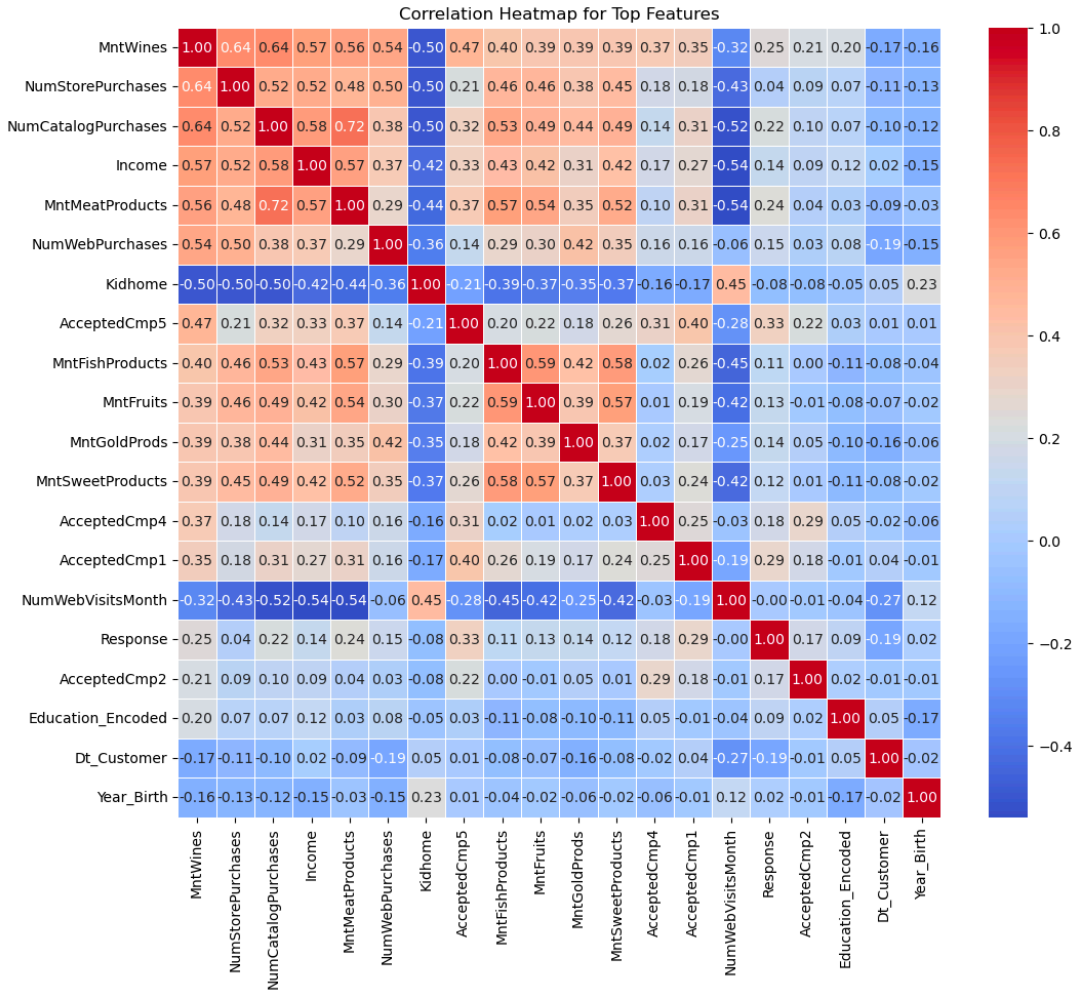
