Final Report: Time Series Forecasting - Store Sales

Nayan Arora
ID-u3249907
University of Canberra
ACT, AUSTRALIA

Abstract

The objective of this project is to deploy a prediction model based on the ideologies of time-series forecasting. The model built will learn from historical data to make knowledgeable predictions for the future. The dataset [9] has multiple files with various attributes that will be used to develop different ML models. An extensive literature review is conducted to infer and understand ML algorithms that are known to perform well for time series forecasting tasks. Then different algorithms are used to develop a forecasting model, after the data is explored and pre-processed. The predictions produced by the ML models are then evaluated using the Root Mean Squared Error. A reference implementation was used to infer, understand, and further build to expand upon the current understanding of the problem. The reference implementation uses a straightforward Time Series Generator method in Keras to build a LSTM regression model. The inferences from this solution are used to further build the solutions, evaluations, and important learnings. In summary, this report provides an in-depth analysis of Time Series forecasting using Decision Trees and Neural Networks. It also covers the steps used to evaluate and implement different methods to build a solution. Future work will be done to produce a real-time analysis for evaluating and predicting store sales.

1. Introduction

Time series forecasting is an artificially intelligent tool or technique that has been used in the field of data science in various industries including astronomy, supply chain management, inventory planning, weather forecasting and more. The ideology behind time series forecasting is to use data collected over a period of time to make knowledgeable predictions of the future events [3]. In the past many different Machine Learning (ML) methods like Regression, Neural Networks, Support Vector Machines, Random Forests and XGBoost had been implemented to analyze historical data and make future predictions based on patterns and strategic information learned from the data. Other commonly used time series models include ARIMA (Auto Regressive Integrated Moving Average), Prophet and Exponential Smoothing. Most of these machine learning models fall under the umbrella category of Regression models. Regression models provide us with a function that describes the relationship between one or more independent variable and a dependent or target variable [4]. The overall technique used is Time Series Forecasting is to find patterns and sequences in the data over different periods of time. The main aim is to analyze the trends in the past and then make predictions for the future events assuming that they will be similar to historical events. It is understood that the future events may be completely different form the ones in the past and may be influenced by factors like natural disasters, pandemics, epidemics, and other events that may influence the society. But it is important to understand that some of these trends do repeat in the future and thus we research and build prediction models. Some good examples for repetitive time trends can be a higher number of people shopping closer to the weekends, a high than normal number of people traveling closer to the vacation days, etc. The Time Series ML model can recognize these patterns and learn such trends that may influence a business along with information on a positive or negative impact that a business may have experienced. Such trends are then used to make knowledgeable predictions that could help set up a bases for what to expect. This process is used exactly as explained to make weather predictions to inform and warn the public for a possible Tsunami, cyclone, or a really hot day. The area of Time Series Forecasting is still growing, and research is being done to produce methods that analyze and make informed decisions on what how different time dependent variables should be used to make predictions using different ML models in real time to get the best possible results based on an evaluation metrics (root mean squared error or other).

More specific to the scope of this project, we will maintain a core focus on developing a machine learning time series model. This model will help make future predictions for a large retailer, the Corporación Favorita [1]. Some questions that this project will be able to answer are as follows:

- What did the sales look like for the store in the past years and if there are patterns in the data that influences those sales numbers? Can this historical data be used to make future predictions?
- Summary of best-selling products for Favorita stores over the years.
- What product family should be given priority in the future based on historical data?
- What influence does the fuel prices have on product sales and how future fuel prices could affect sales?
- What influence does different holidays in the calendar year have over store sales and a future prediction for the same.
- How do the number of transactions vary during different days of the week and a prediction for similar future transactions.

The dataset [9] that will be used for this project can be found using this link.
The training dataset has the following attributes: Product ID, Date, Store ID, Product Family, Total Sales, On Promotion. The target attribute will be Predicted Sales. The dataset consists of over 60,000 different product ID's and 300,000 sales records between the years 2013 and 2017. All product information is further explained in different csv files, namely, Store locations, Oil Prices, Holiday Events, and number of Transactions for each store on a given day. All these csv files have further data for different store on different days. All attributes

will initially be merged together to form an extensive dataset for developing the Time series using the Time Series Generator in Keras. Below is an image summarizing the data split. Note that the test csv is missing the sales column, that is why there are a total of 5 attributes instead of 6. We will make predictions after training our model using the training set.

```
Training Data Shape:  (3000888, 6)
Testing Data Shape (28512, 5)
```

	id	date	store_nbr	family	sales	onpromotion
0	0	2013-01-01	1	AUTOMOTIVE	0.0	0
1	1	2013-01-01	1	BABY CARE	0.0	0
2	2	2013-01-01	1	BEAUTY	0.0	0
3	3	2013-01-01	1	BEVERAGES	0.0	0
4	4	2013-01-01	1	BOOKS	0.0	0

Figure 1: Data Description

In the initial evaluation of the data, the shape of all data is summarized along with a description for any missing values, which is found to be none for all the given csv's. Below is a summary of the training dataset. It is important to first analyse and ensure that the data we are using for training our model is not skewed or missing any values, hence further exploratory data analyses is also done past this step which will be covered later in this report.

```
-------------------------------Attributes----------------------------------
Index(['id', 'date', 'store_nbr', 'family', 'sales', 'onpromotion'], dtype='object')
-------------------------------Data Types-------------------------------
id                int64
date             object
store_nbr         int64
family           object
sales           float64
onpromotion       int64
dtype: object
----------------------------Missing Values------------------------------
id             0
date           0
store_nbr      0
family         0
sales          0
onpromotion    0
dtype: int64
----------------------------NULL values---------------------------------
id             0
date           0
store_nbr      0
family         0
sales          0
onpromotion    0
dtype: int64
--------------------------Shape Of Data--------------------------------
(3000888, 6)
```

Figure 2: Data Summary

Note that the other csv files are summarized in the code file attached to this report.

Motivation

The overall aim of this project is to study different regression models and perform a result-based evaluation of the theory in time series forecasting. By implementing a reliable prediction model for Favorita stores across Ecuador, we can generate immense value for the corporation. It will help the corporation make important decisions based on a data driven approach which could help gain completive advantage against other stores in the market. By performing a trend analysis an understanding can be created around the effect of oil prices and holiday events on overall store sales. We can further create an inference to predict store/product demands. This will not only help the corporation implement strategically optimized product pricing but also increase profit by managing inventory and scheduling labor per forecasted needs.

