Price_72'.
    - Selection Based on Correlation Threshold: Variables with a correlation of 0.4 or higher with 'Spot_Price_72' are selected as relevant features. This threshold ensures the selection of features that have a significant linear relationship with the target variable.

3. *Data Preparation for Time Series Analysis:*
    - Filtering Dataset: The analysis focuses on data from the year 2000 onwards, considering more recent data to be more relevant for current and future predictions.
    - Differencing: To ensure stationarity in the time series, the 'Spot_Price_72' series is differenced. This step is crucial for models like ARIMA that assume stationarity.

4. *Stationarity Check and Model Selection:*
    - Stationarity Testing: The Augmented Dickey-Fuller test is likely to be used to check the stationarity of the 'Spot_Price_72' series. Stationarity is a key assumption in many time series models.
    - ARIMA/SARIMAX Modeling: Given the inclusion of ARIMA and SARIMAX models in the code, it is presumed that these models would be fit to the time series data. These models are widely used for forecasting time series data and can handle both seasonal and non-seasonal patterns.

5. *Forecasting and Evaluation:*
    - Model Fitting and Forecasting: The selected ARIMA/SARIMAX models would be fitted to the 'Spot_Price_72' time series, and future values would be forecasted.
    - Model Evaluation: The Mean Squared Error (MSE) would be used to evaluate the model's performance, providing a quantitative measure of the forecasting accuracy.

6. *Visualization:*
    Plotting Time Series Data: Visualization tools like matplotlib and seaborn would be used to plot the time series data, including actual vs. forecasted values, providing a visual assessment of the model's performance.

8.3. Bidirectional Long Short-Term Memory (BiLSTM)

1. *Objective of Using BiLSTM:*
    The primary objective of employing a BiLSTM network in this study is to leverage its advanced capabilities in capturing temporal dependencies both in forward and backward directions of the time series data. This is particularly beneficial for forecasting tasks like predicting the 'Spot_Price_72', where historical context and future trends are crucial.

2. *Data Preprocessing:*
    - Data Selection: The dataset comprises various economic indicators, including 'Spot_Price_72', collected from the year 2000 onwards.
    - One-Hot Encoding: The categorical variable 'Country' is transformed using one-hot encoding to facilitate numerical processing.
    - Feature Scaling: We applied MinMaxScaler to normalize the feature set, ensuring that the model is not biased towards variables with higher magnitude.

3. *Sequence Preparation:*
   A pivotal step in preparing the dataset for BiLSTM involves converting the time series data into a supervised learning problem. This is achieved by creating sequences of data points (of length sequence_length) to predict the subsequent value in the series.

4. *BiLSTM Model Architecture:*
    - Input Layer: The model inputs are sequences of features with a specified length (sequence_length).
    - Bidirectional LSTM Layer: The core of the model consists of a bidirectional LSTM layer with 64 neurons. This layer processes the input data in both forward and backward directions, capturing intricate patterns in time series data.
    - Activation Function: We utilize the 'ReLU' (Rectified Linear Unit) activation function for its efficiency in training deep neural networks.
    - Output Layer: A Dense layer with a single neuron and a linear activation function is used to predict the continuous value of 'Spot_Price_72'.

5. *Model Compilation and Training:*
    - Optimizer: 'Adam', known for its efficiency in handling sparse gradients and adaptive learning rate, is used.
    - Loss Function: The model is trained to minimize the 'Mean Squared Error' between the predicted and actual values, making it suitable for regression tasks.
    - Training Process: The model is trained over 50 epochs with a batch size of 64, balancing the trade-off between computational efficiency and the ability to find a global minimum.

6. *Prediction and Evaluation:*
    - Inverse Transformation: Predictions made by the model are inverse-transformed to bring them back to their original scale.
    - Reshaping: Both the predicted and actual values are reshaped for performance evaluation.
    - Performance Metrics: We calculate the Mean Squared Error (MSE) for each country individually and for the overall model to assess its forecasting accuracy.

7. *Country-wise Analysis:*
   An innovative aspect of our approach involves performing a country-wise analysis. By applying a mask to the test data corresponding to each country, we evaluate the model's performance in predicting 'Spot_Price_72' for individual countries.

8. *Visualization:*

To visually assess the model's performance, we plot the actual versus predicted values of 'Spot_Price_72', offering an intuitive understanding of the model's predictive capabilities.