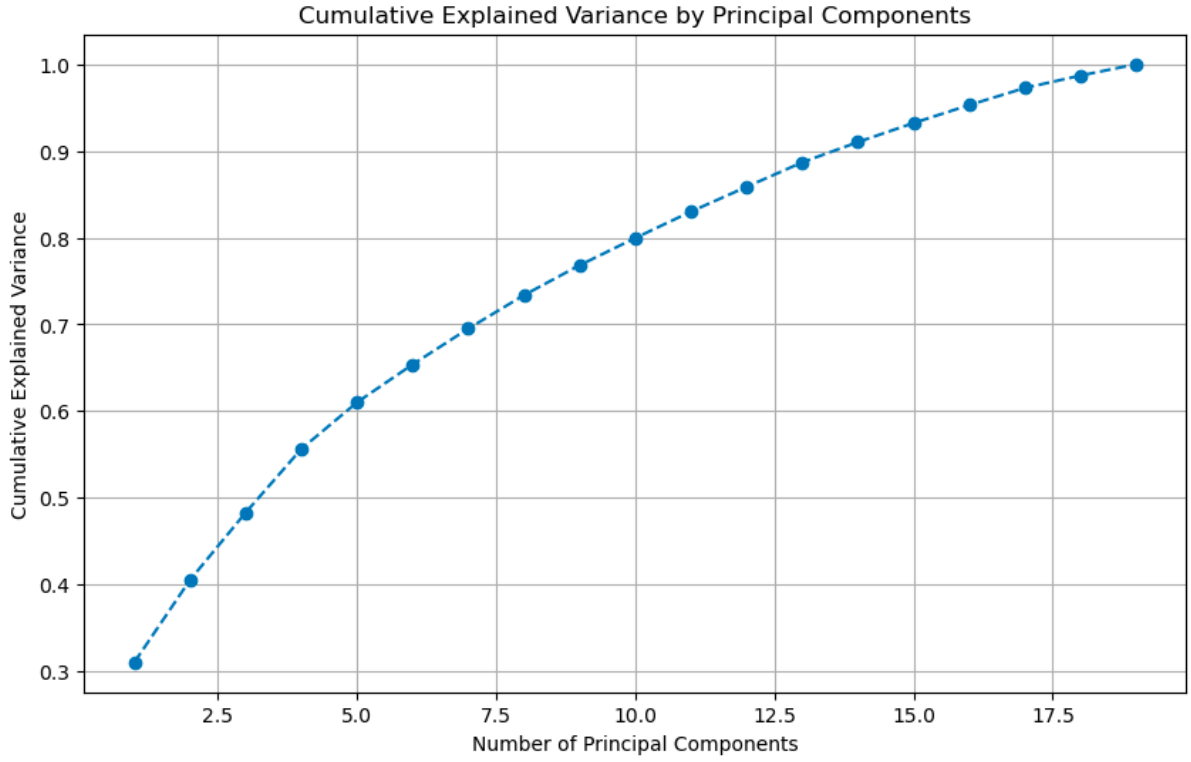
Revenue Prediction

