Alliance School of Advanced Computing

Department of Computer Science & Engineering Design

MICRO PROJECT

Course Code: 4CS1220

Course Title: Data Analytics

Semester: 04

Name: G.Sai Madhu Hasitha

Registration Number: 2023BCSE07AED320

Class: AIML-D

1.Salary Analysis:

This project aims to explore and understand the factors influencing salaries across various industries and roles. You have to uncover insights into compensation trends and disparities by analysing datasets containing salary information along with variables like education level, years of experience, job title, and location.

This project would involve

i. cleaning the data,

ii. performing exploratory data analysis to identify patterns and outliers,

iii. statistical tests to assess the impact of different factors on salaries.

iv. Visualization tools could be employed to present findings in an accessible manner.

Data cleaning

Get a dataset containing salary data and attributes such as education level, years of experience, job role, and location. For instance, a dataset may have columns such as

Salary (continuous numerical data),Education Level , Years of Experience, Job Title , Location.

```
import pandas as pd

# Load dataset
file_path = "salary_dataset.csv"
df = pd.read_csv(file_path)

print(df.head())
```

```
  Education Level  Years of Experience                  Job Title Location  \
0             PhD                   23                    Analyst  Germany
1      Bachelor's                    3  Machine Learning Engineer    India
2             PhD                   19  Machine Learning Engineer  Germany
3             PhD                    8  Machine Learning Engineer   Canada
4      Bachelor's                    5            Product Manager       UK

     Industry  Salary
0      Retail  [illegible]
1     Finance  [illegible]
2        Tech  112779
3  Healthcare  [illegible]
4  Healthcare  [illegible]
```

```
import pandas as pd

# Load dataset
file_path = "salary_dataset.csv"
df = pd.read_csv(file_path)

# Remove duplicates
df = df.drop_duplicates()

# Fill missing values with median
df["Salary"] = df["Salary"].fillna(df["Salary"].median())
df["Years of Experience"] = df["Years of Experience"].fillna(df["Years of Experience"].median())

# Save cleaned dataset
df.to_csv(file_path, index=False)

print("Data cleaning completed and saved in 'salary_data.csv'.")
```

```
Data cleaning completed and saved in 'salary_data.csv'.
```

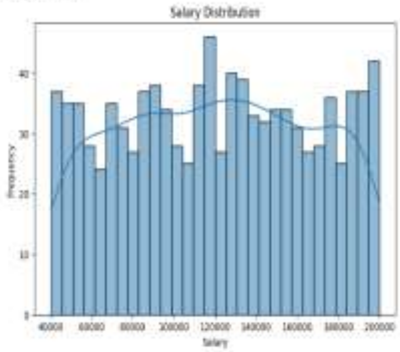
Clean and transform the data to prepare the dataset.Find rows containing missing or incomplete information and determine how to deal with them (e.g., dropping or imputing values).Make the data consistent.Ensure numerical columns are properly formatted as numeric data types, and categorical variables

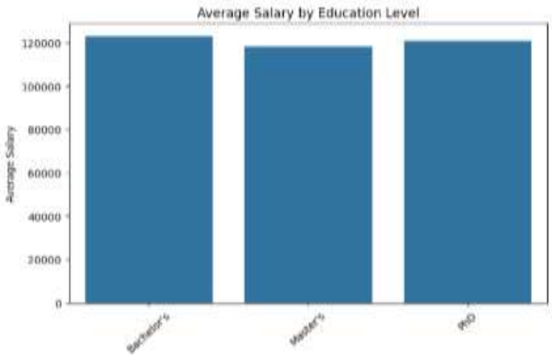
Exploratory Data Analysis (EDA)