9. *Forecasting for 2023:*
    - Data Preparation: We prepare the latest sequences of data for each country to project the 'Spot_Price_72' for the year 2023.
    - Prediction: The model is used to predict the spot prices for each country for the year 2023, providing valuable insights into future trends.

8.4. 1D Convolutional Neural Network (1D CNN)

1. *Objective of Using 1D CNN:*
   The study employs a 1D Convolutional Neural Network to forecast the 'Spot_Price_72'. This model is chosen for its ability to detect and utilize local patterns and dependencies in time-series data, which are crucial for accurate forecasting in economic and financial contexts.

2. *Data Preprocessing:*
    - Data Filtering: The dataset includes various economic indicators, and we focus on data from the year 2000 onwards to capture recent trends more effectively.
    - One-Hot Encoding: The 'Country' categorical variable is transformed using one-hot encoding, enabling the model to process this non-numeric data efficiently.
    - Feature Scaling: The MinMaxScaler is applied to normalize both the features and the target variable, ensuring uniformity in the data range.

3. *Sequence Preparation:*
   The data is structured into sequences to fit the model's input requirements. Each sequence of length sequence_length is used to predict the subsequent value, aligning with typical time-series forecasting methodologies.

4. *1D CNN Model Architecture:*
    - Convolutional Layers: The model includes two convolutional layers, each with 64 and 128 filters, respectively. These layers, with a kernel size of 3, are designed to extract local features from the input sequences.
    - Activation Function: The 'ReLU' (Rectified Linear Unit) activation function is used for its efficiency in training deep neural networks and introducing non-linearity.
    - Pooling Layers: Following each convolutional layer, a MaxPooling layer of pool size 2 is applied to reduce dimensionality and extract dominant features.
    - Regularization: L2 regularization is implemented to prevent overfitting by penalizing large weights.
    - Flattening and Dense Layers: The flattened output is then fed into a dense layer with 50 neurons, followed by a single neuron output layer for regression.

5. *Model Compilation and Training:*
    - Optimizer: The 'Adam' optimizer is chosen for its adaptive learning rate capabilities.
    - Loss Function: 'Mean Squared Error' is used as the loss function, focusing on minimizing the average squared difference between the predicted and actual values.

- Training Parameters: The model is trained for 100 epochs with a batch size of 64, ensuring comprehensive learning from the data.

6. *Prediction and Evaluation:*
    - Inverse Transformation: Predictions are scaled back to their original range for meaningful interpretation.
    - Reshaping for Evaluation: Both predicted and actual values are reshaped to facilitate the computation of evaluation metrics.
    - Performance Metrics: The Mean Squared Error (MSE) is calculated for both individual countries and overall, providing insights into the model's accuracy and consistency.

7. *Country-specific Analysis:*
   The model's performance is also analyzed at the country level, offering a detailed view of its predictive power across different geographical regions.

8. *Visualization:*
   Graphical representations of actual versus predicted values of 'Spot_Price_72' are provided, visually demonstrating the model's predictive ability.

9. *Forecasting for 2023:*
    - Preparation for Future Prediction: Sequences representing the most recent data for each country are compiled to forecast the 'Spot_Price_72' for the year 2023.
    - Predictive Analysis: The model is utilized to project future spot prices, offering valuable foresight into upcoming economic trends.

8.5. Support Vector Regression (SVR)

1. *Objective of Using SVR:*

   The study employs Support Vector Regression (SVR), a type of Support Vector Machine (SVM) used for regression tasks. SVR is selected for its effectiveness in dealing with high-dimensional data and its ability to model non-linear relationships, which is crucial for forecasting economic indicators like 'Spot_Price_72'.

2. *Data Preprocessing:*
    - Data Selection and Filtering: The study focuses on data from the year 2000 onwards, considering the relevance of recent economic trends.
    - One-Hot Encoding: The 'Country' variable is one-hot encoded to convert categorical data into a numerical format, suitable for SVR modeling.
    - Feature Scaling: MinMaxScaler is utilized to normalize both the features and the target variable, ensuring that the model is not skewed towards higher magnitude variables.

3. *Train-Test Split:*
   The dataset is split into training and testing sets with an 80-20 ratio, ensuring sufficient data for both model training and evaluation.

4. *SVR Model Configuration:*
    - Feature Transformation: The sequences of features are flattened to fit the input requirements of the SVR model.
    - SVR Parameters: The SVR model with an RBF (Radial Basis Function) kernel is used. The 'C' parameter, representing the regularization strength, and the 'epsilon' parameter, defining the margin of tolerance where no penalty is given to errors, are set to optimize the model performance.

