MoveInSync Task Report

Problem Statement

1. Design a time series forecasting model to predict values for the next three months based on the provided historical data
2. Output should be filled csv file(test1.csv).
3. Date range for testing - 01-10-2017 to 31-12-2017

Data Overview

The data consists of two datasets: train1.csv and test1.csv. The primary objective is to analyze the trends, seasonality, and other patterns in the price column over time, and to build predictive models. The datasets were preprocessed to ensure uniformity and readiness for time-series modeling.

Data Preprocessing

1. Datetime Conversion:
   The date column was converted to a datetime format (%d-%m-%Y) for both datasets to enable time-series analysis.
2. Sorting and Missing Value Check:
   The train_data was sorted by date.
   Missing values were checked, and no imputation or interpolation was required.

Feature Engineering:

Extracted additional temporal features such as year, month, day, day_of_week, and is_weekend (binary feature to indicate weekends).

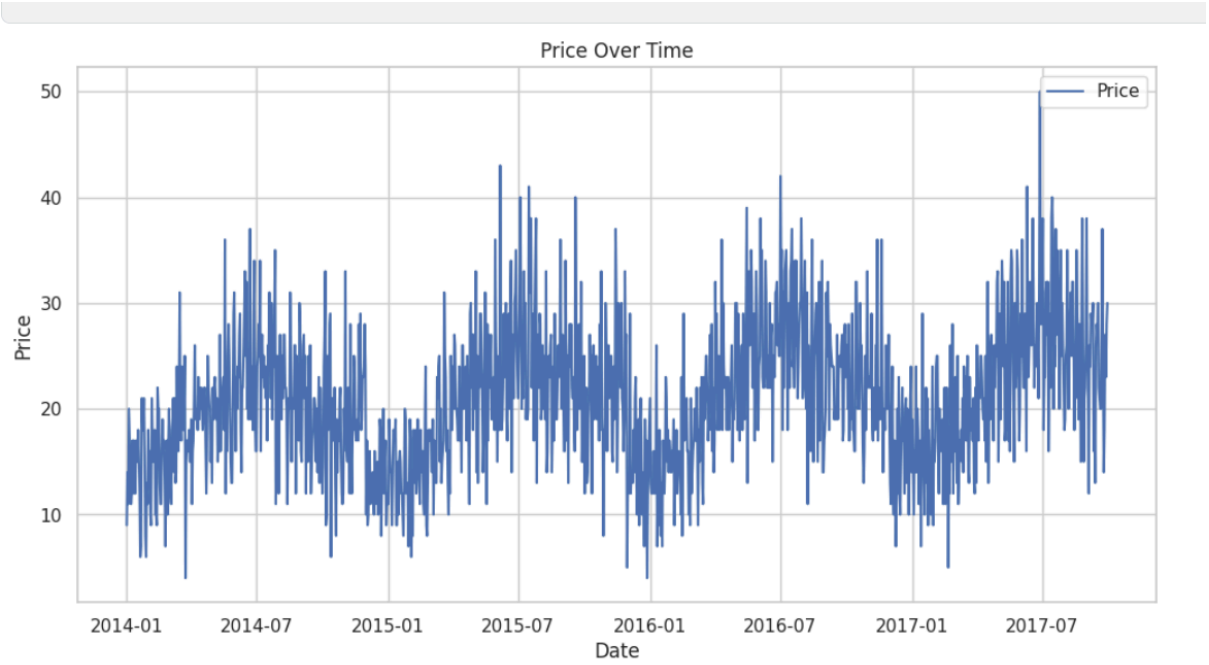
Exploratory Data Analysis

Descriptive Statistics

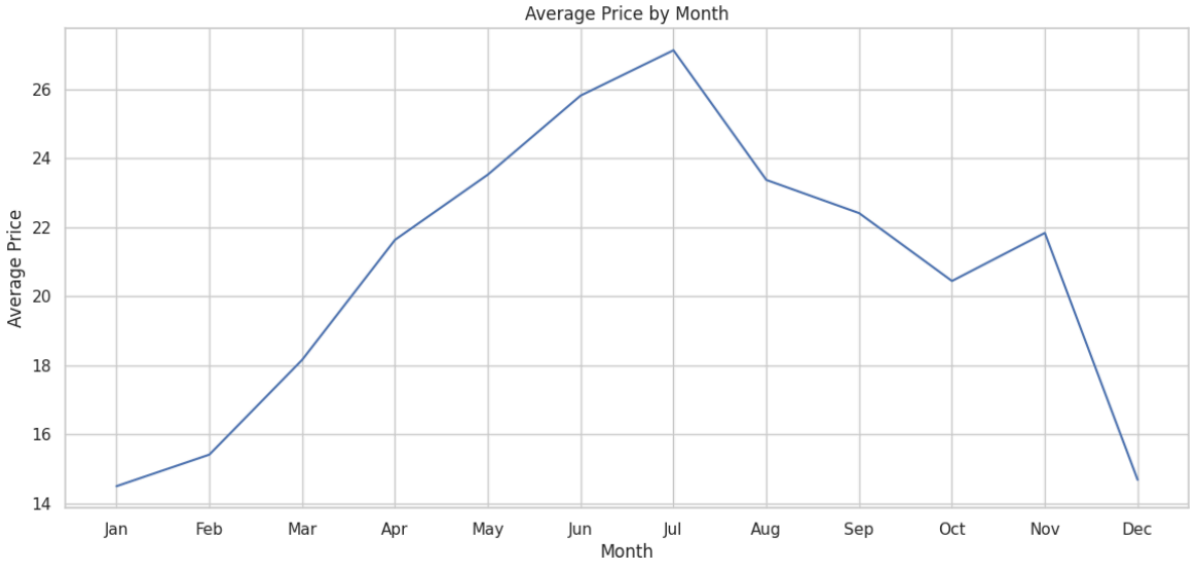
1. Mean:Calculated as the average price.
2. Variance: Measures the price variability.
3. Median:Provides the middle value.

Visualizations

Price Over Time:

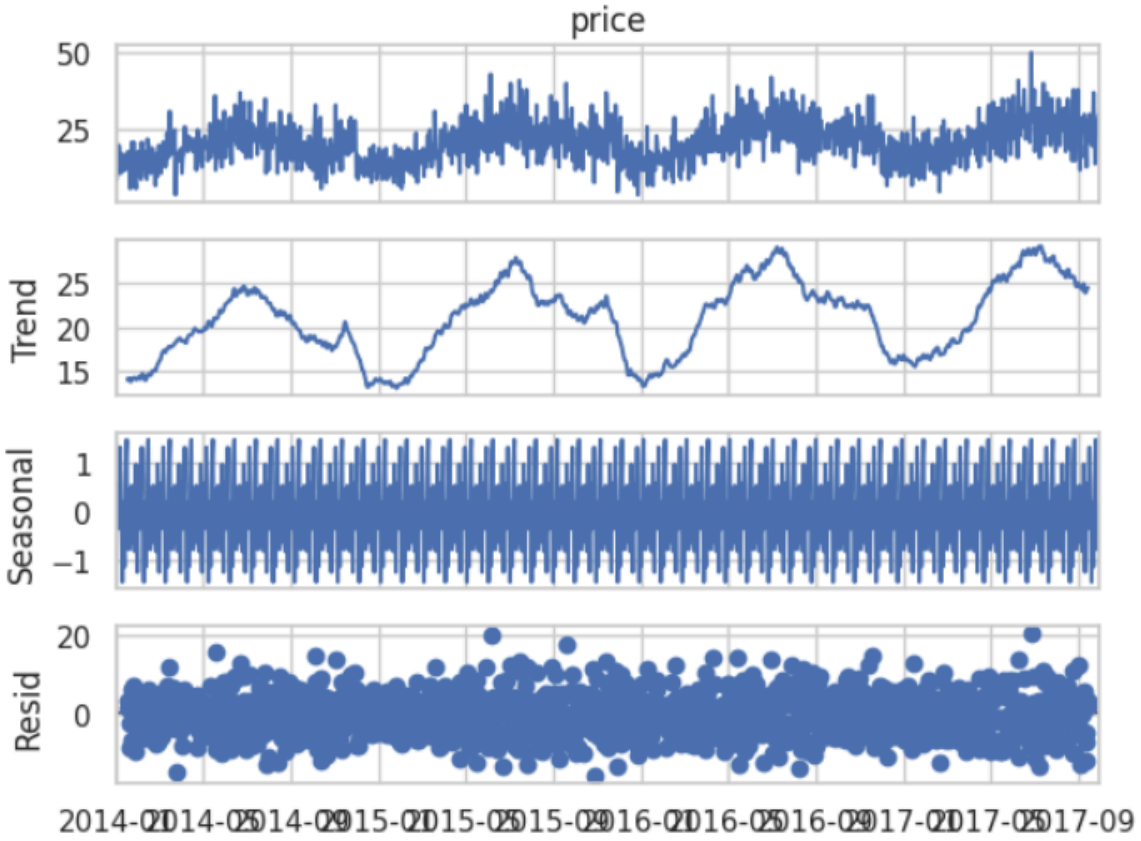


A line plot of prices over months reveals overall trends and seasonality.

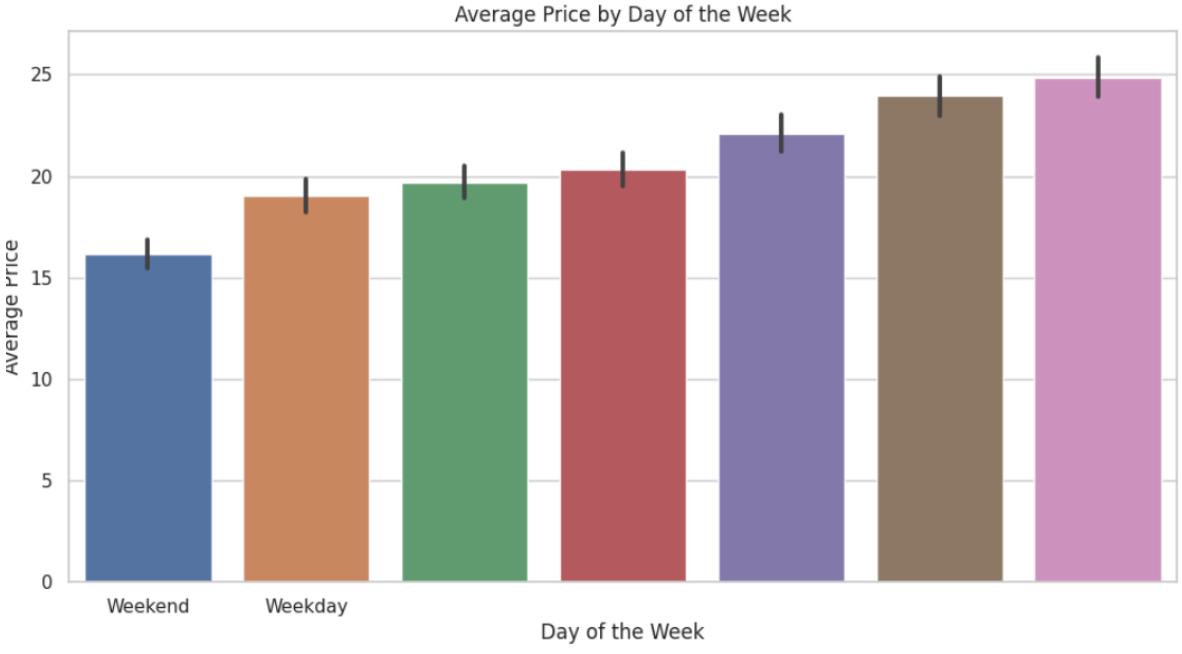


Seasonal Decomposition:

Decomposed the price data into trend, seasonal, and residual components using an additive and multiplicative model with a period of 30 days.