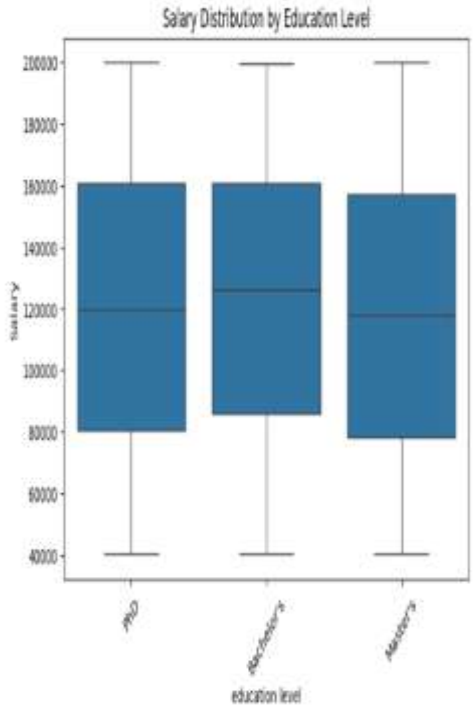
Determine patterns, trends, and outliers in the data.Compute elementary statistics for continuous variables such as salary and experience.Graph histograms or box plots for salary and experience to view distributions and detect outliers.Compare salary variations between various job titles, education levels, and locations based on bar charts or pivot tables.Investigate correlations between continuous variables (e.g., years of experience vs. salary) with scatter plots or correlation matrices.Investigate correlations between continuous variables with scatter plots or correlation matrices.

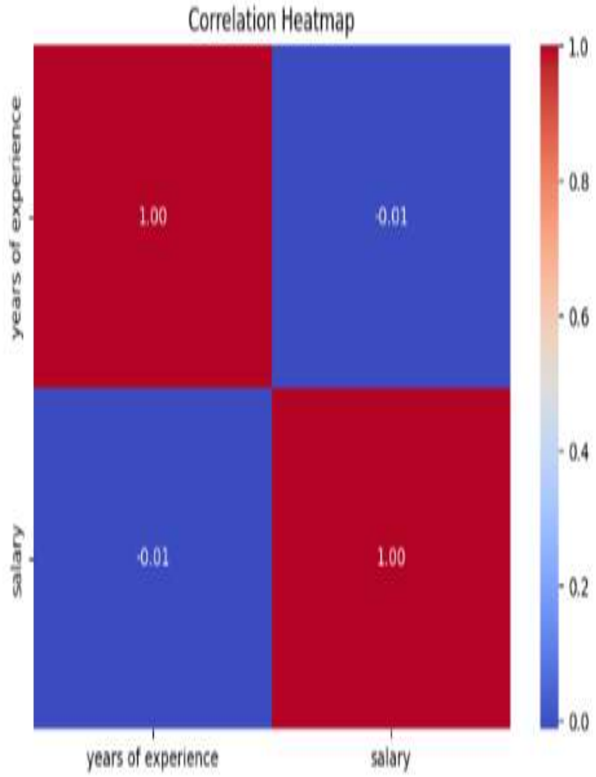
```python
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
file_path = "salary_dataset.csv"
df = pd.read_csv(file_path)
df.columns = df.columns.str.strip().str.lower()
print("Available Columns:", df.columns.tolist())
print("\nSummary Statistics:\n", df.describe())
print("\nMissing Values:\n", df.isnull().sum())
duplicates = df.duplicated().sum()
print("\nDuplicate Rows:", duplicates)
plt.figure(figsize=(8,5))
sns.histplot(df["salary"], bins=30, kde=True)
plt.title("Salary Distribution")
plt.xlabel("Salary")
plt.ylabel("Frequency")
plt.show()
if "years of experience" in df.columns:
    plt.figure(figsize=(8,5))
    sns.scatterplot(x=df["years of experience"], y=df["salary"])
    plt.title("Salary vs. Years of Experience")
    plt.xlabel("Years of Experience")
    plt.ylabel("Salary")
    plt.show()
else:
    print("\n'Years of Experience' column not found. Skipping experience analysis.")
education_col = next((col for col in df.columns if "education" in col), None)
if education_col:
    df_grouped = df.groupby(education_col)["salary"].mean().reset_index()
    plt.figure(figsize=(8,5))
    sns.barplot(x=education_col, y="salary", data=df_grouped)
    plt.title("Average Salary by Education Level")
    plt.xlabel(education_col)
    plt.ylabel("Average Salary")
    plt.xticks(rotation=45)
    plt.show()
else:
    print("\nNo 'Education Level' column found. Skipping education analysis.")
if education_col:
    plt.figure(figsize=(8,5))
    sns.boxplot(x=df[education_col], y=df["salary"])
    plt.title("Salary Distribution by Education Level")
    plt.xlabel(education_col)
    plt.ylabel("Salary")
    plt.xticks(rotation=45)
    plt.show()
numeric_df = df.select_dtypes(include=['number'])
correlation_matrix = numeric_df.corr()
print("\nCorrelation Matrix:\n", correlation_matrix["salary"])
plt.figure(figsize=(8,5))
sns.heatmap(correlation_matrix, annot=True, cmap="coolwarm", fmt=".2f")
plt.title("Correlation Heatmap")
plt.show()
```

```
Available Columns: ['education level', 'years of experience', 'job title', 'location', 'industry', 'salary']

Summary Statistics:
       years of experience         salary
count          1000.000000    1000.000000
mean             14.815000  120675.425000
std               8.490101   46266.059762
min               1.000000   40137.000000
25%               7.000000   83151.000000
50%              15.000000  120492.500000
75%              22.000000  160318.250000
max              29.000000  199951.000000

Missing Values:
 education level        0
years of experience    0
job title              0
location               0
industry               0
salary                 0
dtype: int64

Duplicate Rows: 0
```