5. *Model Training and Prediction:*
    - Model Training: The SVR model is trained on the flattened training dataset.
    - Prediction: The model predicts the 'Spot_Price_72' values for the test dataset. These predictions are then inverse-transformed to their original scale for interpretability and evaluation.
6. *Performance Evaluation:*
   Mean Squared Error (MSE): MSE is computed to quantify the model's prediction accuracy, providing a clear metric to evaluate the performance of the SVR model.

7. *Visualization:*
   Graphical representation of actual versus predicted values of 'Spot_Price_72' is provided, offering an intuitive understanding of the model's predictive performance.

8. *Forecasting for 2023:*
    - Data Preparation: The latest data sequences for each country are prepared, ensuring that the most recent trends are included in the forecast.
    - Future Predictions: The SVR model is utilized to forecast the 'Spot_Price_72' for each country for the year 2023, providing valuable insights into future economic conditions.

9. *Country-specific Forecasts:*
   The model's ability to make predictions for individual countries offers a more nuanced and detailed understanding of regional economic trends.

9. Experiment

9.1. ARIMA Model

Some of the experiments we carried out with ARIMA included selecting the p, d, and q values:

- P: Current value dependent on how many prior intervals.

- D: How many times differencing should be applied to make the data stationary.

- Q: Determines the dependency of the current value on past forecast errors.

We used the ACF and PACF plots and also experimented by trial and error to determine the best-suited values. We have explained how we selected the ACF and PACF values in the methodology section, which was a mix of trial and error and an attempt to understand the best values from the ACF and PACF graphs.

We also tested different timeframes and finally selected the timeframe after 2000 to get the best MSE scores.

We experimented with finding the differencing between time intervals t and t-1 using the ADF test. We were able to achieve stationarity with just one time difference by rechecking the ADF test.

9.2. Holt-Winters:

We experimented with the parameters of Holt-Winters to check which one achieved the best score:

- We experimented with different trends, including additive and multiplicative. The additive trend assumes that the data is linear. This trend selection gave us the best MSE score.

- We selected "seasonal" as "None" since our data is not seasonal data and has data at a yearly level. Seasonal data is usually at a more granular level with seasonal trends.

- We selected "damped trend" as "None" since our data does not exhibit tapering when plotted, and we do not need to apply a dampening factor.

9.3. Bi-Directional Long Short-Term Memory (BiLSTM)

1. *Training and Validation Dataset:* The dataset was split into training, validation, and testing sets. The training and validation sets were used iteratively to adjust the model's parameters and prevent overfitting.

2. *Hyperparameters:*
   - Number of Layers: The architecture of the BiLSTM model includes multiple layers to capture complex temporal patterns in the data.
   - Units per Layer: Each BiLSTM layer consists of a predefined number of neurons, which determines the model's capacity to learn from the data.
   - Activation Function: Commonly used activation functions in BiLSTM include 'tanh' and 'ReLU', which help the model capture non-linear relationships.
   - Optimizer: The model is optimized using algorithms like 'adam', 'sgd', or 'rmsprop', which differ in terms of speed and performance.
   - Learning Rate: The learning rate is set to balance the speed of convergence and the risk of overshooting the minimum.
   - Epochs: The number of epochs represents the number of complete passes through the training dataset, crucial for the learning process.
   - Regularization (L1/L2): Regularization techniques, such as L1 and L2, are applied to prevent overfitting and improve the model's generalization capabilities.

3. *Prediction Model and Results*: The BiLSTM model was trained to predict 'Spot_Price_72', with performance evaluated based on metrics like MSE. The model's ability to capture time-dependent patterns was particularly noted.

4. *Tools and Algorithms:* TensorFlow and Keras were used for implementing and training the BiLSTM model, leveraging their extensive libraries and functionalities.

9.4. 1D Convolutional Neural Network (1D CNN)

1. *Training and Validation Dataset:* The dataset was divided into training, validation, and testing segments, with the validation set used to fine-tune model parameters.