Furthermore, by way of this project we will be exploring different ML algorithms to evaluate our results, which will help create an in depth understanding of Time Series Models in a real-world setting. Different applications of such models in the real world will then be understood. This will then help expand and help brainstorm other applications of Time Series Forecasting in my future work. By understanding the depth of how these models operate and how different attributes may influence the results is the overall learning goal that I aim to achieve. Some of the regression methods that this report will cover are as follows:

- <u>XGBBoost</u> (Decision Tree): Evaluation based on minimizing the loss function through gradient boosting while iteratively improving predictions [5]. XGB Regressor maintains a core consideration of interdependences between variables.
- <u>CNN</u> (Convolutional Neural Network): CNNs can efficiently capture short-term temporal dependencies and patterns in the data [17]. The neural network approach has been widely implemented for various classification tasks but It will be interesting to implement and understand its working for a regression task.
- <u>LSTM</u> (Deep Neural Network): Evaluation ideology is based on Recurrent Neural Networks (RNN's), which are used to effectively capture long-term dependencies and complex temporal patterns in a dataset [2].
- <u>CNN-LSTM</u>: The benefit of using this model is that it can read long sequences of data as blocks using the CNN model, then all sequences can be pieced together using the LSTM model [16]. This approach should thus capture the trends and seasonality in the data better while also being able to learn any short term temporal dependencies.

There are other options below that were also considered but not implemented in the current solution to meet the scope of this project and its time constraints. But they will be implemented in my future work and updated code script will be posted to in the GitHub repository.

- ARIMA (AutoRegressive Integrated Moving Average): It helps in capturing the standard termporal dependencies that are unique to a time series data [6].

- Random Forest: Evaluation is based on ensemble averaging and aggregating predictions from multiple decision trees to reduce overfitting to produce a more generalized result [7].

The metric used for the evaluation of results: The future predictions/predictions on unseen data will be compared to a target output variable to compute the RMSE or Root Mean Squared Error which is evaluated as below:

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(\,predicted(i) - actual(i)\,)\text{^}2}$$

Evaluations are done using the RMSE instead of MSE or MAE, because RMSE penalizes the underestimation of the actual value more severely than it does for the overestimation and because it is more robust in terms of handling outliers [8]. Also, since the data is obtained from a Kaggle challenge, the structure and evaluation of the solution for this project will be such that it can easily be uploaded to the Kaggle challenge as a solution.

Upon evaluations of the Regression models we build, the best fitting model will be chosen based on the lowest score for RMSE. The lower the RMSE scores the better the model predicts overall sales based on the historical training data.

In summary, this project will help build a strong basis for all my future work that would relate to working with regression tasks. I will not restrict the literature review and learnings through scholarly research to just the scope of this project but will expand it to understand Regressions tasks in detail. Furthermore, I aim to apply and learn how real time change in temporal data could influence the results achieved. This learning should help me provide a useful learning that I will be able to implement to other projects like forecasting stock prices based on real time changes in the market and more.

2. Literature Review

The Gap

During the research conducted, around 42 different scholarly papers and articles were studied to create an understanding of Time Series Forecasting and the different methods that are used to build forecasting models. Only the most important and relevant information is included in this report. In my research, I also narrowed down my understanding to identify the current gap in Time Series Forecasting models used in the retail industry. A two-step approach is described here [10] – First, an original time series is de-trained using a moving windows averaging approach in LSTM. Subsequently, the residual time series are modeled by Non-Linear Auto-Regressive (NAR) models, by using Neuro-Fuzzy and Feed-Forwards Neural Networks. Another approach used [11] here to calculate

Metropolitan Wages in the Los Angeles area uses Kahlman filter Algorithm. This paper [12] further does a contrasting comparison between traditional statistical methods and several machine learning methods for predicting Demand in the retail industry using transactional record of a large grocery retailer.

While there have been significant improvements in the methods used for time series forecasting, it has to its core been related to improvements in the historical data available which is able to capture several historical events. All the review conducted, points to the gap in current forecasting methods that are related to External Uncontrollable factors. For example, local conditions, weather impacts, economic and political stability and other factors like Covid-19 that can lead to false or unreliable future predictions. Few methods that are more current and are still being researched to handle such external factors that influences the time series forecasting models are, Real-Time forecasting and Adaptive Forecasting. The most recent work for the same has been found [13] to implement a novel Online Adaptive Multi Variate Time Series Forecasting (OAMTS). These methods would follow the ideologies of a generative, self-learning, unsupervised AI model which would help decrease the overall dependance on historical data. But implementation and execution of such a model is not in the scope of this project as it requires extensive research and work to create and implement a novel algorithm to create specialized python libraries with custom code. Furthermore, the dataset selected for this problem records the store sales for the years 2013 – 2017 along with a list of external events that may have influenced store sales on a particular day like holiday events, oil prices and historical events are all recorded for various stores across the region. Therefore, the OAMTS will not provide any significant insights in our current problem subset. Once we have done the initial analyses and built a working model that meets all the standards for real-world application then we can work to expand on our current work and include further analysis on real-time spatial temporal data.

Research on Time Series Models

According to [19], the most common mistake in building ML models for Time series forecasting is to load all the data into a model in a black box fashion, with an expectation to get accurate predictions. The most accurate patterns and logic in the data is obtained when considerations are set around seasonality trends and appropriate time periods that are used for predictions. The XGBRegressor uses a number of gradient boosted trees or n estimators to make future predictions. The gradient boosted trees are essentiality combined decision trees to form a strong learning model. It is also important to understand the appropriate values of lookback period that in turn will help in capturing the appropriate trends in the data. For example, predicting the sales of air tickets can use a lookback period of 14 days (a fortnight) or even 30 (monthly). But for a grocery retailer it is important to use a smaller lookback period to capture weekly or daily trends in the sales data. Such seasonality and trends in the data can be captured using a Deterministic Fourier Series. This approach helps us decompose a time series into its constituent periodic components. In this method the data is expressed as a sum of known, fixed sinusoidal functions. This helps to capture the periodic patterns and trends. The deterministic Fourier series is defined as follows:

$$f(t) = a_0 + \sum_{n=1}^{N}[a_n \cos(2\pi n f_0 t) + b_n \sin(2\pi n f_0 t)]$$

Here:

- $f(t)$ is the time series data being analyzed.
- a0 is the average value of the time series.
- a_n and b_n are the coefficients associated with the n-th harmonic component.
- N is the number of harmonics used in the series.
- *f*0 is the fundamental frequency of the series, which represents the inverse of the period of the fundamental cycle in the data.

The coefficients a_n and b_n represent the amplitudes of the sine and cosine components at different frequencies that helps in modeling different periodic patterns and trends.

The research [20], uses a CNN based stock market prediction model that helped brainstorm and implement a similar approach for the sales prediction model in our case. This paper covers all details on how CNNs have been implemented for similar problems in the past because they learn both Linear and Non-Linear relationships in the data without making any assumptions as a model would in the Decision Tree Approach. Also, it does not require an in-depth Feature Analyses as it can automatically select and learn important features and trends in the Time Series. The CNN does not actually view the data as having time steps, instead, it is treated as a sequence over which convolutional read operations can be performed [17]. This does reduce the overall control over hyperparameters but is still able to capture the short-term temporal dependencies and trends in the data. Also, Neural networks are robust to noise in the data, which makes using them an overall good choice.