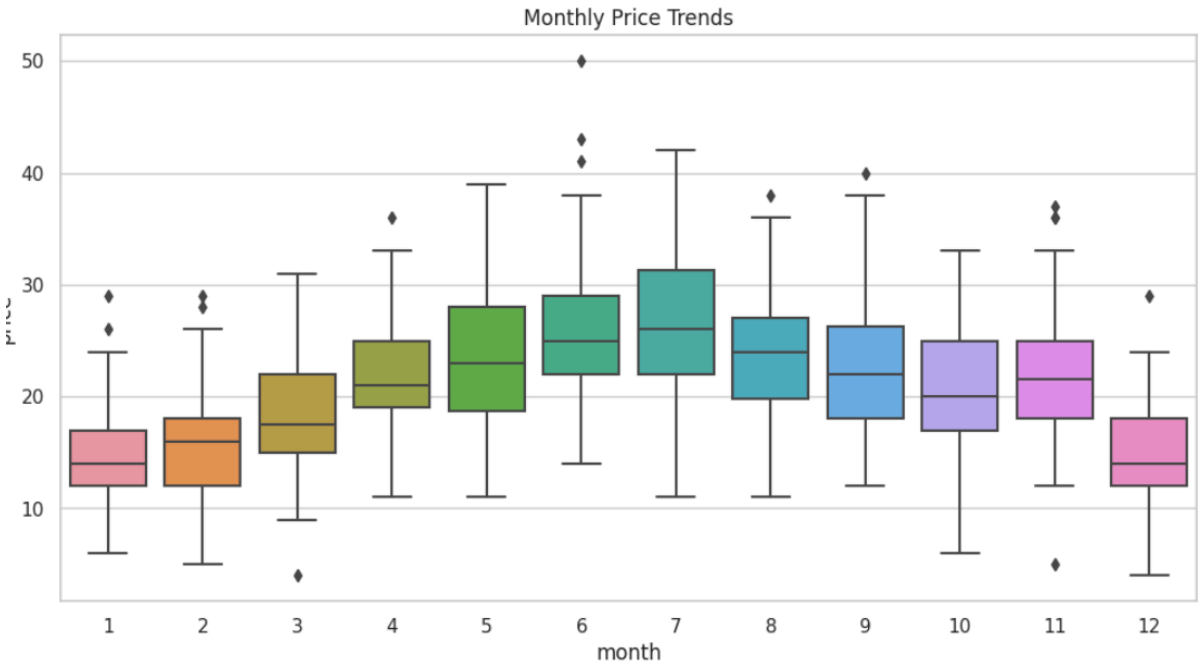


Monthly Trends:



Outlier Detection:

Boxplots of price vs. month highlight monthly price fluctuations.



Explored outliers with different whisker lengths (whis) in boxplots.

Stationarity Check
Stationarity is a key assumption for many time-series models. The Augmented Dickey-Fuller (ADF) test was applied:

A time series is said to be stationary if its statistical properties, such as mean, variance, and autocorrelation, do not change over time. Stationarity is crucial in time series analysis because many statistical methods and models, like ARIMA, assume that the data is stationary

If p-value > 0.05, the series is not stationary. Differencing was applied where necessary.

Predictive Modeling
ARIMA Model
Training and Validation Split:

The dataset was split into 80% training and 20% validation sets.

Model Fitting:

