PROJECT DOCUMENTATION

WEB TRAFFIC PREDICTION

Done by:

Sneha M

Aim:

To develop an analytical model for Web Traffic Prediction (WTP) using Time Series Analysis, Clustering, and Principal Component Analysis techniques.

Abstract:

Web Traffic Prediction (WTP) is a crucial task in the field of network engineering and operations, enabling effective planning and management of network resources. This project aims to develop an analytical model for WTP by employing Time Series Analysis, Clustering, and Principal Component Analysis (PCA) techniques.

The project begins by collecting a comprehensive dataset of historical website traffic data, which serves as the foundation for model training and evaluation. The website chosen is “Fandom.com.” Time Series Analysis is applied to uncover patterns, trends, and seasonality within the data, providing valuable insights into the dynamics of web traffic.

Clustering algorithms are utilized to group similar traffic patterns, enabling the identification of distinct traffic profiles. This step helps to understand different user behaviours and traffic characteristics, which can further enhance the accuracy of the prediction model.

To reduce the dimensionality of the dataset and extract meaningful features, Principal Component Analysis (PCA) is employed. PCA allows for the identification of the most relevant components of the traffic data, facilitating more efficient and accurate predictions.

The model's predictions offer valuable insights and recommendations for resource planning and management, facilitating more efficient and reliable operations in the online environment.

Concept Explanation:

Website traffic prediction refers to the process of forecasting the future volume and patterns of internet traffic to a website. It involves analysing historical data and applying predictive modelling techniques to estimate the future behaviour of website visitors.

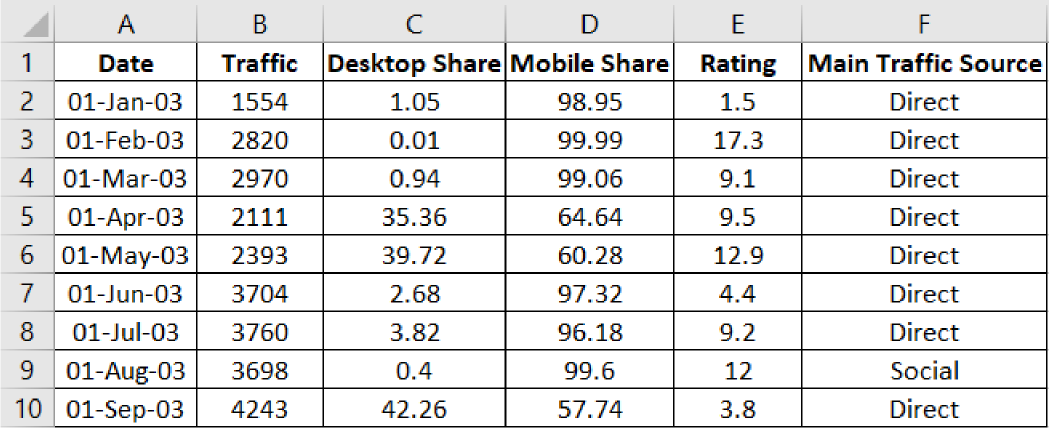
Website traffic prediction is a valuable tool for website owners, administrators, marketers, and businesses to optimize their resources, improve performance, enhance user experiences, and maximize revenue generation in the online ecosystem.

Predictive analysis involves using historical web traffic data and various analytical techniques to forecast and understand future web traffic patterns. Here are some key aspects of website trafficking in predictive analysis:

- Data Collection
- Data Pre-processing
- Time Series Analysis
- Performance Evaluation
    - Mean Absolute Error (MAE)
    - Mean Squared Error (MSE)
    - Mean Absolute Percentage Error (MAPE)
- Optimization and Improvement

Dataset:

The dataset used in this analysis has been downloaded from the Kaggle.com. The dataset is the internet traffic data of the website “Fandom.com” for the period of 1st January, 2003 to 1st April, 2022.



	A	B	C	D	E	F
1	**Date**	**Traffic**	**Desktop Share**	**Mobile Share**	**Rating**	**Main Traffic Source**
2	01-Jan-03	1554	1.05	98.95	1.5	Direct
3	01-Feb-03	2820	0.01	99.99	17.3	Direct
4	01-Mar-03	2970	0.94	99.06	9.1	Direct
5	01-Apr-03	2111	35.36	64.64	9.5	Direct
6	01-May-03	2393	39.72	60.28	12.9	Direct
7	01-Jun-03	3704	2.68	97.32	4.4	Direct
8	01-Jul-03	3760	3.82	96.18	9.2	Direct
9	01-Aug-03	3698	0.4	99.6	12	Social
10	01-Sep-03	4243	42.26	57.74	3.8	Direct

Packages Used:

- Pandas
- Numpy
- Sklearn
- Seaborn
- Statsmodels
- Matplotlib.pyplot

Code:

```
import pandas as pd
import numpy as np
from sklearn.preprocessing import LabelEncoder, StandardScaler
import matplotlib.pyplot as plt
from sklearn.decomposition import PCA
import seaborn as sns
from sklearn.cluster import KMeans
from statsmodels.tsa.arima.model import ARIMA
from sklearn.metrics import mean_absolute_error, mean_squared_error, mean_absolute_percentage_error
import warnings
# Load the dataset
web_traffic_data = pd.read_csv("Traffic Dataset.csv")
# Pre-processing
web_traffic_data['Date'] = pd.to_datetime(web_traffic_data['Date'])
web_traffic_data.set_index('Date', inplace=True)
print(web_traffic_data.head())  # View the data
# missing values
missing_values = web_traffic_data.isnull().sum()
print("Number of missing values:")
print(missing_values)
# Fill missing values
web_traffic_data['Rating'].fillna(web_traffic_data['Rating'].mean(), inplace=True)
# Label Encode
label_encoder = LabelEncoder()
web_traffic_data['Main Traffic Source'] = label_encoder.fit_transform(web_traffic_data['Main Traffic Source'])
# category mapping
```

```
category_mapping=dict(zip(label_encoder.classes,label_encoder.transform(label_encoder.classes_)))
for category, encoded_value in category_mapping.items():
    print(f"Category: {category} - {encoded_value}")
# descriptive
print(web_traffic_data.head())
print(web_traffic_data.describe())

# Exploratory Analysis
desktop_share = web_traffic_data['Desktop Share']
mobile_share = web_traffic_data['Mobile Share']
# Line plot for Desktop and Mobile Shares
plt.plot(web_traffic_data.index, desktop_share, label='Desktop Share', color='blue')
plt.plot(web_traffic_data.index, mobile_share, label='Mobile Share', color='orange')
plt.xlabel('Date')
plt.ylabel('Share')
plt.title('Desktop Share vs Mobile Share')
plt.xticks(rotation=45)
plt.legend()
plt.show()
# Rating
plt.fill_between(web_traffic_data.index, web_traffic_data['Rating'], color='skyblue', alpha=0.4)
plt.plot(web_traffic_data.index, web_traffic_data['Rating'], color='blue')
plt.xlabel('Date')
plt.ylabel('Rating')
plt.title('Rating')
plt.xticks(rotation=45)
plt.show()
# Main Traffic Source vs Traffic
```