The article [21], discusses the LSTM algorithm and how it's a widely applied approach for various Time Series Forecasting models. LSTMs are a type of Recurrent Neural Network (RNN) that are explicitly designed to handle Time Series Data by allowing information to persist over multiple time steps. It is different from other neural network approaches in the way it functions as a feedback network and not feedforward neural network [21]. The feedback approach is used to handle the sequence nature of time series data that helps capture historical trends that can be used for making accurate future predictions. The design of LSTMs also helps in capturing long term dependencies in the data that are usually more complex and span over multiple time steps. It can also handle varying sequence lengths more effectively [22]. Overall, it makes a good choice to use LSTMs for our prediction model subject to hyperparameter tuning and ensuring appropriate values are used to achieve accurate and reliable forecast.

This literature review has been updated and expanded as more work was completed over the course of this semester. The Results obtained by CNNs, and LSTMs were 'good enough' as a relatively low RMSE score was achieved but further research concluded that a combination approach may help achieve even more accurate results. An approach that uses both the CNN and LSTM methods combined was found. In this algorithm the CNN part of the algorithm can help read long sequences of data as blocks or periods, then the LSTM par pieces the periodical sequences together to learn trends and seasonality in the data. This algorithm helps us handle the complexity of our data and allows more hyperparameter tuning as in this approach we can explicitly define the number of timesteps per subsequence for each feature in the data. The research [23], uses the CNN-LSTM for stock predictions. The research explains how in this hybrid approach the researchers were able to combine the CNN that extracts effective features from the data, and LSTM that not only finds the interdependence of data in time series but also automatically detects the best mode suitable for relevant data. I aim to use this hybrid approach to achieve similar successful results for predicting store sales. As the sales data should rather less complex than the noisy stock price data, the results we achieve should provide valuable insights into how CNN-LSTM works with very features but the same overall goal.

Further research was done on the traditional models like ARIMA, Prophet and Random Forest Regressor. A couple of which will be implemented in my future work. The article [18], discusses Prophet and how it has been widely implemented to achieve fast and completely automated forecasts that can even be tuned further. Prophet follows a Bayesian Structural model for building the Time Series. Since there are multiple csv files given in the dataset that also includes the effect of holidays on sales, we could build a Prophet model to fit non-linear trends that will help capture the daily, weekly or monthly patterns in the data including seasonality trends and effect of holidays on those patterns. Prophet is robust to any missing data and handles outliers well. It will be interesting to evaluate the results obtained by the Prophet model as well as other models researched for future work.

Review of Ethical and Quality Concerns

Further review conducted accounts for the ethical and quality related concerns in relation to Time Series Forecasts [14]. There are several ethical concerns like bias in forecast which may lead to serious implications. For example, a false prediction could lead to a resource unavailability for a certain store in a certain region, which combined with rural disadvantages may lead to a famine. Furthermore, complex models for multivariate time series forecasting are not very transparent and require a deep dive into the functionalities to correctly understand how predictions are made. The lack of understanding of such complex predictive models may lead to mistrust for stakeholders. In general, it is not easy to predict cases like Covid-19, war or instability in a countries economy or political state, which is always a factor for unreliable results. Hence, a good solution needs to cover all aspects and take into consideration the Ethical and Quality related challenges entailed to such future forecasts.

3. Methodology and Related Work

In development of a solution methodology for this project, a result-evaluation-based approach is used. First, we create a merged dataset with all training and testing attributes as unique numpy arrays. The numpy arrays are then visualized to make interpretations on the pre-processing steps that will be used. Then we perform a data exploration to visualize the data strategically. After that we implement different solution algorithms known to have performed well for a Time series forecasting problem [2], [3], [5], [6], [7]. Then we evaluate the results obtained by these algorithms by calculating the RMSE on the test dataset as explained earlier. Based on the best score obtained on the dataset we choose our final model for future predictions on unseen data.

The approach used for the Favorita stores dataset. Once we load the multiple csv data files provided, we merge all the data using different unique ids for different files given. Once the training and testing dataset is ready, we follow the pre-processing steps. It is found that over 54 stores are present in this dataset, each having 33 types of products sold, this amounts to 1782 target series for forecasting. Some Visualizations produced are attached below:

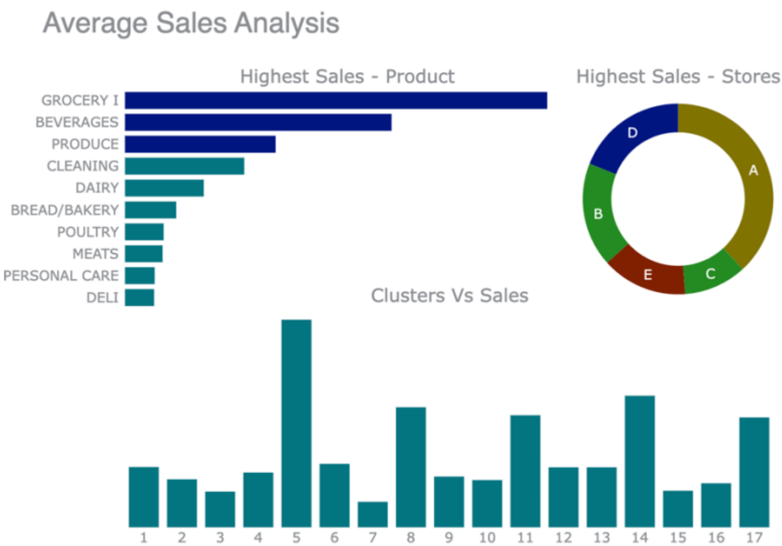

Figure 3: Average Sales Analyses

Various Inferences can be made using the above plot that can help us in further cleaning the data as well as during the training process. For example. The highest sales numbers during the years 2013-2017 was recorded for Groceries, Beverages and Fresh Produce – with store A having the highest sales numbers.

Some of the notable events in the data provided are mentioned as below [9]:

- Wages in the public sector are paid every two weeks on the 15th and the last day of the month which could affect sales on those days.
- An earthquake of magnitude 7.8 struck Ecuador on 16th April 2016. Water and other first need products were donated following the earthquake which greatly affected sales for several weeks.

Further visual plots are made through to gather inferences from the data as below:

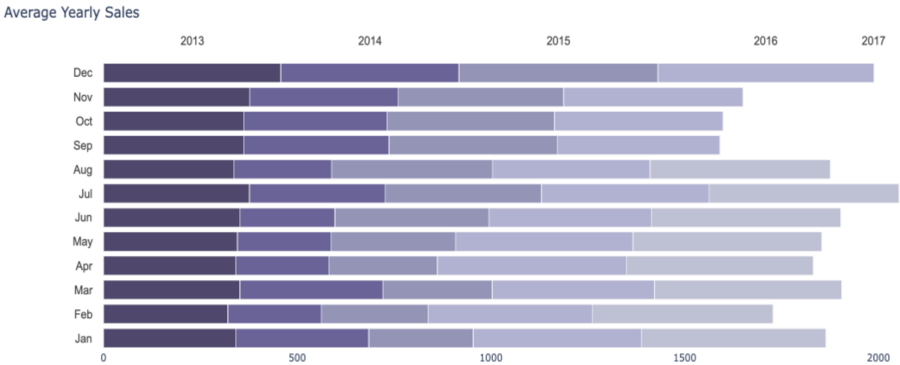

Figure 4: Average Yearly Sales

The above plot shows how sales are going up each year relative to the previous year. Each color is coded to a year and the length of the bar represents the sales that year. So, the major inference here is the visual trend of the time series over years for each month. The months of October, November and December have more sales compared to the rest of the months. This may be because of the festivals and vacation days during those months.

