Experiment Use Case Document for DiscountMate Price Prediction Model

Introduction:

The objective of this project is to develop a robust price prediction model for DiscountMate, utilizing comprehensive data from various stores and brands related to their inventory. The model will forecast future prices, considering factors such as inventory levels and market trends, to provide accurate price predictions. Additionally, it will incorporate a recommendation system to suggest products to customers, offering personalized and value-driven choices based on predicted pricing and available inventory. This model aims to enhance DiscountMate's pricing strategy, ensuring competitive pricing and customer satisfaction.

Dataset overview

Dataset found on GitHub and Explanation of the Datasets
1. Australia Grocery Product Dataset:
https://www.kaggle.com/datasets/thedevastator/grocery-product-prices-for-australian-states/data

This dataset provides information on grocery products available in Australia, including pricing information. The data was extracted from the Grocery department of coles.com.au, and includes a selected list of categories. Columns include postal code, category, subcategory, product group, product name, package price, price per unit, package size, estimated status, special status, stock status, retail price, product URL, brand, SKU number, run date, unit price, and unit price unit

2. Synthetic Australian Grocery Dataset:
https://www.kaggle.com/datasets/nathannguyendev/synthetic-australian-grocery-dataset

In order to have a dataset of products with price fluctuations between time of year and geo location, this data was created synthetically based on Australian Grocery Dataset.

Pre-process Time Series Data

1. **Data Exploration:**
   - Explore the dataset to understand its structure, frequency, and any missing values or anomalies.
   - I suggest joining both the datasets – Australia grocery and synthetic dataset to attain more features for model development. We need to select the features that make sense and drop the ones which cannot fit into the joined data. E.g date, price.
   - Plot the time series to visualize trends, seasonality, and patterns.

2. **Handling Missing Values:**
   - Identify missing values and decide on a strategy to handle them (e.g., forward fill, backward fill, interpolation). [Note – We have many missing values in our dataset Australia Grocery dataset]

3. **Outlier Detection and Treatment:**
   - Detect any outliers in the data and decide whether to remove them, transform them, or keep them based on their significance.

4. **Data Transformation:**
   - **Differencing:** Apply differencing to remove trends and make the time series stationary, if needed.
   - **Log Transformation:** Apply log transformation to stabilize variance if the series shows exponential growth.
   - **Seasonal Adjustment:** Decompose the series to remove seasonality.

5. **Resampling:**
   - Resample the data to ensure it has a consistent frequency (e.g., daily, weekly, monthly).
   - Aggregate or interpolate the data to match the required time interval.

6. **Stationarity Testing:**
   - **Stationarity**: A time series is said to be *stationary* if its statistical properties, such as mean, variance, and autocorrelation, remain constant over time. This means that the data does not exhibit trends or seasonal patterns that change over time. Stationarity is important because many time series forecasting models, like ARIMA, assume that the underlying data is stationary.
   - Perform tests like the Augmented Dickey-Fuller (ADF) test to check for stationarity.
   - If non-stationary, apply transformations such as differencing until the series becomes stationary.

7. **Feature Engineering:**
   - Create additional features like lagged values, rolling statistics (mean, standard deviation), and time-based features (e.g., month, day of the week) if required.
   - Consider external factors or exogenous variables that might influence the price (e.g., holidays, economic indicators).

8. **Scaling/Normalization:**
   - Normalize or standardize the data to improve model performance, especially for models like Neural Networks.

9. **Train-Test Split:**
   - Split the data into training and test sets. Ensure the split respects the temporal order, with the training set covering the earlier period and the test set covering the later period.

Models for Forecasting

1. **ARIMA (AutoRegressive Integrated Moving Average):**

- **Use Case:** Best for data without strong seasonal patterns but with trends or patterns that evolve over time.
- **Components:**
  - **AR (AutoRegressive):** Uses the dependency between an observation and a number of lagged observations.
  - **I (Integrated):** Uses differencing to make the series stationary.
  - **MA (Moving Average):** Uses the dependency between an observation and a residual error from a moving average model applied to lagged observations.
- **Advantages:** Simple to implement and interpret.
- **Disadvantages:** Assumes linear relationships and requires stationary data.