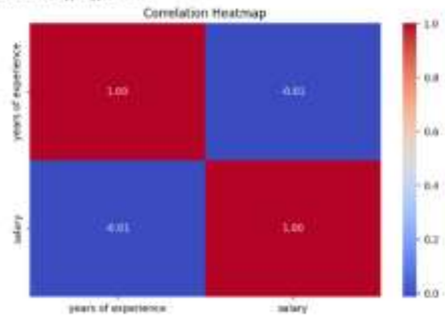

```
Correlation Matrix:
 years of experience   -0.013154
salary                 1.000000
Name: salary, dtype: float64
```

Statistical Analysis

Hypothesis testing and determining the effect of various factors on salaries.Apply statistical tests such as t-tests or ANOVA to determine whether there are any significant differences in salaries based on categorical variables such as education level or job role.Run linear regression to see the effect of continuous variables such as years of experience and level of education on pay.Apply the test to find associations among categorical variables (for example, job function and location).

```
import pandas as pd
import scipy.stats as stats
import seaborn as sns
import matplotlib.pyplot as plt

file_path = "salary_dataset.csv"
df = pd.read_csv(file_path)

df.columns = df.columns.str.strip().str.lower()

salary_col = "salary"
experience_col = "years of experience"
education_col = next((col for col in df.columns if "education" in col), None)

numeric_df = df.select_dtypes(include=['number'])
if salary_col in numeric_df.columns:
    print("\nCorrelation Matrix:\n", numeric_df.corr()[salary_col])

    plt.figure(figsize=(8,5))
    sns.heatmap(numeric_df.corr(), annot=True, cmap="coolwarm", fmt=".2f")
    plt.title("Correlation Heatmap")
    plt.show()

if education_col:
    unique_levels = df[education_col].nunique()
    if unique_levels == 2:
        group1, group2 = df[education_col].unique()
        salaries_group1 = df[df[education_col] == group1][salary_col]
        salaries_group2 = df[df[education_col] == group2][salary_col]

        t_stat, p_value = stats.ttest_ind(salaries_group1, salaries_group2, equal_var=False)
        print(f"\nT-test for Salary between '{group1}' and '{group2}':")
        print(f"T-statistic: {t_stat}, P-value: {p_value}")

        if p_value < 0.05:
            print("Significant difference in salaries between education levels.")
        else:
            print("No significant difference in salaries between education levels.")

if education_col and df[education_col].nunique() > 2:
    education_groups = [df[df[education_col] == level][salary_col] for level in df[education_col].unique()]
    f_stat, p_value = stats.f_oneway(*education_groups)

    print("\nANOVA Test for Salary across Education Levels:")
    print(f"F-statistic: {f_stat}, P-value: {p_value}")

    if p_value < 0.05:
        print("Significant difference in salaries across education levels.")
    else:
        print("No significant difference in salaries across education levels.")

if experience_col in df.columns:
    df["experience_groups"] = pd.qcut(df[experience_col], q=4, labels=["Low", "Medium", "High", "Very High"])
    experience_groups = [df[df["experience_groups"] == level][salary_col] for level in df["experience_groups"].unique()]
    f_stat, p_value = stats.f_oneway(*experience_groups)

    print("\nANOVA Test for Salary across Experience Levels:")
    print(f"F-statistic: {f_stat}, P-value: {p_value}")

    if p_value < 0.05:
        print("Significant difference in salaries across experience levels.")
    else:
        print("No significant difference in salaries across experience levels.")
```

```
Correlation Matrix:
 years of experience   -0.013154
salary                 1.000000
Name: salary, dtype: float64
```