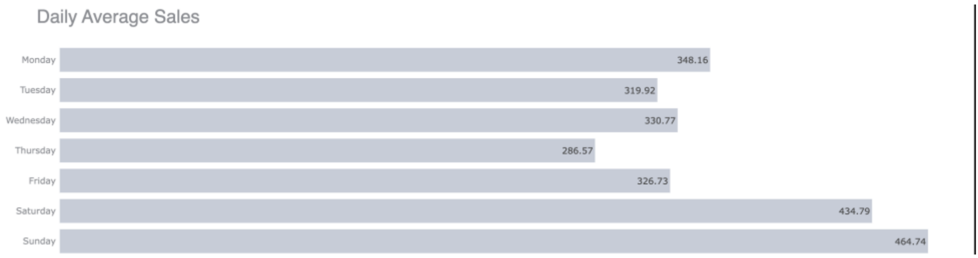

Figure 5: Average Daily Sales

The plot above summarizes the daily average spread of sales over different days of the weak. We can easily infer that there is a trend of high sales numbers over the weekend.

Sunday having the highest sales number followed by Saturday and Monday. It is important that we account for such trends in the periodicity we build for training our ML models.

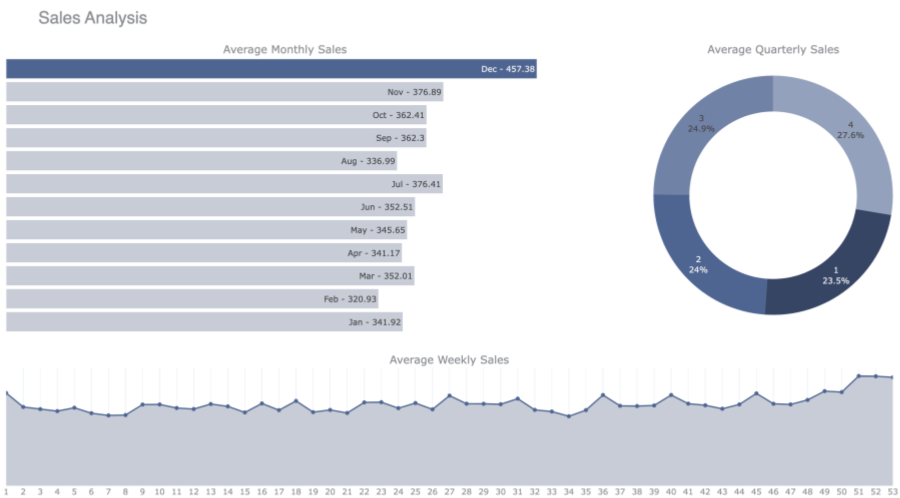

Figure 6: Overall Summary of Sales Records

The plot above is used to summarize the overall sales numbers for each month. The pie chart summarizes the quarterly sales for each store. And at the bottom we have the overall average weekly sales. The important inference here is that the data attribute Onpromotion plays low significance and does not influence the sales numbers as we still got similar results with November and December as the months with highest sales and Store 1 or Store A as the best performing store.

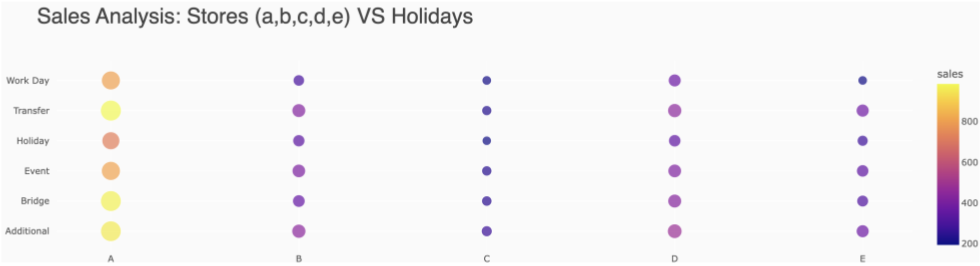

Figure 7: Sales numbers VS Holiday Types for all Stores

Using the above plot, I made the decision to not include the holiday data for training purposes. This is because there are absolutely no dependencies found that may help us conclude that a certain holiday type influenced sales numbers. Since all stores have similar sized circles for all holiday types it summarizes that the overall number of sales were not affected. It just follows the previous inference that on a workday, that is, Monday to Friday the sales numbers are relatively low.

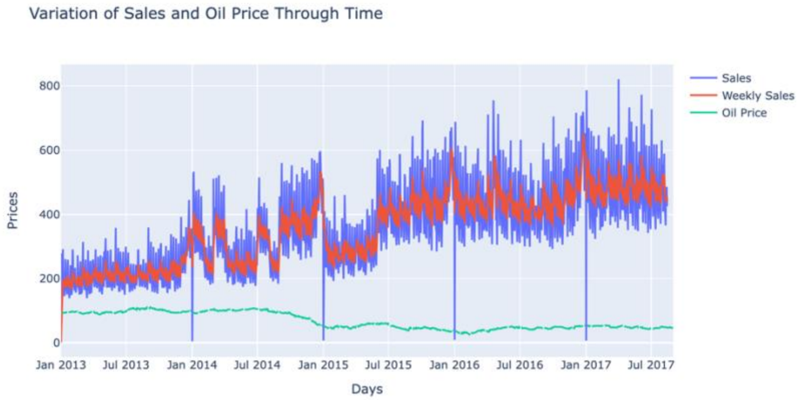

Figure 8: Sales vs Oil Prices

Using the plot above, the summarized inference is that Oil Prices have no effect on the overall sales numbers and will be omitted during the training process. It is necessary to understand that the overall Sales Time Series in the plot above follows a seasonal trend and a sole focus will be maintained to capture that using different algorithms.

The ML models in this report will be hyper tuned to infer and learn the periodic patterns in the sales data over the years and predictions will be made solely on sales data. A mean of Sales for all stores compiled together on a specific date will be used as the Time Series to simplify the modeling process. The sales data used is a smaller subset of the original data to include only the year 2017. This helps significantly reduce the training time and the overall computational power needed for training, testing and debugging purposes.