2. **SARIMA (Seasonal AutoRegressive Integrated Moving Average):**

- **Use Case:** Extends ARIMA to handle seasonal data, making it suitable for time series with strong seasonal patterns.
- **Components:**
  - Seasonal parts are analogous to the AR, I, and MA components but applied to seasonal lags.
  - Adds seasonality with parameters: Seasonal order (P, D, Q, s), where s is the length of the season.
- **Advantages:** Effectively captures seasonality.
- **Disadvantages:** Can become complex with many parameters to tune.

3. **SARIMAX (Seasonal AutoRegressive Integrated Moving-Average with eXogenous factors):**

- **Use Case:** Similar to SARIMA but includes exogenous variables (other variables that might influence the target variable) in the model.
- **Components:**
  - Combines the SARIMA model with external regressors.
  - Useful when you have additional factors that could influence the series (e.g., promotions, holidays).
- **Advantages:** Allows for the inclusion of additional information, improving forecasting accuracy.
- **Disadvantages:** Requires careful selection of exogenous variables and more complex to implement.

4. **Exponential Smoothing (ETS):**

- **Use Case:** Suitable for time series with level, trend, and seasonality components.
- **Components:**
  - **Error (E):** How much the model deviates from the actual data.
  - **Trend (T):** The upward or downward direction of the data.
  - **Seasonality (S):** Repeating cycles in the data.
- **Variants:**
  - **Simple Exponential Smoothing:** For data without trend or seasonality.
  - **Holt's Linear Trend Model:** For data with a trend but no seasonality.
  - **Holt-Winters Seasonal Model:** For data with both trend and seasonality.
- **Advantages:** Simple and effective, especially for shorter-term forecasts.
- **Disadvantages:** Less effective for long-term forecasts or data with complex patterns.

5. **Prophet:**

- **Use Case:** Developed by Facebook, Prophet is well-suited for time series with daily observations, holidays, and seasonality.
- **Components:**
  - Captures trends, seasonality, and holiday effects.
  - Handles missing data and outliers well.

- **Advantages:** Easy to use, handles non-linear trends and seasonality.
- **Disadvantages:** Requires careful tuning of parameters, especially for complex datasets.

6. **LSTM (Long Short-Term Memory) Networks:**

- **Use Case:** A type of Recurrent Neural Network (RNN) that is effective for time series with long-term dependencies and non-linear patterns.
- **Components:**
    - ➢ LSTM cells that maintain and update memory over time, capturing complex dependencies.
- **Advantages:** Captures long-term dependencies and non-linearities.
- **Disadvantages:** Requires large datasets and is computationally expensive.

7. **TBATS (Trigonometric Box-Cox ARMA Trend Seasonal):**

- **Use Case:** A model designed for complex seasonal patterns and high-frequency data.
- **Components:**
    - ➢ Incorporates Box-Cox transformation, ARMA errors, trigonometric seasonal components, and trend.
- **Advantages:** Handles multiple seasonalities and complex seasonality patterns.
- **Disadvantages:** Computationally intensive and more complex to interpret.

8. **Seasonal Decomposition of Time Series (STL):**

- **Use Case:** Decomposes the time series into seasonal, trend, and residual components, which can be forecasted separately.
- **Components:**
    - ➢ Seasonal component, trend component, and remainder (residual) component.
- **Advantages:** Flexible in handling seasonality and trends separately.
- **Disadvantages:** Not a standalone forecasting model but useful in pre-processing.

9. **Simple Moving Average (SMA):**

- **Use Case:** Basic model for smoothing and short-term forecasting.
- **Components:**
    - ➢ The average of the last n observations.
- **Advantages:** Simple and easy to implement.
- **Disadvantages:** Limited forecasting power, especially for long-term predictions.

Choosing the Right Model

- **The basic time series:** ARIMA
- **If our data has strong seasonal patterns:** SARIMA or SARIMAX.
- **If we need to incorporate external factors:** SARIMAX.
- **If our data shows complex seasonality:** TBATS or STL.
- **For neural networks:** LSTM, especially for complex and non-linear relationships.
- **For a simple, easy-to-use approach:** ETS or Prophet.