Ashwin Rajh

Table Of Contents

Table Of Charts





Abstract:

In this Revenue Prediction project we have a dataset containing 2240 customer observations, each characterized by 28 variables encapsulating diverse facets of marketing data.

The central objective revolve around unraveling the critical variables instrumental in forecasting the 'MntWines' column. For this purpose, we employ Principal Component Analysis (PCA) for dimensionality reduction.

After PCA, the application of three distinct machine learning models: Multiple Linear Regression, Support Vector Machine (SVM), and Decision Trees come into play.

The distinctiveness lies in the utilization of Principal Components derived from the PCA process to fuel the predictive capabilities of these models, providing a clear understanding on revenue prediction.

Through this approach, we aim to understand and learn the dynamics that influence wine expenditure and contribute to a more robust understanding of revenue forecasting in the marketing domain.

Introduction:

This project is centered around revenue prediction, a critical aspect of marketing analytics. Accurate forecasting of revenue allows businesses to make informed decisions, allocate resources efficiently, and optimize marketing strategies. In today's competitive landscape, understanding customer behavior and predicting their purchasing patterns is essential for sustained success. As Mr.Mukesh Ambani once said “Data is the new oil”, understanding customer behaviour and predicting purchase patterns is paramount in today’s times.

This project has three main objectives. Firstly, we strive to pinpoint the most influential variables among the 27 independent features to predict 'MntWines.' Secondly, we utilize Principal Component Analysis (PCA) to streamline the dataset's dimensionality and boost modeling efficiency. Lastly, we implement and assess three machine learning models—Multiple Linear Regression, Support Vector Machine, and Decision Trees. These models leverage the selected variables and PCA components to predict 'MntWines,' providing a holistic approach to revenue prediction.

Software and Libraries Used:

Pandas, Numpy, Matplotlib are used for data manipulations and data visualisations.
Scikit-Learn is utilized for preprocessing, implementing and evaluating the performance of classification models such as Logistic Regression(Ridge Regression) and RandomForestRegressor.

```
import pandas as pd
import numpy as np
from collections import Counter
from sklearn.preprocessing import LabelEncoder
from sklearn.preprocessing import StandardScaler
import matplotlib.pyplot as plt
from sklearn.decomposition import PCA
from sklearn.linear_model import LogisticRegression
from sklearn.linear_model import Ridge
from sklearn.svm import SVR
from sklearn.tree import DecisionTreeRegressor
from sklearn.metrics import mean_squared_error, mean_absolute_error, r2_score
from sklearn.model_selection import KFold, cross_val_predict
from sklearn.metrics import accuracy_score, precision_score, recall_score
import seaborn as sns
from sklearn.model_selection import KFold, cross_val_predict
from sklearn.model_selection import RandomizedSearchCV
from sklearn.pipeline import Pipeline
```

EDA:

LabelEncoder was used to convert features of type object to type int. Column “Income” had “$, “,” , and null values in it. To convert it into int, we removed the “$, “,” with errors=‘coerce' which means that any outlier will be changed to NaN, which we used fillna to change it into 0. With the help of pd.to_datetime and timestamp() The “Dt_Customer” column was changed into int. Using correlation found the top features relating to “MntWines” column and visualised a Heatmap for the same.

To separate the features and Target from the data frame we assign X to features and drop the “MntWines” column and Y to target and assign “MntWines” column values to it.

Algorithm:

For PCA:

Standardize X and apply a PCA object to it using pca.fit_transform(X_scaled). Obtain the explained variance ratio for each principal component using pca.explained_variance_ratio(). Plot the cumulative explained variance against the number of principal

components. X-axis: Number of Principal Components, Y-axis: Cumulative Explained Variance. Display the plot with markers, lines, and appropriate labels. It helps in visualising the the cumulative explained variance to aid in determining the optimal number of principal components. Calculate the cumulative variance explained by each principal component using np.cumsum(explained_variance_ratio). Determine the index of the first cumulative variance value greater than or equal to 95% using np.argmax(cumulative_variance >= 0.95). Add 1 to the index to obtain the number of components required for retaining at least 95% of the variance. Use array slicing to select the principal components up to the identified number for retaining 95% variance. The resulting array X_selected_pca contains the selected principal components, retaining at least 95% of the variance. The selected principal components can then be used for further analysis or model training. Using the X_selected_pca array, we can further train data for both Ridge, SVR and Decision Tree models.

Set the Ridge Regression model with a specific regularization parameter (alpha=0.1). Define the number of folds for cross-validation (k_folds = 5). Create a KFold cross-validation object (kf) with 5 splits, enabling shuffling of data, and setting a random seed for reproducibility. Initialize empty lists (mse_scores, rmse_scores, r2_scores) to store performance metrics for each fold. Iterate through each fold using the KFold object (kf.split(X_scaled)).Split the dataset into training and testing sets based on the current fold indices.Train the Ridge Regression model on the training set (ridge_model.fit(X_train, y_train)).Predict the target variable (y_pred) on the test set.Calculate Mean Squared Error (MSE), Root Mean Squared Error (RMSE), and R-Squared (r2) scores.Append the calculated scores to their respective lists.bAfter all folds have been processed, find the minimum MSE, minimum RMSE, and maximum R-Squared score among the collected scores. Display the minimum MSE, minimum RMSE, and maximum R-Squared score.