XGBRegressor

The decision tree approach used by XGBRegressor is the first method that is implemented to evaluate its performance on a complex sales data. We use the deterministic process to create a deterministic Fourier Series that captures the seasonality in the data.

```
fourier = CalendarFourier(freq='M', order=4)
dp = DeterministicProcess(
        index=y.index,
        constant=True,
        order=1,
        seasonal=True,
        additional_terms=[fourier],
        drop=True,)
```

Figure 9: Fourier Series Definition

The Fourier Series defined uses the parameter 'M' to capture Monthly seasonality in the data and the order 4 helps define the periodicity to capture weekly trends. The Constant - True helps capture and remove any bias in the data. Order, 1, helps capture the relationship between each data point with the previous data point in the time series. And lastly, the Seasonal parameter is set to True, to capture daily, weekly and monthly trends in the data. Below is a summary of the parameters and the objective function XGBRegressor uses:

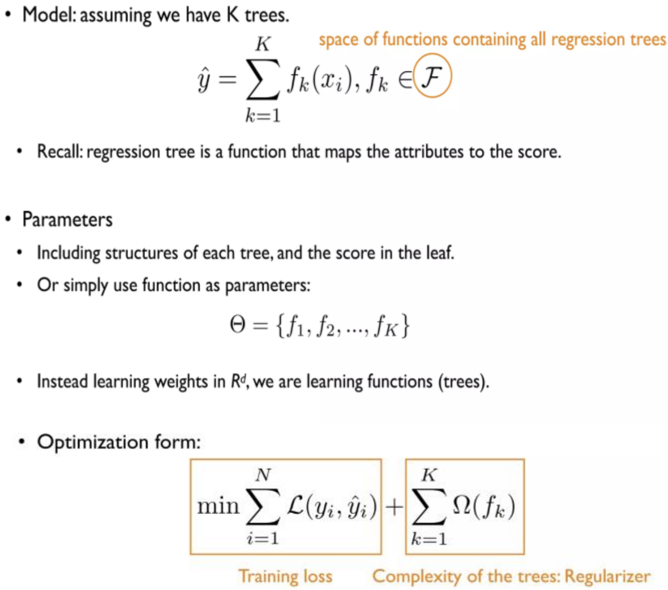

Figure 10: Statistical summary of XGBRegressor

Below are the predictions made by the XGBRegressor for the lag time of 16 days. So, the orange bars represent the next 16 days of sales forecast and the blue bars are the original Sales values for those days. Thus, we get accurate results.

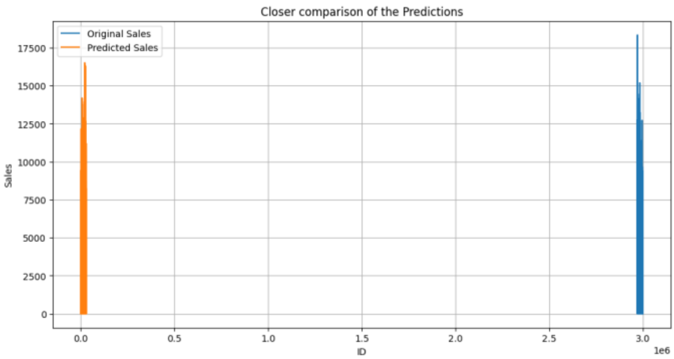

Figure 11: XGBRegressor Results

Convolutional Neural Network (CNN)

A detailed analyses has previously been covered in the literature review section that provides arguments for why it was chosen to implement a CNN model. A different approach using neural networks was the initial idea but because of extensive documentation available a neural network approach is implemented below. The ease to interpret as well as implement along with the hyperparameter tuning provides us equally good results as the XGBRegressor but more much faster.
Below is a summary of the CNN model used:

```
 Layer (type)                Output Shape              Param #
=================================================================
 conv1d (Conv1D)             (None, 30, 128)           384

 max_pooling1d (MaxPooling1  (None, 15, 128)           0
 D)

 flatten (Flatten)           (None, 1920)              0

 dense_1 (Dense)             (None, 128)               245888

 dropout_3 (Dropout)         (None, 128)               0

 dense_2 (Dense)             (None, 128)               16512

 dropout_4 (Dropout)         (None, 128)               0

 dense_3 (Dense)             (None, 34)                4386

 dropout_5 (Dropout)         (None, 34)                0

 dense_4 (Dense)             (None, 1)                 35

...
Total params: 267205 (1.02 MB)
Trainable params: 267205 (1.02 MB)
Non-trainable params: 0 (0.00 Byte)
```

Figure 12: CNN Model Summary

The reshaped data feeded to the model is- Train set shape (10364, 31, 1) and Test set shape (2592, 31, 1). That is, [samples, timestamps, features].

One convolutional hidden layer is used followed by a max pooling layer and fully connected layer was used. Then, we flatten the filter maps before we feed them into the dense layers with Relu activation to finally produce a forecast. Adam optimizer was used, and the loss calculated as the Root Mean Squared Error. Dropout = 0.2 helps avoid overfitting.

Long Short-Term Memory (LSTM)

The Long Short-Term Memory Neural Network can see the data as a sequence. Thus, it is able to learn sequenced time patterns better than a basic CNN that has only

proven to be efficient in automatic feature selection. In the model the LSTM layer used to process input sequence, followed by dense layers with a dropout, 0.2, to avoid overfitting. Same Adam optimizer and Root Mean Squared Error loss is calculated for further evaluation of results.
Below is a summary of the model used:

```
Model: "sequential_2"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 lstm_3 (LSTM)               (None, 256)               264192

 dense_5 (Dense)             (None, 128)               32896

 dropout_6 (Dropout)         (None, 128)               0

 dense_6 (Dense)             (None, 64)                8256

 dense_7 (Dense)             (None, 32)                2080

 dense_8 (Dense)             (None, 1)                 33

=================================================================
Total params: 307457 (1.17 MB)
Trainable params: 307457 (1.17 MB)
Non-trainable params: 0 (0.00 Byte)
_________________________________________________________________
```

Figure 13: LSTM Model Summary

The reshaped data feeded to the model is- Train set shape (10364, 31, 1) and Test set shape (2592, 31, 1). That is, [samples, timestamps, features].

Hybrid CNN-LSTM Model

This approach helps us explicitly define the number of timesteps per subsequence for each feature in the data. Thus, we can customize the parameters better. CNN-LSTM can read long sequences of data as blocks using the CNN model, then all sequences can be pieced together using the LSTM model [24]. The overall structure used in this model is explained using the flowchart below:

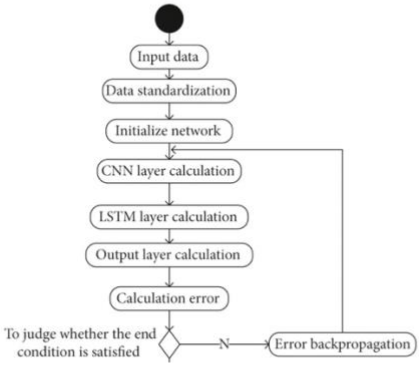

Figure 14: CNN-LSTM Model Structure

Below is a summary of the model used:

Layer (type)	Output Shape	Param #
time_distributed_8 (TimeDistributed)	(None, None, 31, 128)	256
time_distributed_9 (TimeDistributed)	(None, None, 15, 128)	0
time_distributed_10 (TimeDistributed)	(None, None, 1920)	0
time_distributed_11 (TimeDistributed)	(None, None, 1920)	0
lstm_7 (LSTM)	(None, 64)	508160
dense_21 (Dense)	(None, 32)	2080
dense_22 (Dense)	(None, 1)	33

Total params: 510529 (1.95 MB)
Trainable params: 510529 (1.95 MB)
Non-trainable params: 0 (0.00 Byte)

Figure 15: CNN-LSTM Model Summary

We use a Time Distributed Wrapper to wrap the Conv1D, MaxPooling, Flatten and Dropout layers. This wrapper allows to apply a layer to every temporal slice of Input. The time wrapped CNN layers are followed by the LSTM layers. The reshaped data feeded to the CNN-LSTM model is Train set shape (10364, 1, 31, 1) and Test set shape (2592, 1, 31, 1). Here the input format is [samples, subsequences, timestamps, features]. The additional parameter subsequence allows for even further hyperparameter tuning. In this case we used the subsequence as 1 which would create a subsequence at each time step.