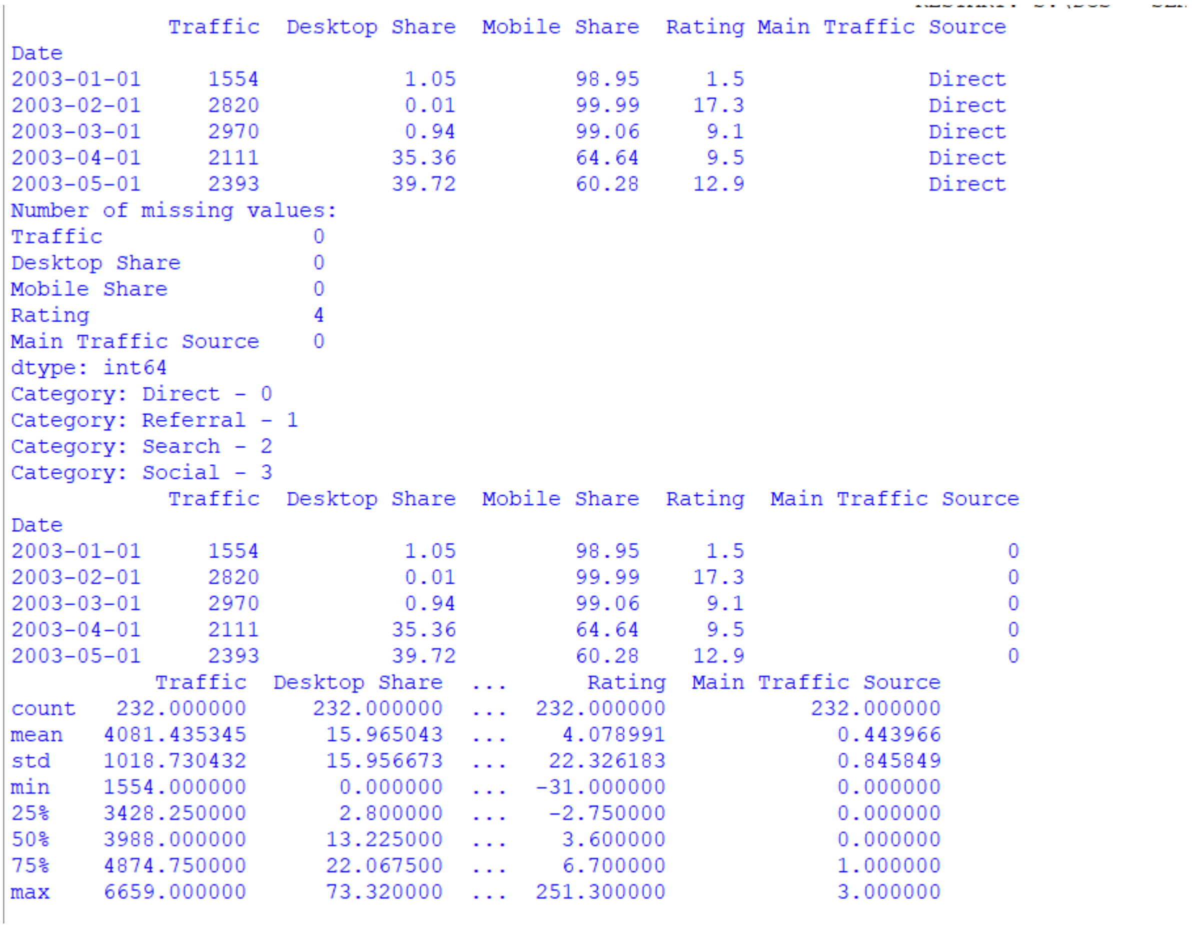
```
traffic_source_labels = label_encoder.inverse_transform(web_traffic_data['Main Traffic Source'].unique())
traffic_source_counts = web_traffic_data['Main Traffic Source'].value_counts().sort_index()
plt.bar(traffic_source_labels, traffic_source_counts)
plt.bar(traffic_source_labels, traffic_source_counts, color=['orange', 'green', 'blue', 'purple'])
plt.xlabel('Main Traffic Source')
plt.ylabel('Traffic')
plt.title('Exploratory Analysis: Main Traffic Source vs Traffic')
plt.xticks(rotation=45)
plt.show()
warnings.filterwarnings("ignore")

# ARIMA
# Split the data
train_data = web_traffic_data.iloc[:152]
test_data = web_traffic_data.iloc[152:]
train_data = web_traffic_data['Traffic']
p, d, q = 1,0,1
# fit the ARIMA model
arima_model = ARIMA(train_data, order=(p, d, q))
arima_model_fit = arima_model.fit()
# predictions
predictions = arima_model_fit.predict(start=152, end=231)
print(predictions)
# Calculate Mean Absolute Error, Squared Error and Error Percentage
test_data = train_data[152:233]
print(test_data)
mae = mean_absolute_error(test_data, predictions)
print("Mean Absolute Error (MAE):", mae)
mse = mean_squared_error(test_data, predictions)
```

```
print("Mean Squared Error (MSE):", mse)
mape = mean_absolute_percentage_error(test_data, predictions)
print("Mean Absolute Percentage Error (MAPE):", mape)
plt.plot(test_data.index, test_data.values, label='Actual')
plt.plot(predictions.index, predictions.values, label='Predicted')
plt.xlabel('Past')
plt.ylabel('Traffic')
plt.title('ARIMA Model Forecast')
plt.legend()
plt.show()
# PCA and K-means Clustering
X = web_traffic_data[['Traffic', 'Desktop Share', 'Mobile Share', 'Rating', 'Main Traffic Source']]
y = web_traffic_data['Traffic']
scaler = StandardScaler()
scaled_data = scaler.fit_transform(X)
pca = PCA(n_components=2)
X_pca = pca.fit_transform(scaled_data)
print(X_pca)
scatter = plt.scatter(X_pca[:, 0], X_pca[:, 1], c=y, cmap='rainbow')
plt.colorbar(scatter, label='Traffic')
plt.xlabel('Principal Component 1')
plt.ylabel('Principal Component 2')
plt.show()
# Correlation Matrix
scaled_data = pd.DataFrame(scaled_data, columns=X.columns)
sns.heatmap(scaled_data.corr(), annot=True, cmap='coolwarm')
plt.title('Correlation Matrix')
plt.show()
components = pca.components_
```

```
print("Principal Components:")
for i, component in enumerate(components):
    print(f"Principal Component {i+1}: {component}")
k = 2
kmeans = KMeans(n_clusters=k, n_init=10)
kmeans.fit(X_pca)
colors = ['yellow', 'red']
markers = ['o', 's']
# Scatter plot with cluster colors and markers
for i in range(k):
    plt.scatter(X_pca[kmeans.labels_ == i, 0], X_pca[kmeans.labels_ == i, 1], color=colors[i],
marker=markers[i],label=f'Cluster {i+1}')
# cluster centers
plt.scatter(kmeans.cluster_centers_[:, 0], kmeans.cluster_centers_[:, 1], color='black',
marker='X', label='Cluster Centers')
plt.xlabel("Principal Component 1")
plt.ylabel("Principal Component 2")
plt.title("K-means Clustering")
plt.legend()
plt.show()
```