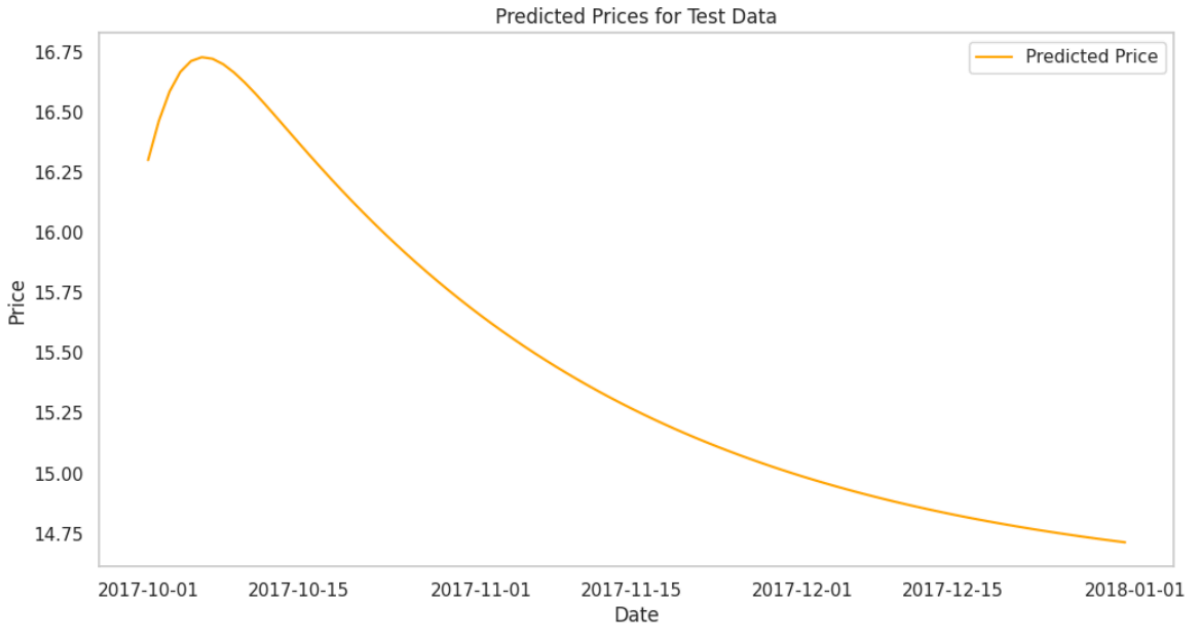
An ARIMA model with parameters (3, 0, 3) was trained.
Performance Evaluation:
Validation predictions were compared to actual prices using RMSE.
Box-Cox Transformation
To stabilize variance and make the series more stationary:
Applied inverse Box-Cox transformation after predictions.

Advanced Modeling Techniques
Prophet Model
Implementation:
Prophet, a time-series forecasting tool, was applied using the date and price columns.

LSTM Neural Network
Data Preparation:
Used a sliding window approach with 60 timesteps to generate sequences.
Scaled the price column using Min-Max scaling.
Model Architecture:
A sequential LSTM model was built with:
2 LSTM layers (50 units each, one with return_sequences=True)
Dropout layers for regularization
A dense layer for output.

Performance:
Evaluated accuracy for both training and validation sets.
Train Accuracy: 75.62%
Validation Accuracy: 79.26%

Results:
Generated future forecasts and visualized them.



Seasonality:
Clear seasonal patterns were observed in the price data, as indicated by the decomposition and Prophet results.

Outliers:
Outliers were detected, but they did not significantly affect the overall analysis.

Stationarity:
The series required transformations and differencing to achieve stationarity, as indicated by the ADF test.

Conclusion
The analysis provided deep insights into the time-series data. ARIMA and LSTM models performed well, with acceptable accuracy and error metrics. The Prophet model added another layer of interpretability and visualization. Future work could involve experimenting with hybrid models or ensembles to further improve prediction accuracy.I’ve tried and explored a few different methods for data analysis and forecasting. One of them is Simple Exponential Smoothing. This method focuses on giving more weight to recent data points

and less weight to older ones, which can help in making short-term forecasts. It's simple and easy to use, but it's not ideal for data with strong trends or seasonality, and it doesn't work as well for longer-term predictions.

I also worked with Random Forest Regression, which is a more advanced technique that uses multiple decision trees to improve accuracy. It's really good for handling complex data with non-linear relationships and can manage large datasets well. It's less likely to overfit compared to a single decision tree and even helps identify which features are most important. However, it's more computationally intensive and not as easy to interpret as a single tree.

Lastly, I explored Simple Moving Average. This is one of the easiest methods for smoothing out time series data to spot trends. It works by averaging a fixed number of past data points, making it simple and fast to calculate. However, it can be slow to react to sudden changes because of its fixed window size, and it may not be the best for data with sharp changes or seasonality.

Writtenby
Leelavamsikrishna
IMT2020111