For all the models a learning rate of 0.00001 was used and the model was run for 500 epochs with a batch size of 128. The early stopping criterion was implemented with a patience tolerance of 10 to ensure we get the best possible results.

4. Performance Evaluation and Learnings

In this section, first all the evaluated results from the above explained methodology are summarized. Then, further learnings and inferences will be explained.

As stated earlier the evaluation of results are done by calculating the Root Mean Squared Error (RMSE) for all the implemented algorithms. The best model is then used for making a sales forecast.

XGBRegressor RMSE plot below:

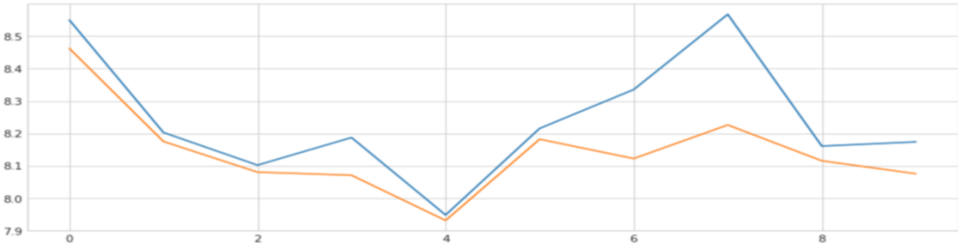

Figure 16: XGBRegressor RMSE

CNN RMSE plot below:

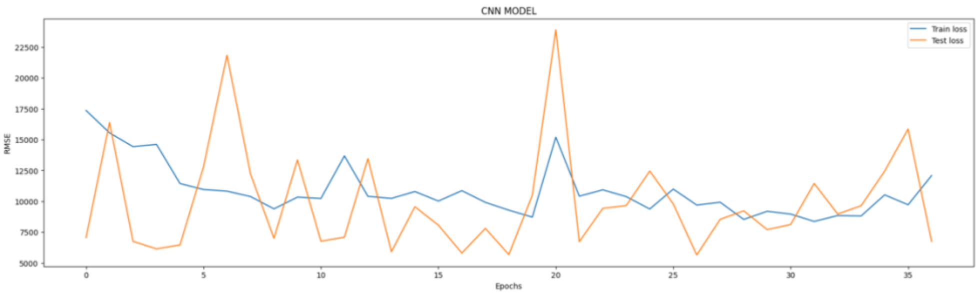

Figure 17: CNN RMSE Plot

<u>LSTM</u> RMSE plot below:

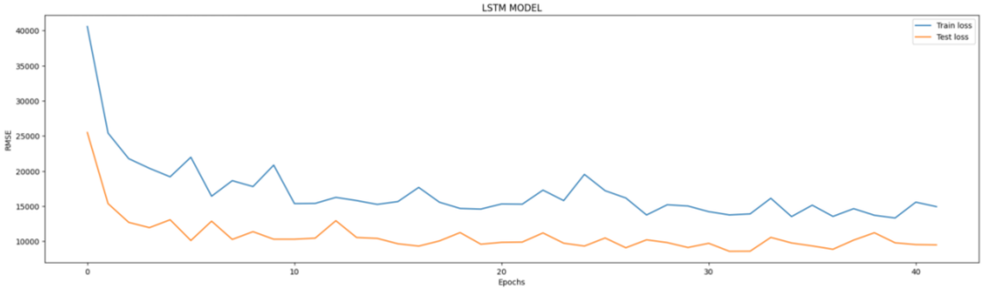

Figure 18: LSTM RMSE Plot

<u>CNN-LSTM</u> RMSE plot below:

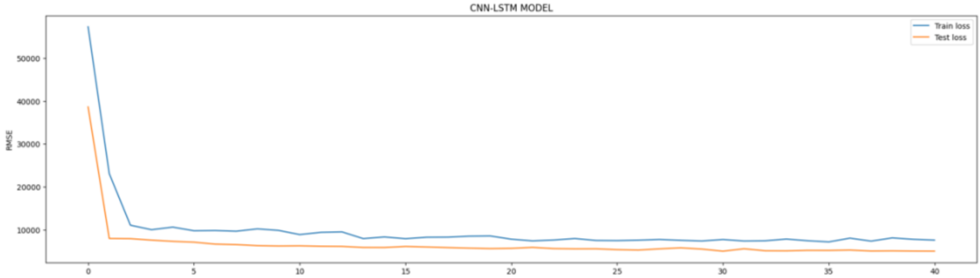

Figure 19: CNN-LSTM RMSE Plot

XGBRegressor evaluation results were attached earlier in the methodology section. Below are the evaluation results on *unseen* data for neural networks:

```
324/324 [==============================] - 1s 1ms/step
81/81 [==============================] - 0s 1ms/step
Train CNN RMSE: 9.330836290203937
Test CNN RMSE: 9.06167509646119
```

Figure 20: CNN Evaluations

```
324/324 [==============================] - 4s 11ms/step
81/81 [==============================] - 1s 11ms/step
Train LSTM RMSE: 10.202909790761792
Test LSTM RMSE: 9.871836227364517
```

Figure 21: LSTM Evaluations

```
324/324 [==============================] - 1s 2ms/step
81/81 [==============================] - 0s 2ms/step
Train CNN-LSTM RMSE: 9.186436292290416
Test CNN-LSTM RMSE: 8.397783580415902
```

Figure 22: CNN-LSTM Evaluations

After we create and run all the explained models and have the results, we can easily infer that CNN-LSTM model performed the best of all the other models. In Figure 19 we can easily note how the Error goes down significantly and is maintained low through all the Epochs. It is important to understand that the RMSE values obtained for all the models on unseen data are close to each other. But we go with a combination of low RMSE on unseen data as well as low RMSE when building the model. The lowest value in the evaluation metric was obtained for CNN-LSTM and thus stands out to be the best fitting model. Below we plot the actual vs predicted results as mentioned during the presentation. This should help visualize how close the CNN-LSTM model is to the actual values.