Output with Plots:

#pre-processing

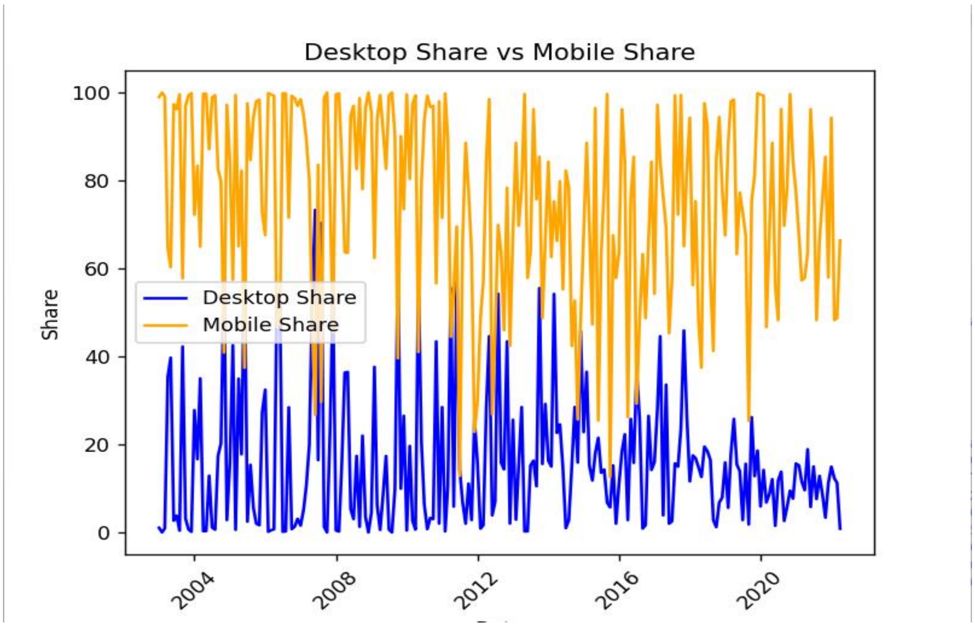


```
            Traffic  Desktop Share  Mobile Share  Rating Main Traffic Source
Date
2003-01-01     1554           1.05         98.95     1.5              Direct
2003-02-01     2820           0.01         99.99    17.3              Direct
2003-03-01     2970           0.94         99.06     9.1              Direct
2003-04-01     2111          35.36         64.64     9.5              Direct
2003-05-01     2393          39.72         60.28    12.9              Direct
Number of missing values:
Traffic                0
Desktop Share          0
Mobile Share           0
Rating                 4
Main Traffic Source    0
dtype: int64
Category: Direct - 0
Category: Referral - 1
Category: Search - 2
Category: Social - 3
            Traffic  Desktop Share  Mobile Share  Rating  Main Traffic Source
Date
2003-01-01     1554           1.05         98.95     1.5                    0
2003-02-01     2820           0.01         99.99    17.3                    0
2003-03-01     2970           0.94         99.06     9.1                    0
2003-04-01     2111          35.36         64.64     9.5                    0
2003-05-01     2393          39.72         60.28    12.9                    0
           Traffic  Desktop Share  ...      Rating  Main Traffic Source
count   232.000000     232.000000  ...  232.000000           232.000000
mean   4081.435345      15.965043  ...    4.078991             0.443966
std    1018.730432      15.956673  ...   22.326183             0.845849
min    1554.000000       0.000000  ...  -31.000000             0.000000
25%    3428.250000       2.800000  ...   -2.750000             0.000000
50%    3988.000000      13.225000  ...    3.600000             0.000000
75%    4874.750000      22.067500  ...    6.700000             1.000000
max    6659.000000      73.320000  ...  251.300000             3.000000
```

