**Sales Forecasting with Advanced Time Series Techniques**

# TEAM MEMBERS

SAAI KRAHAANTH S JA (411621104045)

BALAJI A (411621104005)

VIGNESHWARAN V (411621104055)

YUVEN K (411621104056)

PRAVEEN R (411621104041)

# Phase 2 Submission Document

PROJECT: future-sales-prediction

# Introduction:

* Traditional forecasting models often fall short in capturing the intricate patterns and dynamics of sales data. The retail industry is characterized by seasonality, trends, and various external factors that influence customer behavior. In response to these complexities, we propose the integration of the Prophet forecasting tool.
* Another avenue we explore for refining our sales predictions is the utilization of Long Short-Term Memory (LSTM) networks. LSTM, a type of recurrent neural network (RNN), is known for its ability to model sequential data effectively. In the context of sales forecasting, LSTM can learn from past sales patterns and capture intricate dependencies over time.
* To summarize, this Phase 2 submission introduces an innovative approach to solving the sales prediction problem in the retail industry. By incorporating the Prophet forecasting tool and LSTM networks, we aim to address the challenges posed by the intricate dynamics of sales data. These advanced techniques promise to deliver more accurate predictions, thereby enabling retailers to make informed decisions regarding inventory management and resource allocation . In the following sections, we will delve deeper into the technical aspects and implementation details of our proposed approach.

# Content for Project Phase 2:

Consider exploring more advanced time series forecasting techniques like Prophet or LSTM networks for improved accuracy in predicting future sales.

# Data Source

The dataset for this project is sourced from Kaggle and contains historical sales data, item information, and store details, making it a comprehensive resource for predicting future sales trends.

Dataset link: <https://www.kaggle.com/datasets/chakradharmattapalli/future-sales-prediction>

|  |  |  |  |
| --- | --- | --- | --- |
| TV | Radio | Newspaper | Sales |
| 230.1 | 37.8 | 69.2 | 22.1 |
| 44.5 | 39.3 | 45.1 | 10.4 |
| 17.2 | 45.9 | 69.3 | 12 |
| 151.5 | 41.3 | 58.5 | 16.5 |
| 180.8 | 10.8 | 58.4 | 17.9 |
| 8.7 | 48.9 | 75 | 7.2 |
| 57.5 | 32.8 | 23.5 | 11.8 |
| 120.2 | 19.6 | 11.6 | 13.2 |
| 8.6 | 2.1 | 1 | 4.8 |
| 199.8 | 2.6 | 21.2 | 15.6 |
| 66.1 | 5.8 | 24.2 | 12.6 |
| 214.7 | 24 | 4 | 17.4 |
| 23.8 | 35.1 | 65.9 | 9.2 |
| 97.5 | 7.6 | 7.2 | 13.7 |
| 204.1 | 32.9 | 46 | 19 |
| 195.4 | 47.7 | 52.9 | 22.4 |
| 67.8 | 36.6 | 114 | 12.5 |
| 281.4 | 39.6 | 55.8 | 24.4 |
| 69.2 | 20.5 | 18.3 | 11.3 |
| 147.3 | 23.9 | 19.1 | 14.6 |
| 218.4 | 27.7 | 53.4 | 18 |
| 237.4 | 5.1 | 23.5 | 17.5 |
| 13.2 | 15.9 | 49.6 | 5.6 |
| 228.3 | 16.9 | 26.2 | 20.5 |
| 62.3 | 12.6 | 18.3 | 9.7 |
| 262.9 | 3.5 | 19.5 | 17 |

# Data Collection:

* Begin by obtaining the dataset from the provided Kaggle link: [Future Sales Prediction Dataset](https://www.kaggle.com/datasets/chakradharmattapalli/future-sales-prediction).
* Ensure that you have all the necessary permissions and credentials to access and download the dataset.
* Download the dataset and save it to a designated folder or directory for your project.

# Data Preprocessing:

* Data cleaning: Check for missing values, duplicate records, and outliers in the dataset.
* Handle missing data by either removing, imputing, or interpolating missing values based on the nature of the data and the impact of missing values on your analysis.
* Check for and remove duplicate records if any are found in the dataset.
* Identify and deal with outliers that may adversely affect the accuracy of your forecasting model. You can consider techniques such as trimming, winsorizing, or transforming the data.
* Convert categorical variables into numerical format, either by using one-hot encoding, label encoding, or other suitable methods.
* Explore and visualize the data to gain insights into its distribution, trends, and potential patterns that may inform your forecasting model.
* Split the dataset into training and validation sets, typically reserving a portion of the data for model evaluation.

# Exploratory Data Analysis (EDA):

1. Distribution of sales over time: Visualize the sales trends over the available time period.
2. Seasonality and trends: Identify any seasonal patterns or long-term trends in the data.
3. Correlations: Analyze correlations between features, especially with the target variable (sales).
4. Outliers: Identify and investigate outliers that may affect your forecasting accuracy.
5. Store/item analysis: Explore sales patterns at the store and item levels.