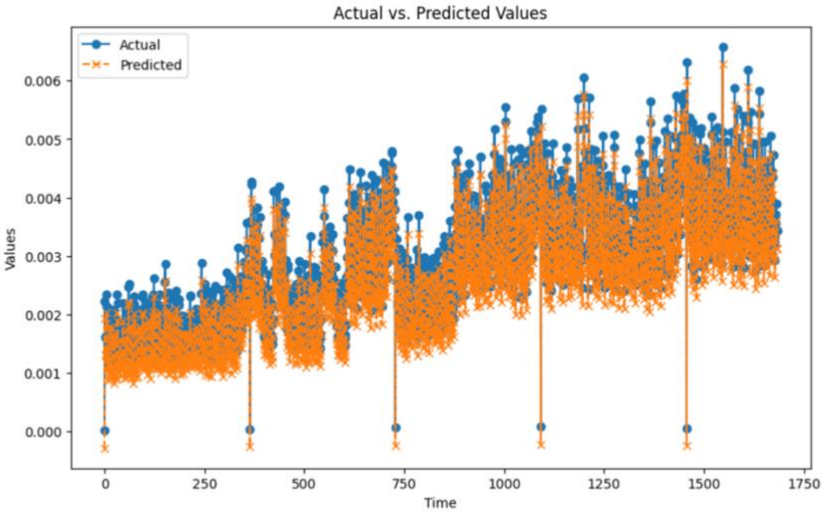

Figure 23: Final Predictions using CNN-LSTM Model

Learnings

This study contributes to the theory of Time Series Forecasting that will help improve retail practices by allowing retailers to make data-driven business decisions. A comparison of multiple regression models on the Faviorita Stores Dataset helped to find the best model for the given problem. One of the most important learning by way of implementing this project was how different neural networks can be combined to form a hybrid model. I personally used a hybrid model for the first time, and it was interesting to learn how a feed-forwards CNN can be combined with a feed-backward LSTM by wrapping the CNN using in a Time Distributed Wrapper. This highlights the further potential of using different hybrid models by combining different neural networks to achieve an improved overall performance.

Data Preprocessing that included handling missing values, scaling, transforming, and feature selection were all important steps that helped me infer how different preprocessing steps are used to achieve certain goals. Another important learning was how Time Series is generated for further evaluations.

There are multiple methods that I implemented in this project, one was to use the Time Series Generator in the Keras Library, another was to use a Deterministic Process to generate a Fourier series and the last one was to manually define a function that converted the data into a time series [24].

Furthermore, the importance of hyperparameter tuning is understood to its depth as well as how different regularization techniques can be used to improve results in a Decision Tree approach (L1, L2). While all the approaches are easily explainable it is also important that they are easily interpretable and understood by an individual who may not have all the knowledge in the field of ML. The CNN-LSTM being a hybrid model is complex and it is necessary to ensure that its complexity can be broken down into smaller pieces of information for better understanding as tried in the execution of this project report.

Acknowledgements

Lastly, I would mention that I really enjoyed the open-endedness of this project. I was able to research and learn various new things while trying to achieve my goal and I would really like to acknowledge Professor Ram for challenging us as well as providing extensive support through the semester. It would not be the same learning experience without the guidance of Professor Ram and our Tutor James.

5. Reference Solutions

The reference solution [15] being used for the execution of this project uses a different dataset as the one provided for Faviorita Stores. The objective followed in this solution is to use deep neural network methodologies to create a regression LSTM model [15] for time series forecasting. This solution has been used as an overall idea of what needs to be achieved and was referred to for learning purposes. Although the solution created for the Favorita Store is unique and only uses this [15] solution as a reference. In the reference solution MinMaxScaler has been used for scaling and LSTM layers used have different hyperparameters. The TimeSeriesGenerator method has been used in a similar manner and the loss function used is Root Mean Squared Error. The overall solution built is straightforward and does not perform and pre-processing or exploration on the dataset.

In the reference solution LSTM Regression model was used because the dataset showed low and negative correlation between features and the targeted variable [15]. In such scenarios a deep neural network that follows the ideologies of RNN's (Recurrent Neural Networks) can help interpret and make better inferences from the data which helps exponentially in the feature extraction process.

The LSTM model used here requires further hyperparameter tuning and algorithmic modifications, which were performed in the solution I built explained earlier in this report. These modifications helps cover all considerations set around the dataset. For example, eliminating outliers and modifying certain aspects to produce more accurate result specifically for Corporación Favorita.

The LSTM model in this solution uses the sequential model from the keras library. The first layer has 50 neurons and 'relu' activation function which is passed back to the previous layer with 50 neurons again which is passes onto the third layer which will finally lead to the output layer with linear activation function that will help produce the target – predicted sales. The model is summarized below:

```
Model: "sequential_3"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 lstm_9 (LSTM)               (None, 30, 50)            10600

 dropout_9 (Dropout)         (None, 30, 50)            0

 lstm_10 (LSTM)              (None, 30, 50)            20200

 dropout_10 (Dropout)        (None, 30, 50)            0

 lstm_11 (LSTM)              (None, 50)                20200

 dropout_11 (Dropout)        (None, 50)                0

 dense_3 (Dense)             (None, 1)                 51

=================================================================
Total params: 51051 (199.42 KB)
Trainable params: 51051 (199.42 KB)
Non-trainable params: 0 (0.00 Byte)
_________________________________________________________________
```

Figure 24: Model Summary

Below is the result produced which is measured using the RMSLE as explained earlier and plotted over 20 epochs. A reducing overall value of RMSLE is a good sign.

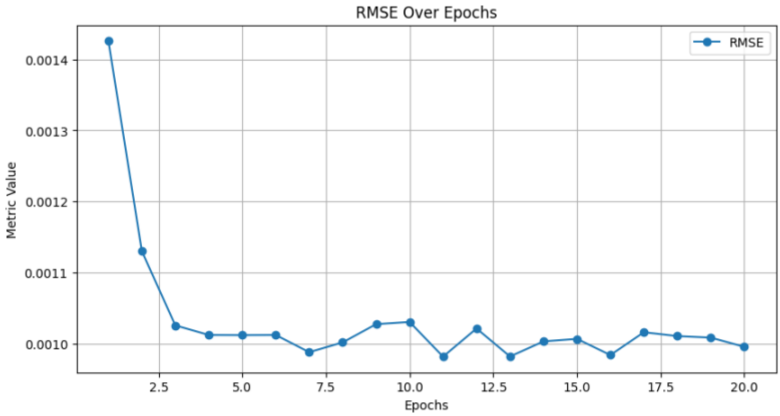

Figure 25: RMSE Over Epochs

Note that in this reference solution the dataset was compiled. That is all the different CSV files were compiled together and then a Time Series Generator Method was used from the Keras library that helps produce a Time Series like data that can be used for training a LSTM model. In this method I have no control over the data and there is a lot of rework every time we try to change the features that we would use for training the model. This approach is efficient but does not allow setting up the parameters based on personal interpretations. For example, in the solution that I built I only used a smaller subset of the original data for the year 2017. Based on the exploratory data analyses I opted out of using certain data like the oil prices, holidays, etc. It is much more convenient to use a self-written method to create the time series that allows more control and modifications.

6. Ethical and Quality Considerations

Several Ethical concerns like bias in forecast may lead to serious implications. A false prediction could lead to a resource unavailability for a certain store in a certain region, which combined with rural disadvantages may lead to a famine. The lack of understanding of complex predictive models may lead to mistrust for stakeholders. These factors can be further examined deeply to ensure that in a real-world application all considerations are set to control such biased outcomes from the predictions.