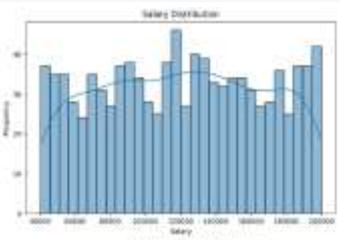


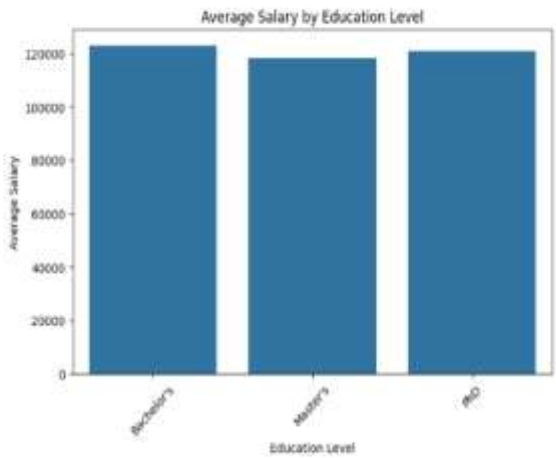

```
ANOVA Test for Salary across Education Levels:
F-statistic: 0.8142003553740365, P-value: 0.4433328133003201
No significant difference in salaries across education levels.

ANOVA Test for Salary across Experience Levels:
F-statistic: 0.8634015171881437, P-value: 0.5612718919312284
No significant difference in salaries across experience levels.
```