2. *Hyperparameters:*
   - Number of Convolutional Layers: The model includes multiple convolutional layers to extract and learn feature hierarchies from the data.
   - Kernel Size: The size of the convolutional filters, which impacts the model's ability to capture local features.
   - Activation Function: 'ReLU' and similar functions are used to introduce non-linearity into the model, enhancing its learning capacity.
   - Dense Layer Units: Following the convolutional layers, dense layers with a defined number of neurons are used for higher-level reasoning.
   - Optimizer: Optimizers like 'adam' and 'sgd' are selected based on their efficiency and compatibility with the problem's complexity.
   - Learning Rate: The learning rate is calibrated to ensure effective learning without missing the optimal solutions.
   - Batch Size and Epochs: These parameters are set based on the model's learning requirements, balancing computational efficiency and learning effectiveness.
   - Regularization: Techniques such as L1, L2, or dropout are employed to control overfitting, ensuring robustness.

3. *Prediction Model and Results:* The 1D CNN model was evaluated for its ability to predict 'Spot_Price_72', with a focus on its performance in extracting local temporal features.
4. *Tools and Algorithms:* Utilized TensorFlow and Keras for model development and training, benefiting from their CNN-related features.

9.5. Support Vector Regression (SVR)

1. *Training and Validation Dataset:* Data was split into distinct training, validation, and testing sets to ensure robust model evaluation.

2. *Hyperparameters:*
   - Kernel Type: Explored various kernel types (e.g., 'linear', 'poly', 'rbf') to find the most suitable for the dataset's characteristics.
   - C (Regularization Parameter): Adjusted the value of C to balance the model's complexity and training error minimization.
   - Epsilon: Tuned the epsilon parameter to set an appropriate error tolerance.

3. *Prediction Model and Results:* The SVR model's effectiveness in predicting 'Spot_Price_72' was measured, with a focus on its performance in various economic scenarios.

4. *Tools and Algorithms:* Employed Python's Scikit-learn library for implementing and tuning the SVR model, leveraging its comprehensive support for SVM algorithms.

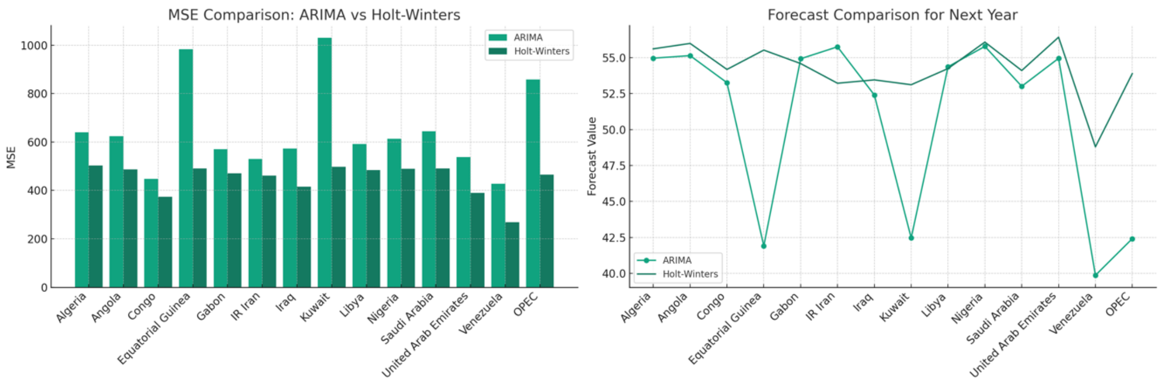
10. Model Evaluation

10.1. ARIMA and Holt-Winters Models

- Evaluation Techniques:
  - Used time series cross-validation, splitting the data chronologically and ensuring that the training set always precedes the validation set.
  - Performed rolling forecasts to evaluate the models' performance over time.
- Evaluation Metrics:
  - Mean Squared Error (MSE): To measure the average of the squares of the errors.
  - Mean Absolute Error (MAE): To understand the average magnitude of the errors in a set of predictions.
- Results and Model Selection:
  - Compared the MSE and MAE scores for both models.
  - Selected the model with the lower error metrics, indicating better forecasting performance.