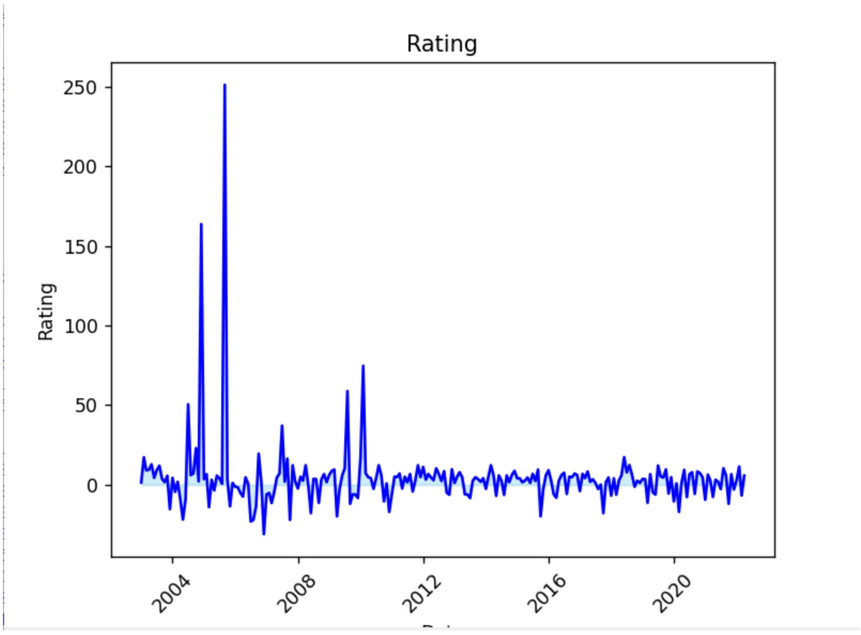
First, the first 5 rows of the dataset are printed to give an overview of how the dataset is and then the number of missing values in each column is found and replaced with the mean of the respective column. Then the column ‘Main Traffic Source’ labels are converted into numerical values and the dataset is again printed to show how the dataset has changed after treating these values. Finally, the summary statistics of the dataset is printed to know the overview of the data available.