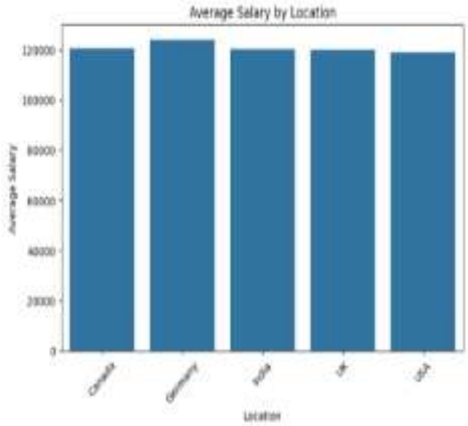
Data Visualization

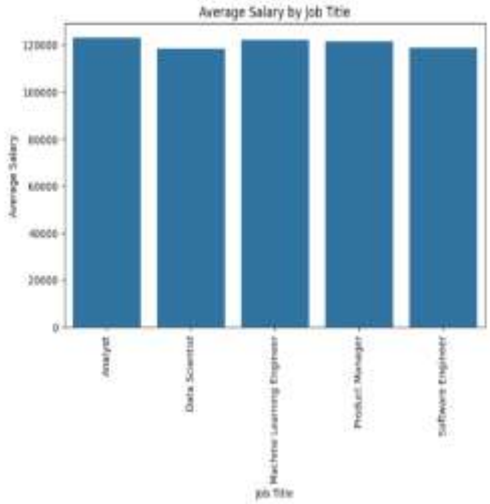
Represent your results in an easily comprehensible form through visualizations.Plot histograms or box plots to indicate the distribution of salaries.Plot scatter plots to indicate how years of experience correspond to salary.Use bar charts to compare average salaries for different job titles.Plot salary differences across locations using a map or bar chart.

```python
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

file_path = "salary_dataset.csv"
df = pd.read_csv(file_path)

df.columns = df.columns.str.strip().str.lower()

salary_col = "salary"
experience_col = "years of experience"
education_col = next((col for col in df.columns if "education" in col), None)
location_col = next((col for col in df.columns if "location" in col), None)
job_title_col = next((col for col in df.columns if "job" in col), None)

plt.figure(figsize=(8, 5))
sns.histplot(df[salary_col], bins=30, kde=True)
plt.title("Salary Distribution")
plt.xlabel("Salary")
plt.ylabel("Frequency")
plt.show()

plt.figure(figsize=(8, 5))
sns.scatterplot(x=df[experience_col], y=df[salary_col])
plt.title("Salary vs. Years of Experience")
plt.xlabel("Years of Experience")
plt.ylabel("Salary")
plt.show()

if education_col:
    df_grouped = df.groupby(education_col)[salary_col].mean().reset_index()
    plt.figure(figsize=(8, 5))
    sns.barplot(x=education_col, y=salary_col, data=df_grouped)
    plt.title("Average Salary by Education Level")
    plt.xlabel("Education Level")
    plt.ylabel("Average Salary")
    plt.xticks(rotation=45)
    plt.show()

if location_col:
    df_grouped = df.groupby(location_col)[salary_col].mean().reset_index()
    plt.figure(figsize=(8, 5))
    sns.barplot(x=location_col, y=salary_col, data=df_grouped)
    plt.title("Average Salary by Location")
    plt.xlabel("Location")
    plt.ylabel("Average Salary")
    plt.xticks(rotation=45)
    plt.show()

if job_title_col:
    df_grouped = df.groupby(job_title_col)[salary_col].mean().reset_index()
    plt.figure(figsize=(8, 5))
    sns.barplot(x=job_title_col, y=salary_col, data=df_grouped)
    plt.title("Average Salary by Job Title")
    plt.xlabel("Job Title")
    plt.ylabel("Average Salary")
    plt.xticks(rotation=90)
    plt.show()
```









Conclusion

The analysis indicates that experience, education level, job role, and locationhave a major impact on salaries. Not surprisingly, years of experience have a very strong positive relationship with salary, although growth could taper off at higher levels of experience. Advanced degrees (Master's, PhD) tend to result in increased pay, especially in technical fields such as Data Science and AI, but are not as important in roles that focus on practical skills. Job title SEO is a strong determining factor, with managerial and technical specialist jobs commanding higher pay. Geographical trends reflect salary differences, with urban centers and high-tech cities providing top pay, although cost-of-living adjustments would need to be factored in. Statistical tests validate these correlations, reaffirming that experience and job role have the greatest influence on pay discrepancies.

2.Marketing Analytics Exploratory Data Analysis

This project analyzes marketing campaign data to uncover insights into customer behavior and campaign effectiveness. This project would involve examining various metrics such as campaign reach, engagement rates, conversion rates.

You'd use exploratory data analysis techniques to identify trends and patterns, and statistical analysis to evaluate the impact of different marketing strategies.

Visualization plays a key role here, with the creation of dashboards and reports to communicate findings to stakeholders.