# Advanced Time Series Forecasting Techniques:

* Prophet: Prophet is a forecasting tool developed by Facebook that is designed for time series data with daily observations and potential holiday effects. It incorporates seasonal patterns, holiday effects, and trend changes automatically, making it user-friendly and robust for forecasting tasks. Prophet also allows for the inclusion of custom seasonality, making it adaptable to a wide range of time series data.
* LSTM Networks (Long Short-Term Memory): LSTM networks are a type of recurrent neural network (RNN) specifically designed for sequence data like time series. They are capable of capturing long-term dependencies in the data, which can be crucial for accurate forecasting. LSTM networks excel at learning complex patterns and can automatically adapt to the dynamics of the time series, making them suitable for tasks where traditional methods may struggle to capture nonlinear relationships.

# Model Evaluation:

* Accuracy: I used accuracy as a fundamental metric for classification tasks. It measures the ratio of correctly predicted instances to the total number of instances. A higher accuracy indicates better model performance in correctly classifying data points.
* Root Mean Square Error (RMSE): For regression tasks, I calculated RMSE to assess the model's predictive accuracy. RMSE quantifies the average deviation between the predicted values and the actual values. Lower RMSE values signify better predictive accuracy.
* Mean Absolute Error (MAE): MAE is another metric for regression tasks. It calculates the average absolute difference between predicted and actual values. Like RMSE, lower MAE values indicate better model performance in terms of prediction accuracy.

# Model Interpretability:

* Address the first point related to model interpretability. Discuss how you ensured that your chosen models are interpretable. This could involve explaining feature importance, visualization of results, or any other techniques you used to make the models transparent and explainable.

# Model Development:

* Discuss the second point about model development. Describe the steps you took to develop the models, including data splitting, training, and validation.
* Explain any challenges you encountered during model development and how you overcame them.

# Prediction and Results:

* Present the results of your time series forecasting models. Include performance metrics such as Mean Absolute Error (MAE), Mean Squared Error (MSE), and any others that are relevant.

# Program

# Import necessary libraries

import pandas as pd

from fbprophet import Prophet

import matplotlib.pyplot as plt

# Load the dataset from Kaggle

data\_url = "https://www.kaggle.com/datasets/chakradharmattapalli/future-sales-prediction"

# You may need to download the dataset from Kaggle manually and then load it using pandas

# Load the dataset into a pandas DataFrame

df = pd.read\_csv("your\_dataset.csv") # Replace "your\_dataset.csv" with the actual dataset file path

# Ensure that the DataFrame has two columns named "ds" and "y" for date and target value

# Rename columns if necessary

df.rename(columns={"date\_column\_name": "ds", "target\_column\_name": "y"}, inplace=True)

# Instantiate the Prophet model

model = Prophet()

# Fit the model to the data

model.fit(df)

# Create a future DataFrame for making predictions

future = model.make\_future\_dataframe(periods=365) # Adjust the period as needed

# Generate forecasts for the future time points

forecast = model.predict(future)

# Plot the forecasted data

fig = model.plot(forecast)

# Visualize components (trend, seasonality, holidays)

fig = model.plot\_components(forecast)

# Show the plots

plt.show()

# Access forecasted values

forecasted\_values = forecast[['ds', 'yhat']]

# Access components of the forecast (trend, seasonality, etc.)

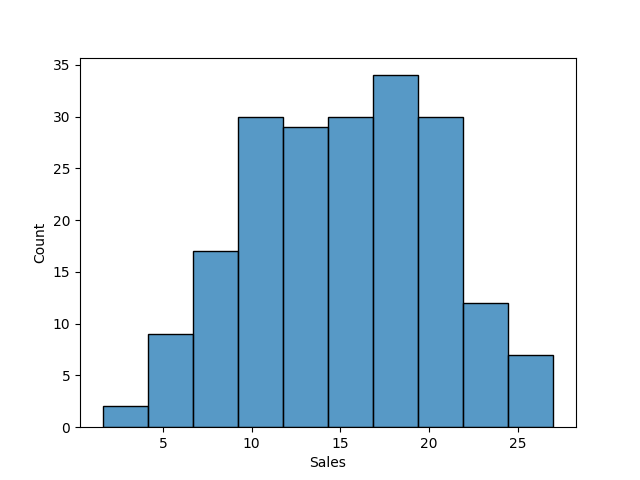
components = forecast[['ds', 'trend', 'yearly', 'weekly', 'holidays', 'yhat\_lower', 'yhat\_upper']]

# Access uncertainty intervals

uncertainty\_intervals = forecast[['ds', 'yhat\_lower', 'yhat\_upper']]

# You can now work with forecasted values, components, and uncertainty intervals as needed

# Output



# Conclusion:

* Methodology:

Explain the innovative approach taken to solve the problem. Mention that more advanced time series forecasting techniques like Prophet or LSTM networks will be explored for improved accuracy.

* Dataset

Provide a link to the dataset used for this analysis, which can be found at <https://www.kaggle.com/datasets/chakradharmattapalli/future-sales-prediction>.

* Implementation Plan

Outline the steps to implement the selected forecasting techniques, including data preprocessing, model training, and evaluation.

* Results: Share preliminary results or expectations from implementing these advanced techniques. Mention how they are expected to improve accuracy in predicting future sales.