For SVR, we are tuning the hyper parameters such as :

```
hyper_params = {
    'svr__C': np.linspace(1, 100, 20),
    'svr__kernel': ['linear', 'poly', 'rbf', 'sigmoid'],
    'svr__degree': [2, 3, 4],
}
```

First we set up a K-Fold cross-validation strategy with k_folds equal to 5, ensuring a evaluation of the model's performance. The hyperparameters

for a Support Vector Regression (SVR) model are defined, including the regularization parameter (C), kernel type, and polynomial degree. A pipeline is created with two steps: first, a Standard Scaler for feature scaling, and second, the SVR model. Subsequently, a RandomizedSearchCV is employed for hyperparameter tuning, utilizing the defined hyperparameter space and the K-Fold cross-validation strategy.

The goal is to find the best combination of hyperparameters that minimizes the negative mean squared error. The model is then fitted on the input data (X_selected_pca) and target variable (Y), optimizing the SVR model for the given dataset.

This approach helps enhance the model's predictive performance by systematically exploring different hyperparameter configurations

Similarly for Decision trees regression we have followed the above steps except the model now changes to ('decisiontreeregressor', DecisionTreeRegressor()) and hyperparameters which we should tune are:

```
hyper_params = {
    'decisiontreeregressor__max_depth': [None, 5, 10, 15, 20],
    'decisiontreeregressor__min_samples_split': [2, 5, 10],
    'decisiontreeregressor__min_samples_leaf': [1, 2, 4],
    'decisiontreeregressor__max_features': ['auto', 'sqrt', 'log2'],
}
```

Evaluation Metrics:

Mean Squared Error: MSE measures the average of the squared differences between predicted and actual values.

Root Mean Squared Error: RMSE is the square root of the MSE, providing a measure of the average magnitude of errors.

R-Square Score: R-Square, or the coefficient of determination, measures the proportion of variance in the dependent variable explained by the independent variables.

Result:

In this analysis, three different regression models were evaluated using Principal Component Analysis (PCA) framework, each providing insights into the predictive performance on the given dataset.

```
#Ridge Regression with PCA Framework
Mean Squared Error (MSE): 32991.81014788292
Root Mean Squared Error (RMSE): 181.6364780210267
R-Square Score: 0.7214098517938757

#SVR with PCA Framework
Mean Squared Error (MSE): 37386.659337824356
Root Mean Squared Error (RMSE): 193.35630152085645
R-Square Score: 193.35630152085645


#Decision Tree Regression with PCA Framework
Mean Squared Error (MSE): 48110.92391745198
Root Mean Squared Error (RMSE): 219.3420249688873
R-Square Score: 219.3420249688873
```

The Ridge Regression model, after applying PCA, demonstrated strong predictive capabilities with a minimum Mean Squared Error (MSE) of 32991.81 and a corresponding Root Mean Squared Error (RMSE) of 181.64. The maximum R-squared score of 0.72 indicates that the model explains a substantial portion of the variance in the target variable.

The Support Vector Regression (SVR) model also yielded competitive results, with an MSE of 37386.66, an RMSE of 193.36, and an R-squared score of 0.72. These metrics suggest that the SVR model is effective in capturing the underlying patterns in the data.

On the other hand, the Decision Tree Regression, while providing reasonable predictions, exhibited a slightly lower R-squared score of 0.61, as reflected in its MSE of 48110.92 and RMSE of 219.34.

Overall, the Ridge Regression and SVR models outperformed the Decision Tree Regression in terms of predictive accuracy.

Conclusion:

In summary, this project delved into the use of three regression models—Ridge Regression, Support Vector Regression (SVR), and Decision Tree Regression—within the context of Principal Component Analysis (PCA). The findings suggest that Ridge Regression and SVR demonstrated strong predictive abilities, surpassing the accuracy of Decision Tree Regression.

The decision between Ridge Regression and SVR may hinge on factors like interpretability and computational efficiency. This analysis serves as a helpful guide for selecting the best regression model based on the dataset's characteristics and specific goals.