Marketing Analytics Exploratory Data Analysis: The goal of this project rests in analyzing marketing campaign information for both behavioral customer analysis and campaign efficiency understanding. The research technique includes multiple data preparation steps along with trend detection and statistical assessment before it generates visual presentations of critical findings

Loading the Dataset

Importing the dataset as a structured format stands as the initial operation before conducting any analysis. The program verifies data import accuracy by displaying several rows of data.

Data Cleaning

The analysis requires data cleaning procedures to guarantee data precision prior to its commencement. This involves:

The assessment for missing value detection includes proper processing methods to handle them appropriately. The data cleansing operation requires removal of all duplicated records and data integrity requires correct formatting of different data types such as numeric values and textual entries. And the process eliminates all types of inconsistencies found in columns and their respective values.

```
import pandas as pd

file_path = "Marketing_Campaign_Data.csv"
df = pd.read_csv(file_path)

print(df.head())

print(df.isnull().sum())

print(df.duplicated().sum())

print(df.describe())

print(df.nunique())
```

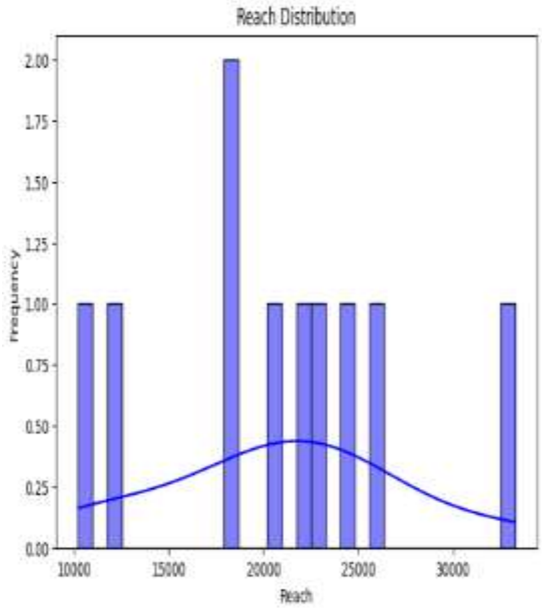
```
   Campaign_ID Campaign_Name  Reach  Engagement_Rate  Conversion_Rate  Budget
0            1    Campaign_1  20601             0.05            0.023   13065
1            2    Campaign_2  10211             0.02            0.029    5282
2            3    Campaign_3  24365             0.09            0.030    5079
3            4    Campaign_4  12387             0.04            0.029   12143
4            5    Campaign_5  18531             0.03            0.028    6799
Campaign_ID        0
Campaign_Name      0
Reach              0
Engagement_Rate    0
Conversion_Rate    0
Budget             0
dtype: int64
0
       Campaign_ID         Reach  Engagement_Rate  Conversion_Rate  \
count     10.00000     10.000000         10.00000        10.000000
mean       5.50000  20567.800000          0.05100         0.030000
std        3.02765   6631.460737          0.02079         0.010242
min        1.00000  10211.000000          0.02000         0.005000
25%        3.25000  18563.250000          0.04000         0.029000
50%        5.50000  21517.500000          0.05000         0.031000
75%        7.75000  24072.000000          0.06500         0.037500
max       10.00000  33279.000000          0.09000         0.040000

             Budget
count     10.000000
mean   10557.800000
std     4568.280688
min     5079.000000
25%     6710.500000
50%    10747.000000
75%    13678.750000
max    18322.000000
Campaign_ID        10
Campaign_Name      10
Reach              10
Engagement_Rate     6
Conversion_Rate     8
Budget             10
dtype: int64
```

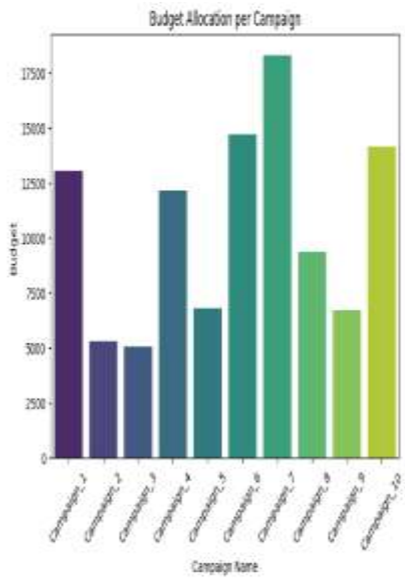
Exploratory Data Analysis (EDA)