# Exploratory Analysis:

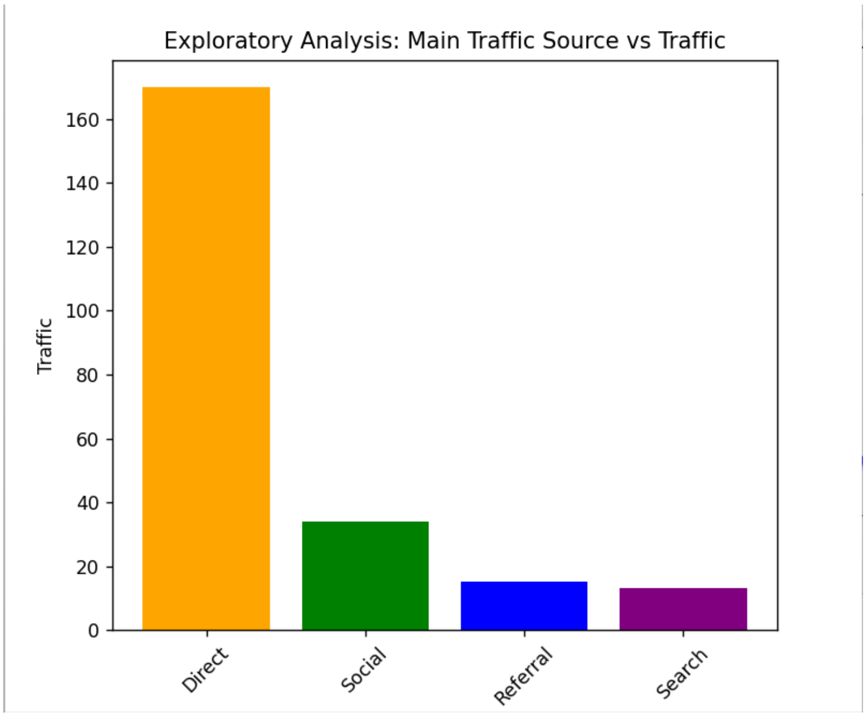
# Mobile vs Desktop Shares



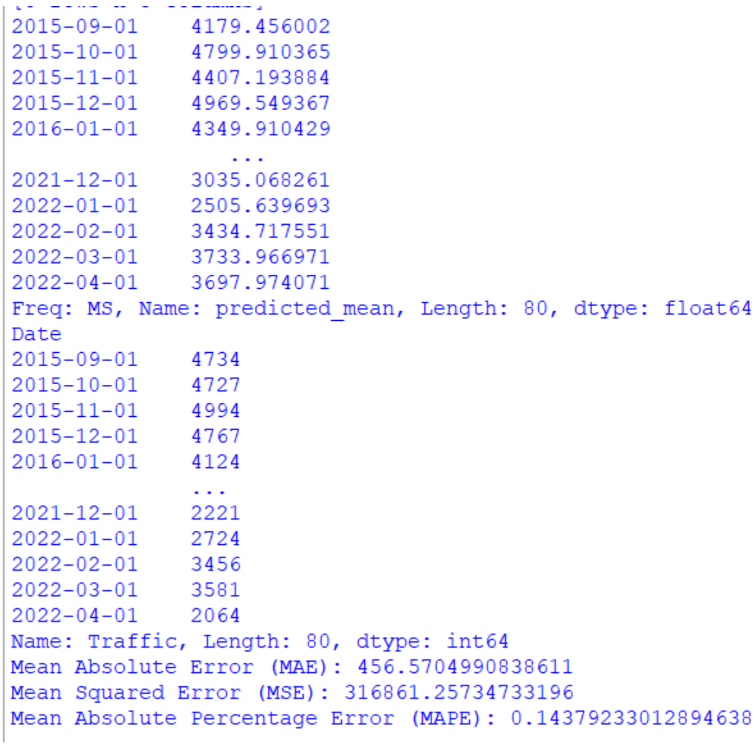
# Rating in each month



# Main Traffic Source of Visits



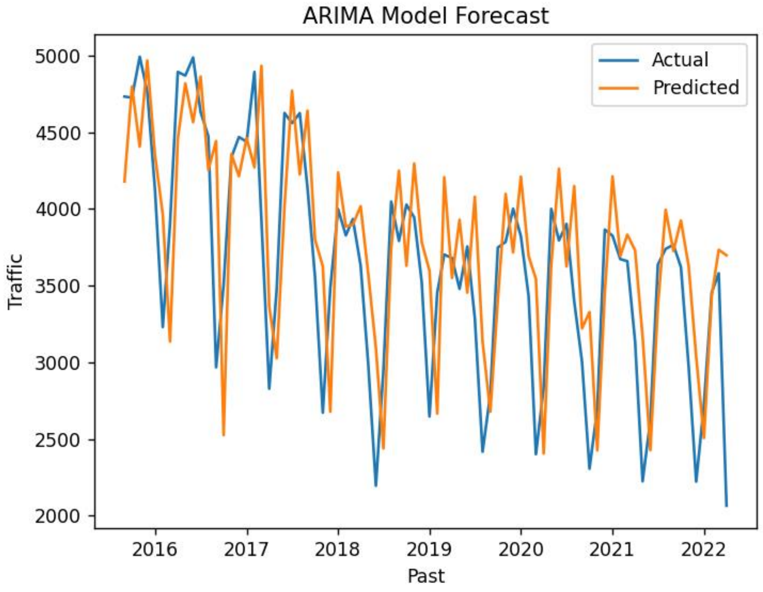
# Time Series – ARIMA



```
2015-09-01    4179.456002
2015-10-01    4799.910365
2015-11-01    4407.193884
2015-12-01    4969.549367
2016-01-01    4349.910429
                  ...
2021-12-01    3035.068261
2022-01-01    2505.639693
2022-02-01    3434.717551
2022-03-01    3733.966971
2022-04-01    3697.974071
Freq: MS, Name: predicted_mean, Length: 80, dtype: float64
Date
2015-09-01    4734
2015-10-01    4727
2015-11-01    4994
2015-12-01    4767
2016-01-01    4124
              ...
2021-12-01    2221
2022-01-01    2724
2022-02-01    3456
2022-03-01    3581
2022-04-01    2064
Name: Traffic, Length: 80, dtype: int64
Mean Absolute Error (MAE): 456.5704990838611
Mean Squared Error (MSE): 316861.25734733196
Mean Absolute Percentage Error (MAPE): 0.14379233012894638
```

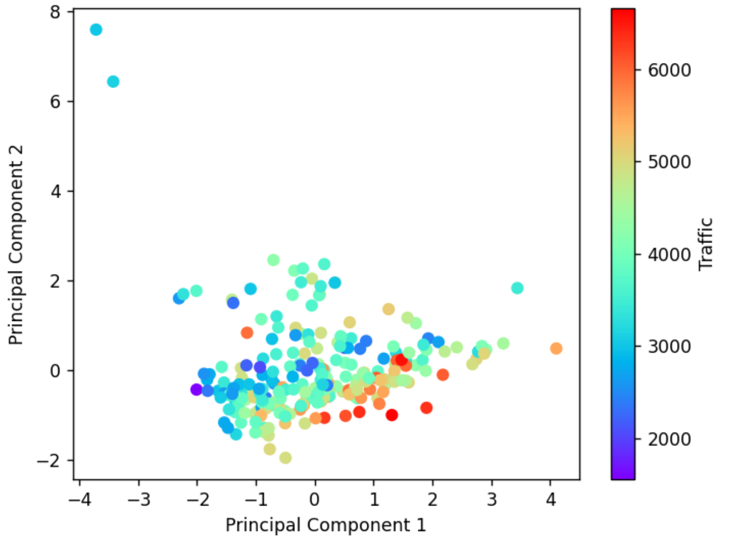
Mean Absolute Error, Mean Squared Error and Mean Absolute Percentage Error are estimated.

These metrics help in quantifying the accuracy and effectiveness of the predictive models.