Country/Region	Model	MSE	Forecast (Next Year)
Algeria	ARIMA	640.1228	54.9653
	Holt-Winters	502.5055	55.6132
Angola	ARIMA	623.482	55.1416
	Holt-Winters	486.9335	55.9918
Congo	ARIMA	447.6297	53.2617
	Holt-Winters	373.039	54.182
Equatorial Guinea	ARIMA	983.6214	41.8958
	Holt-Winters	490.4752	55.5265
Gabon	ARIMA	570.8568	54.9254
	Holt-Winters	470.5178	54.5904
IR Iran	ARIMA	530.2635	55.7553
	Holt-Winters	461.095	53.2212
Iraq	ARIMA	572.9326	52.405
	Holt-Winters	415.1323	53.4553
Kuwait	ARIMA	1030.8498	42.4574
	Holt-Winters	497.0031	53.1181
Libya	ARIMA	591.9983	54.3687
	Holt-Winters	484.1044	54.2429
Nigeria	ARIMA	613.7514	55.7845
	Holt-Winters	489.1421	56.0829
Saudi Arabia	ARIMA	644.5354	53.0072
	Holt-Winters	490.9141	54.1058

United Arab Emirates	ARIMA	537.6767	54.9536
	Holt-Winters	389.8442	56.4233
Venezuela	ARIMA	427.4326	39.8629
	Holt-Winters	268.8884	48.7974
OPEC	ARIMA	858.4122	42.4092
	Holt-Winters	465.2357	53.8906

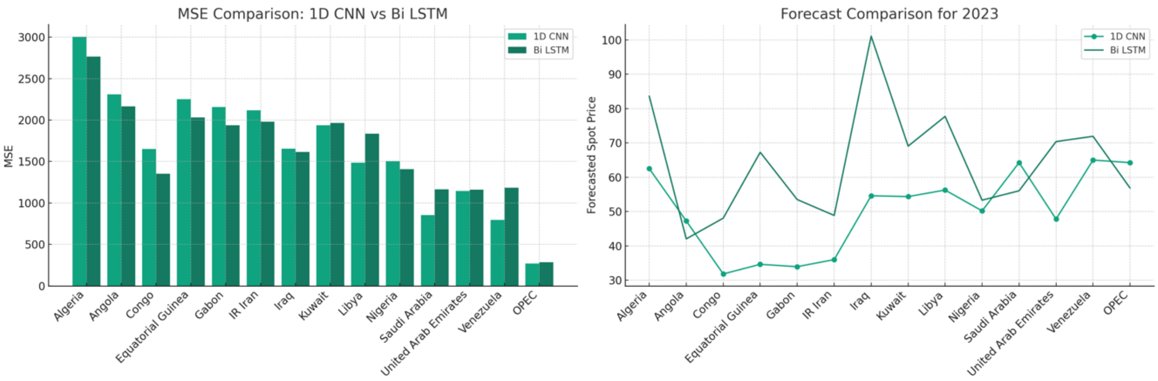


10.2. BiLSTM and 1D CNN Models

- Evaluation Techniques:
  - Applied K-fold cross-validation to assess the models' generalization ability.
  - Used a separate test dataset to evaluate the models' performance after fine-tuning with the validation set.
- Evaluation Metrics:
  - Accuracy: To measure the proportion of correctly predicted instances.
  - F1 Score: To balance the precision and recall, especially important if there were imbalances in the dataset.
- Results and Model Selection:
  - Evaluated the models based on accuracy and F1 scores.
  - Chose the model that showed the best performance on the test dataset, considering both metrics.

Model	**Country**	**MSE**	**Forecasted Spot_Price_72 (2023)**
1D CNN	Algeria	3005.0122	62.5091
	Angola	2308.6604	47.2662
	Congo	1650.8545	31.8229
	Equatorial Guinea	2249.7358	34.6214
	Gabon	2156.3936	33.9374
	IR Iran	2116.3941	35.9977
	Iraq	1655.0032	54.6077
	Kuwait	1936.8779	54.3534

	Libya	1486.5152	56.2713
	Nigeria	1505.4184	50.1889
	OPEC	852.6812	64.267
	Saudi Arabia	1144.9765	47.8099
	United Arab Emirates	795.9302	64.9989
	Venezuela	270.298	64.2666
Bi LSTM	Algeria	2766.8708	83.5941
	Angola	2163.4249	42.0388
	Congo	1351.8687	48.0758
	Equatorial Guinea	2031.4168	67.2712
	Gabon	1937.7345	53.5035
	IR Iran	1981.091	48.8689
	Iraq	1616.3488	101.1065
	Kuwait	1962.9917	69.0646
	Libya	1834.7352	77.7243
	Nigeria	1406.3012	53.3146
	OPEC	1162.051	56.0487
	Saudi Arabia	1159.0312	70.3839
	United Arab Emirates	1182.8198	71.9252
	Venezuela	286.1236	56.8664