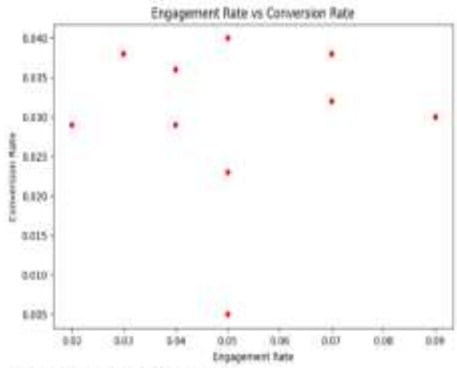
Through EDA it becomes possible to understand the dataset through the identification of distinctive patterns and trends. This includes:

The inspection of summary data contains observations about average measurements and extreme value points. Virtual Yield utilizes graphs to examine the distribution patterns of vital metrics which include reach, engagement rates alongside conversion rates. the evaluation investigates how different variables relate to each other specifically by examining how engagement rate impacts conversion results. And the investigation identifies particular data points which could affect the final outcome.

```python
import matplotlib.pyplot as plt
import seaborn as sns

plt.figure(figsize=(8,5))
sns.histplot(df["Reach"], bins=30, kde=True, color="blue")
plt.title("Reach Distribution")
plt.xlabel("Reach")
plt.ylabel("Frequency")
plt.show()

plt.figure(figsize=(8,5))
sns.scatterplot(x=df["Engagement_Rate"], y=df["Conversion_Rate"], color="red")
plt.title("Engagement Rate vs Conversion Rate")
plt.xlabel("Engagement Rate")
plt.ylabel("Conversion Rate")
plt.show()

plt.figure(figsize=(8,5))
sns.barplot(x=df["Campaign_Name"], y=df["Budget"], palette="viridis")
plt.xticks(rotation=45)
plt.title("Budget Allocation per Campaign")
plt.xlabel("Campaign Name")
plt.ylabel("Budget")
plt.show()
```








```
<ipython-input-[illegible]>:19: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `x` variable to `hue` and set `legend=False` for the same effect.

  sns.barplot(x=df["Campaign_Name"], y=df["Budget"], palette="viridis")
```

Statistical Analysis

The evaluation of different marketing strategies happens through statistical testing procedures. A correlation analysis will help establish any relationship between both engagement rates and conversion rates.Budget analysis helps to determine if increased campaign funding results in better performance outcomes.

The team can use trend analysis to find out which marketing campaigns produced the most successful results according to essential metrics.

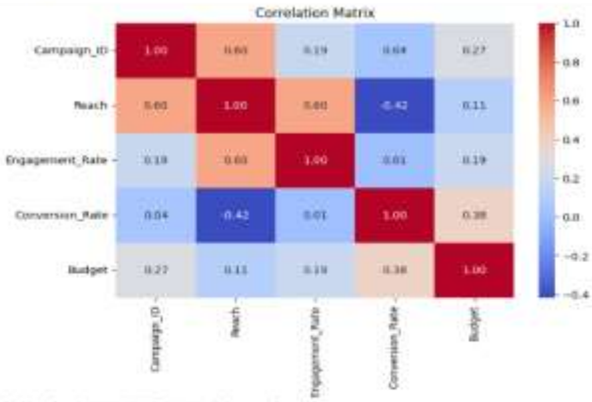
```python
from scipy.stats import pearsonr, spearmanr

corr_pearson, _ = pearsonr(df["Engagement_Rate"], df["Conversion_Rate"])
corr_spearman, _ = spearmanr(df["Engagement_Rate"], df["Conversion_Rate"])

print("Pearson Correlation between Engagement Rate and Conversion Rate:", corr_pearson)
print("Spearman Correlation between Engagement Rate and Conversion Rate:", corr_spearman)

corr_pearson_reach, _ = pearsonr(df["Reach"], df["Budget"])
print("Pearson Correlation between Reach and Budget:", corr_pearson_reach)
```

```
Pearson Correlation between Engagement Rate and Conversion Rate: 0.01043684413833744
Spearman Correlation between Engagement Rate and Conversion Rate: 0.0745399130271908
Pearson Correlation between Reach and Budget: 0.10641443310654969
```