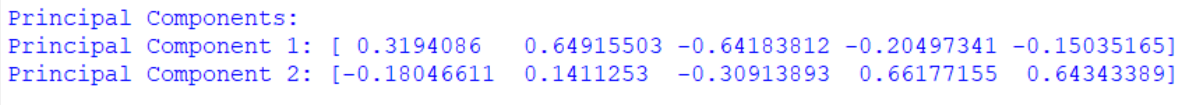


# PCA

Dimensional reduction:

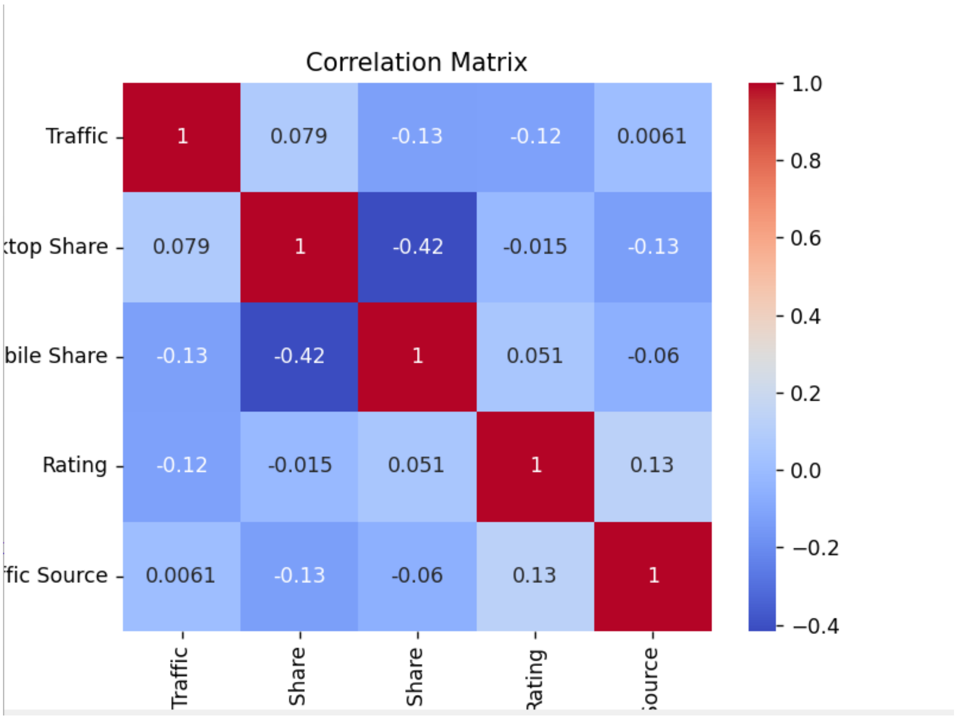


Principal Components:

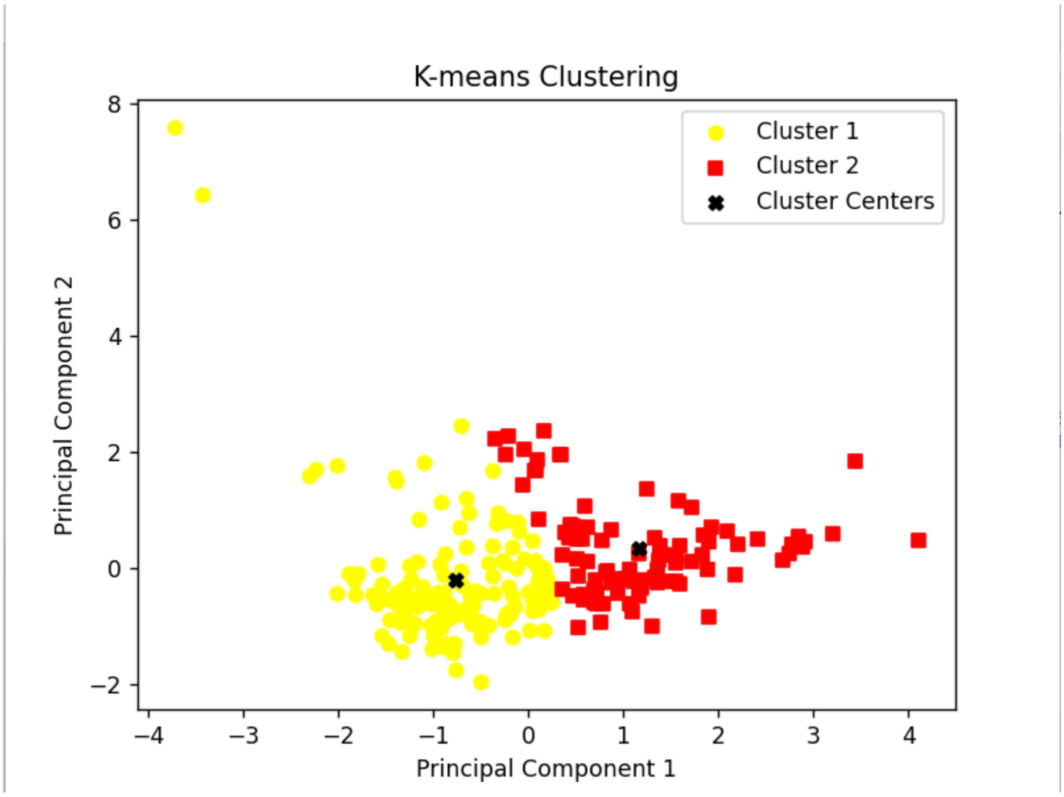


```
Principal Components:
Principal Component 1: [ 0.3194086   0.64915503 -0.64183812 -0.20497341 -0.15035165]
Principal Component 2: [-0.18046611  0.1411253  -0.30913893  0.66177155  0.64343389]
```