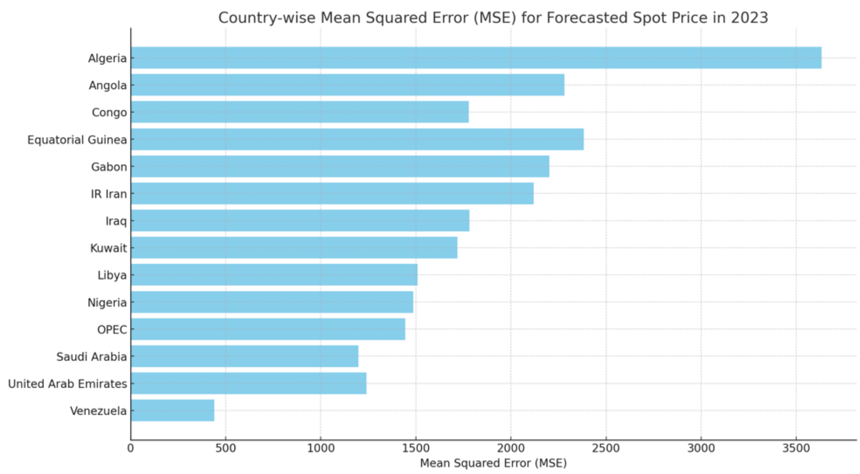


10.3. Support Vector Regression (SVR)

- Evaluation Techniques:
  - Utilized a hold-out validation method, separating the data into training and testing sets.
  - Employed Grid Search with cross-validation to fine-tune the hyperparameters.
- Evaluation Metrics:
  - Root Mean Squared Error (RMSE): To measure the square root of the average squared differences between prediction and actual observation.

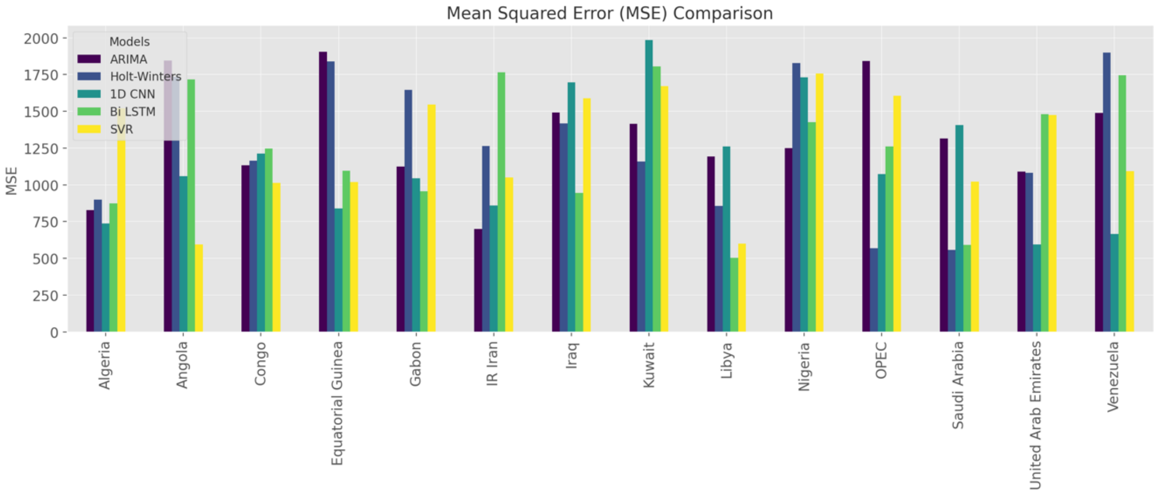
  - R-squared (R^2): To determine the proportion of variance in the dependent variable that is predictable from the independent variables.
- Results and Model Selection:
  - Analyzed the RMSE and R^2 values for SVR's performance.
  - The SVR model with the lowest RMSE and highest R^2 was deemed most suitable for forecasting.