We also need to ensure that dataset used for making predictions is consistent and of high quality. Inaccurate, Inconsistent and/or incomplete data can result in poor and unreliable forecasts. Thus, quality control and data pre-processing are crucial steps to ensure reliable results.

The model should be easily interpretable and explainable to be able to address any bias, errors, or any other issues that way we can build trust in the model's predictions. Lastly it is important to follow all Data Privacy and Security standards.

7. Future Work

In my Literature Review I found a recent research [13] on an Online Adaptive Multivariate Time Series Forecasting. This model uses a fully automatic framework for both adaptive input spatio-temporal dependent variables and adequate forecasting model selection. The adaptation is performed using 'concept-drift' detection in both spatio-temporal dependencies and model performance over time. This type of model will be able to account for real-time changes in certain features and then select the best fitting model from a pool of options in real time. I aim to expand my work further using my learning from this project and implementing this improved model for making predictions on stock prices.

Also, other algorithms - ARIMA, Random Forest and Prophet will be implemented to further understand how the results differ. These explanations will be updated and included in the Github Repository.

8. References

[1]C. Favorita, "Enermax," Corporación Favorita, 2023. https://www.corporacionfavorita.com/marcas/area-comercial/ (accessed Oct. 07, 2023).

[2]B. Lim and S. Zohren, "Time Series Forecasting With Deep Learning: A Survey," Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences, vol. 379, no. 2194, p. 20200209, Apr. 2021, doi: https://doi.org/10.1098/rsta.2020.0209.

[3]P. Dix, "Time Series Forecasting Methods," InfluxData, 2022. https://www.influxdata.com/time-series-forecasting-methods/#:~:text=Time%20series%20forecasting%20is%20a

[4]IMSL, "What Is a Regression Model?," IMSL by Perforce, Jun. 16, 2021. https://www.imsl.com/blog/what-is-regression-model#:~:text=A%20regression%20model%20provides%20a

[5]A. Vidhya, "Understanding the Math behind the XGBoost Algorithm," Analytics Vidhya, Sep. 06, 2018. https://www.analyticsvidhya.com/blog/2018/09/an-end-to-end-guide-to-understand-the-math-behind-xgboost/

[6]J. Nathaniel, "Introduction to ARIMA for Time Series Forecasting," Medium, Aug. 10, 2021. https://towardsdatascience.com/introduction-to-arima-for-time-series-forecasting-ee0bc285807a

[7]K. Prawtama, "Predicting Store Sales — Random Forest Regression," MLearning.ai, Feb. 21, 2022. https://medium.com/mlearning-ai/predicting-store-sales-random-forest-regression-b77abec64c17

[8]S. Saxena, “RMSE vs RMLSE — What’s the Difference? When Should You use Them?,” Analytics Vidhya, Feb. 13, 2020. https://medium.com/analytics-vidhya/root-mean-square-log-error-rmse-vs-rmlse-935c6cc1802a

[9]Kaggle, “Store Sales - Time Series Forecasting,” kaggle.com, 2023. https://www.kaggle.com/competitions/store-sales-time-series-forecasting/data

[10]V. Nunnari, “Forecasting Monthly Sales Retail Time Series: A Case Study,” ieeexplore.ieee.org, Jul. 27, 2017. https://ieeexplore.ieee.org/abstract/document/8010700 (accessed Oct. 08, 2023).

[11]R. Engle and M. Watson, “A One-Factor Multivariate Time Series Model of Metropolitan Wage Rates,” Journal of the American Statistical Association, vol. 76, no. 376, pp. 774–781, Dec. 1981, doi: https://doi.org/10.1080/01621459.1981.10477720.

[12]J. Wang, G. Q. Liu, and L. Liu, “A Selection of Advanced Technologies for Demand Forecasting in the Retail Industry,” IEEE Xplore, Mar. 01, 2019. https://ieeexplore.ieee.org/document/8713196

[13]A. Saadallah, H. Mykula, and K. Morik, “Online Adaptive Multivariate Time Series Forecasting,” 2022. Accessed: Oct. 08, 2023. [Online]. Available: https://2022.ecmlpkdd.org/wp-content/uploads/2022/09/sub_1180.pdf

[14]C. K. Stenmark, A. L. Antes, X. Wang, J. J. Caughron, C. E. Thiel, and M. D. Mumford, “Strategies in Forecasting Outcomes in Ethical Decision-Making: Identifying and Analyzing the Causes of the Problem,” Ethics & Behavior, vol. 20, no. 2, pp. 110–127, Mar. 2010, doi: https://doi.org/10.1080/10508421003595935.

[15]S. Karra, “LSTM for Time Series predictions,” Medium, Oct. 06, 2020. https://sailajakarra.medium.com/lstm-for-time-series-predictions-cc68cc11ce4f (accessed Oct. 08, 2023).

[16]https://www.facebook.com/jason.brownlee.39, “How to Get Started with Deep Learning for Time Series Forecasting (7-Day Mini-Course),” Machine Learning Mastery, Aug. 05, 2019. https://machinelearningmastery.com/how-to-get-started-with-deep-learning-for-time-series-forecasting-7-day-mini-course/

[17]J. Brownlee, “How to Develop Convolutional Neural Network Models for Time Series Forecasting,” Machine Learning Mastery, Nov. 11, 2018. https://machinelearningmastery.com/how-to-develop-convolutional-neural-network-models-for-time-series-forecasting/

[18]Nadeem, “Introduction to Prophet Algorithm,” CodeX, Dec. 20, 2021. https://medium.com/codex/introduction-to-prophet-algorithm-a59e463a6c72

[19]M. Grogan, “XGBoost For Time Series Forecasting: Don’t Use It Blindly,” Medium, Jun. 26, 2023. https://towardsdatascience.com/xgboost-for-time-series-forecasting-dont-use-it-blindly-9ac24dc5dfa9#:~:text=How%20XGBRegressor%20Forecasts%20Time%20Series (accessed Oct. 29, 2023).

[20]E. Hoseinzade and S. Haratizadeh, “CNNPred: CNN-based stock market prediction using several data sources,” arXiv:1810.08923 [cs, q-fin, stat], Oct. 2018, Available: https://arxiv.org/abs/1810.08923

[21]C. Camille, “Time Series Prediction Using LSTM in Python,” CodeX, Feb. 13, 2023. https://medium.com/codex/time-series-prediction-using-lstm-in-python-19b1187f580f#:~:text=LSTMs%20are%20a%20type%20of

[22]K. Team, “Keras documentation: LSTM layer,” keras.io. https://keras.io/api/layers/recurrent_layers/lstm/

[23]W. Lu, J. Li, Y. Li, A. Sun, and J. Wang, “A CNN-LSTM-Based Model to Forecast Stock Prices,” Complexity, Nov. 24, 2020. https://www.hindawi.com/journals/complexity/2020/6622927/

[24]“Deep Learning for Time Series Forecasting,” kaggle.com. https://www.kaggle.com/code/dimitreoliveira/deep-learning-for-time-series-forecasting (accessed Oct. 29, 2023).