Correlation Matrix:



# Clustering – K-Means



Inference:

1. **Pre-processing:** There are missing values in the 'Rating' column, with a total of 4 missing values are filled with the mean value of the column. The 'Date' column is converted to datetime format and set as the index. The 'Main Traffic Source' column is label encoded using sklearn's Label Encoder. This step ensures that the categorical data can be used as input for various machine learning models and statistical analyses. Overall, these aim to prepare the dataset for further analysis.

2. **Descriptive Statistics:**
   - The dataset contains 232 observations.
   - The mean traffic is approximately 4081.44, with a standard deviation of 1018.73.
   - The 'Desktop Share' ranges from 0 to 73.32, while the 'Mobile Share' ranges from 26.68 to 100.
   - The 'Rating' ranges from -31 to 251.3, with a mean of 4.08.
   - Most of the data belongs to the 'Direct' category in the 'Main Traffic Source' column, as indicated by the higher count and 75th percentile value.

3. **ARIMA Model:**

The ARIMA model is fitted using the training data, and predictions are made for the test data. The order of the ARIMA model is specified as (1, 0, 1), indicating an autoregressive (AR) order of 1, a differencing (1) order of 0, and a moving average (MA) order of 1. The ARIMA model is used to generate predictions for the time period from index 152 to 231 using the predict function.

Mean Absolute Error (MAE), Mean Squared Error (MSE), and Mean Absolute Percentage Error (MAPE) are calculated to evaluate the performance of the model. The error percentage is 14 indicating the average difference between the actual and predicted values. The error rate between the predicted and the actual values of the dataset are relatively lower compared to the other possibilities of fitting the model which indicates better model performance.

The line plot shows the actual and predicted traffic values, providing a visual representation of the forecast. According to these values and plot, this ARIMA model captures the underlying patterns and trends in a well-defined manner appropriate to the dataset.

4. PCA and K-Means Clustering:

Principal Component Analysis (PCA) is performed to reduce the dimensionality of the data to **2 components**. A **scatter plot** is created to visualize the data points, where the colours represent the traffic values. It shows the distribution and clustering patterns in the reduced dimension space.

A **correlation matrix** heatmap is generated to visualize the relationships between the features. The correlation coefficient measures the strength and direction of the linear relationship between two variables. It ranges from -1 to 1, where -1 indicates a perfect negative correlation, 1 indicates a perfect positive correlation, and 0 indicates no correlation. The relationship between the features ‘Traffic’ and ‘Desktop Share’ has a positive correlation and has a good relationship comparing to mobile shares.

K-means clustering with k=2 clusters is applied to the reduced PCA data. The scatter plot shows the clustered data points, where each cluster is represented by a different colour and marker shape. The cluster centres are also plotted. The clustering analysis aims to segment the web traffic data based on its features, including 'Traffic', 'Desktop Share', 'Mobile Share', 'Rating', and 'Main Traffic Source'. Each cluster represents a distinct segment of web traffic with similar characteristics. By analysing the scatter plot, we can observe how the data points are grouped into clusters and their distribution across the principal components. Clusters that are closer together indicate similar patterns or behaviours among the corresponding web traffic data points.

The ARIMA model used in the analysis is a suitable choice for predicting and analyzing the future values of website visits and shares. The correlation matrix generated by PCA confirms that the desktop shares have a stronger influence compared to mobile shares. Additionally, the exploratory analysis reveals that the main traffic source for the website is primarily "Direct," indicating that the website is easily accessible to users.

Using the developed model, we can forecast future website visits and make informed decisions to improve its performance. By understanding the patterns and trends in the data, we can optimize the website's content, marketing strategies, and user experience to attract more visitors. The model provides valuable insights for identifying potential areas of improvement and implementing targeted measures to enhance the website's performance.