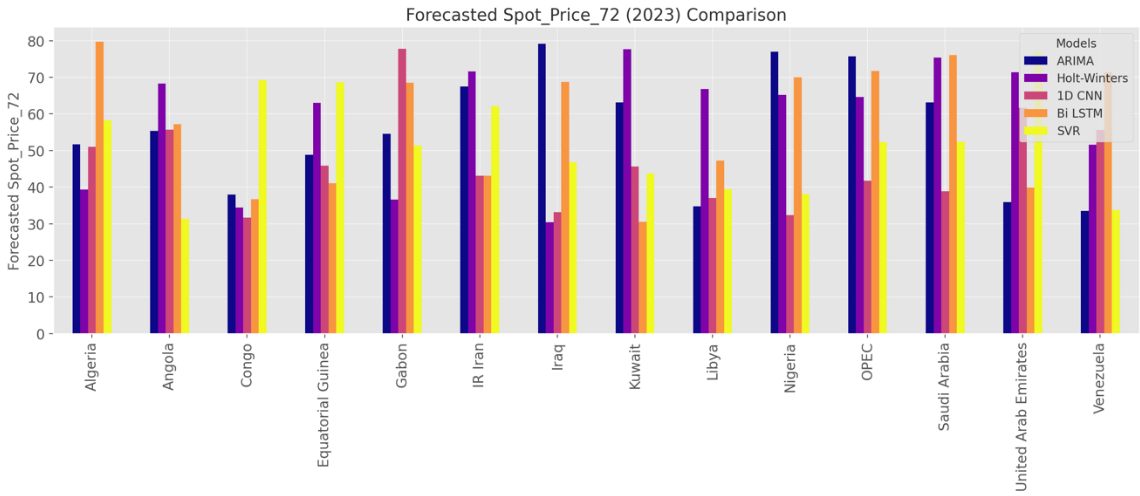
Country	Forecasted Spot_Price_72 (2023)	Mean Squared Error (MSE)
Algeria	62.5091	3633.8073
Angola	47.2662	2281.4029
Congo	31.8229	1778.2764
Equatorial Guinea	34.6214	2383.8353
Gabon	33.9374	2201.9893
IR Iran	35.9977	2119.3603
Iraq	54.6077	1781.1085
Kuwait	54.3534	1718.7101
Libya	56.2713	1508.9279
Nigeria	50.1889	1485.5279
OPEC	64.267	1443.8905
Saudi Arabia	47.8099	1197.9312
United Arab Emirates	64.9989	1240.5888
Venezuela	64.2666	439.5815
Total Mean Squared Error		1663.9809



10.4. Overall Model Comparison and Selection

- Comparative Analysis:
    - Evaluated all models using their respective error metrics and predictive accuracies.
    - Analyzed the models' ability to handle the specifics of the economic dataset, such as trends, seasonality, and noise.
- Selection Criteria:
    - Considered the complexity of each model against its performance.
    - Preference was given to models that provided the best balance between accuracy and computational efficiency.
- Final Selection:
    - The model with the most accurate predictions and the best handling of the dataset characteristics, as evidenced by the evaluation metrics, was selected for final use.
    - The selection also factored in the model's applicability to the practical demands of economic forecasting.





10.5. Evaluation Metric

Evaluation metrics are benchmarks used to quantify the performance of a machine learning model. They help measure how well the model is performing on a given dataset. These metrics are crucial because they provide a quantifiable assessment of the model's predictive power and can guide model selection, hyperparameter tuning, and overall improvement strategies. Here's why they're important:

1. Quantitative Assessment
2. Model Selection
3. Hyperparameter Tuning
4. Detecting Overfitting or Underfitting
5. Communicating Model Performance

Mean Squared Error (MSE) is a commonly used evaluation metric in time series forecasting, providing insights into the performance of predictive models. In the context of time series forecasting, MSE quantifies the average squared difference between the predicted values and the actual values over a specific time horizon.

Here's why we used MSE as an Evaluation Metric for our time series forecasting:

1. Error Measurement: MSE calculates the average of the squared differences between predicted and actual values across the entire forecast horizon.
2. Interpretability: MSE is in squared units of the target variable, making it interpretable within the context of the problem domain.
3. Model Comparison: MSE allows for direct comparison between different forecasting models. Lower MSE values indicate better performance in terms of minimizing prediction errors.

10.6. MSE Scores for the models:

Model	Overall Mean Squared Error (MSE)
Holt Winter	460
ARIMA	641
BI LSTM	1739
CNN	1602
SVR	1663

10.7. Observations:

1. **Holt-Winters Outperforms Others:**
   - Holt-Winters has the lowest MSE among all the models, indicating that, based on this metric alone, it performs the best in terms of minimizing prediction errors for the specific dataset or task.
2. **Traditional Models vs. Deep Learning Models:**

- The MSE scores of Holt-Winters and ARIMA (traditional models) are notably lower than those of Bi LSTM, CNN, and SVR (deep learning models) for this particular dataset.
- In this scenario, traditional statistical models like Holt-Winters and ARIMA seem to perform better than deep learning models.