Note: You guys can research on any more models that can be used for time series price prediction and use it.

Evaluation Metrics

When evaluating time series models, various metrics can be used to measure the accuracy and performance of the predictions. Here are some common evaluation metrics:

1. Mean Absolute Error (MAE):

- **Definition:** The average of the absolute differences between predicted and actual values.
- **Formula:** $MAE = \frac{1}{n}\sum_{t=1}^{n}|Y_t - \hat{Y}_t|$
- **Use Case:** MAE gives an easy-to-understand measure of the prediction errors in the same units as the original data.

2. Mean Squared Error (MSE):

- **Definition:** The average of the squared differences between predicted and actual values.
- **Formula:** $MSE = \frac{1}{n}\sum_{t=1}^{n}(Y_t - \hat{Y}_t)^2$
- **Use Case:** MSE penalizes larger errors more than MAE, making it sensitive to outliers.

3. Root Mean Squared Error (RMSE):

- **Definition:** The square root of the MSE, providing an error metric in the same units as the original data.
- **Formula:** $$RMSE = \sqrt{\frac{1}{n}\sum_{t=1}^{n}(Y_t - \hat{Y}_t)^2}$$
- **Use Case:** RMSE is commonly used in time series forecasting for its interpretability and sensitivity to large errors.

4. Mean Absolute Percentage Error (MAPE):

- **Definition:** The average of the absolute percentage differences between predicted and actual values.
- **Formula:** $$MAPE = \frac{100\%}{n}\sum_{t=1}^{n}\left|\frac{Y_t - \hat{Y}_t}{Y_t}\right|$$
- **Use Case:** MAPE is useful for understanding the relative error in percentage terms, though it can be problematic with very small actual values.

5. Symmetric Mean Absolute Percentage Error (sMAPE):

- **Definition:** A modified version of MAPE that accounts for both under- and over-forecasting.
- **Formula:** $$sMAPE = \frac{100\%}{n}\sum_{t=1}^{n}\frac{|Y_t - \hat{Y}_t|}{(Y_t + \hat{Y}_t)/2}$$
- **Use Case:** sMAPE provides a more balanced error metric when both over- and under-predictions are equally important.

6. Mean Squared Logarithmic Error (MSLE):

- **Definition:** The average of the squared differences between the logarithms of predicted and actual values.
- **Formula:**

$$MSLE = \frac{1}{n}\sum_{t=1}^{n}\left(\log(1 + Y_t) - \log(1 + \hat{Y}_t)\right)^2$$

- **Use Case:** MSLE is particularly useful when the data spans several orders of magnitude, reducing the impact of large errors in large values.

7. Autocorrelation Function (ACF) of Residuals:

- **Definition:** Measures the correlation between residuals at different lags.
- **Use Case:** ACF of residuals helps check if there is any remaining autocorrelation in the residuals, indicating whether the model has captured all temporal dependencies.

8. R-squared (Coefficient of Determination):

- **Definition:** A statistical measure that indicates how well the model's predictions approximate the actual data points.
- **Formula:**

$$R^2 = 1 - \frac{\sum_{t=1}^{n}(Y_t - \hat{Y}_t)^2}{\sum_{t=1}^{n}(Y_t - \bar{Y})^2}$$

- **Use Case:** R-squared provides a measure of how much of the variance in the dependent variable is explained by the model, though it is less commonly used for time series due to its sensitivity to serial correlation.

9. Akaike Information Criterion (AIC):

- **Definition:** A metric that balances model fit and complexity, penalizing models with more parameters.
- **Formula:** $$AIC = 2k - 2\ln(L)$$
- **Use Case:** AIC is useful for model selection, particularly when comparing models with different numbers of parameters.

10. Bayesian Information Criterion (BIC):

- **Definition:** Similar to AIC, but with a stronger penalty for the number of parameters.
- **Formula:** $$BIC = k\ln(n) - 2\ln(L)$$
- **Use Case:** BIC is also used for model selection, favoring simpler models when compared to AIC.