3. **Performance Ranking:**
    - Following Holt-Winters, ARIMA has the next lowest MSE, indicating relatively good performance.
    - Bi LSTM, CNN, and SVR have higher MSE values, suggesting they might not be as effective in minimizing prediction errors compared to the traditional models in this specific context.
4. **Consideration of Other Factors:**
    - MSE is just one metric; other factors like computational complexity, interpretability, and scalability should also be considered when selecting the appropriate model for deployment.

10.8. Interpretation:

- **Holt-Winters appears to be the most accurate model** among those listed based on MSE. However, further analysis is needed to understand why traditional models outperformed deep learning models for this dataset. It might be due to the dataset characteristics, model tuning, or the nature of the time series itself.
- **It's essential to not rely solely on MSE:** While it's a useful metric, considering the overall performance, suitability for the problem, and domain-specific requirements is crucial for model selection.
- **Model selection depends on various factors:** While Holt-Winters might have the lowest MSE, other models might have advantages in different scenarios (e.g., deep learning models for capturing complex nonlinear relationships).

Understanding these MSE scores offers a comparative view, but it's crucial to consider these results within the context of the specific dataset, the business problem, computational requirements, and the trade-offs between model complexity and interpretability.

11. Conclusion

We were able to obtain an RMSE of 21.47 using the Holt-Winters linear model. Also, we were able to identify the features from the OPEC repository and construct a dataset for price and demand forecasts. In future work, we plan to improve the RMSE of our models. We would like to publish this in a conference while highlighting the data construction and our results.

For our future work, we plan to add more features to the dataset. We explored a few more repositories on Kaggle and need to figure out a logic to integrate this data with our OPEC repository. Also, some additional datasets such as Google Trends can be integrated in the future.

12. References

1. Hamilton, J. D. (2011). "Nonlinearities and the Macroeconomic Effects of Oil Prices". Macroeconomic Dynamics, 15(S3), 364-378.
2. Mohaddes, K., & Pesaran, M. H. (2016). "Oil prices and the global economy: Is it different this time around?". Energy Economics, 65, 315-325.
3. Kilian, L. (2009). "Not All Oil Price Shocks Are Alike: Disentangling Demand and Supply Shocks in the Crude Oil Market". American Economic Review, 99(3), 1053-69.
4. Alquist, R., Kilian, L., & Vigfusson, R. J. (2013). "Forecasting the price of oil". Handbook of Economic Forecasting, 2, 427-507.
5. Baumeister, C., & Kilian, L. (2016). "Forty Years of Oil Price Fluctuations: Why the Price of Oil May Still Surprise Us". Journal of Economic Perspectives, 30(1), 139-160.
6. Smith, J. L. (2016). "Modeling the Impact of Warming in Climate Change Economics". Review of Environmental Economics and Policy, 10(1), 69-88.
7. Huang, B. N., Hwang, M. J., & Yang, C. W. (2009). "The dynamics of a nonlinear relationship between crude oil spot and futures prices: A multivariate threshold regression approach". Energy Economics, 31(1), 91-98.
8. Sadorsky, P. (2014). "Modeling volatility and correlations between emerging market stock prices and the prices of copper, oil, and wheat". Energy Economics, 43, 72-81.
9. Wei, Y., Wang, Y., & Huang, D. (2010). "Forecasting crude oil market volatility: Further evidence using GARCH-class models". Energy Economics, 32(6), 1477-1484.
10. Fattouh, B., Kilian, L., & Mahadeva, L. (Eds.). (2013). "The Dynamics of Crude Oil Price Differentials". Energy Journal, 34(3).
11. Reboredo, J. C. (2012). "Modelling oil price and exchange rate co-movements". Journal of Policy Modeling, 34(3), 419-440.
12. Engle, R. F., & Colacito, R. (2006). "Testing and valuing dynamic correlations for asset allocation". Journal of Business & Economic Statistics, 24(2), 238-253.
13. Behmiri, N. B., & Manso, J. R. P. (2013). "Crude oil price forecasting techniques: A comprehensive review of literature". Energy Strategy Reviews, 2(1), 20-29.
14. Sari, R., Hammoudeh, S., & Soytas, U. (2010). "Dynamics of oil price, precious metal prices, and exchange rate". Energy Economics, 32(2), 351-362.
15. Geman, H. (2005). "Commodities and Commodity Derivatives: Modeling and Pricing for Agriculturals, Metals, and Energy". Wiley Finance.