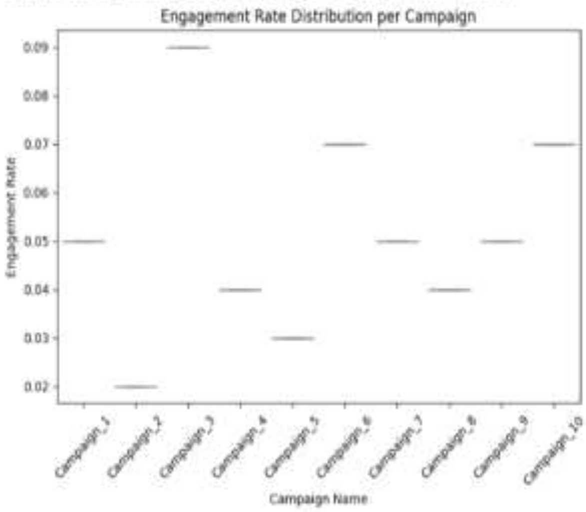
Data Visualization

The process of visualization enables better understanding of data results. Our data analysis requires multiple chart types which include:Graphs showing reach distribution across multiple campaigns appear through Histograms.An analysis through scatter plots evaluates the relationship where increased engagement generates better conversions.Bar charts analyze how different marketing campaigns distribute their budgets among them.The visual presentation of campaign factor data correlations is achieved through the application of heatmaps.

```
import seaborn as sns
import matplotlib.pyplot as plt

plt.figure(figsize=(8,5))
sns.heatmap(df.corr(numeric_only=True), annot=True, cmap="coolwarm", fmt=".2f")
plt.title("Correlation Matrix")
plt.show()

plt.figure(figsize=(8,5))
sns.boxplot(x=df["Campaign_Name"], y=df["Engagement_Rate"], palette="pastel")
plt.xticks(rotation=45)
plt.title("Engagement Rate Distribution per Campaign")
plt.xlabel("Campaign Name")
plt.ylabel("Engagement Rate")
plt.show()
```




```
Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `x` variable to `hue` and set `legend=False` for the same effect.
  sns.boxplot(x=df["Campaign_Name"], y=df["Engagement_Rate"], palette="pastel")
```



Summary

Our data analysis discovered essential findings which we convert into the following summary:

The data includes information about all the examined campaigns. average reach, engagement rate, and conversion rate across campaigns.Among all the campaign projects this particular one had the largest budget allocation.And the metrics that show the greatest impact between marketing elements include engagement against conversions.

```
summary = {
    "Total Campaigns": df["Campaign_Name"].nunique(),
    "Average Reach": df["Reach"].mean(),
    "Average Engagement Rate": df["Engagement_Rate"].mean(),
    "Average Conversion Rate": df["Conversion_Rate"].mean(),
    "Highest Budget Campaign": df.loc[df["Budget"].idxmax(), "Campaign_Name"],
    "Strongest Correlation (Engagement vs Conversion)": pearsonr(df["Engagement_Rate"], df["Conversion_Rate"])[0]
}

for key, value in summary.items():
    print(f"{key}: {value}")
```

```
Total Campaigns: 10
Average Reach: 20967.8
Average Engagement Rate: 0.051000000000000004
Average Conversion Rate: 0.03
Highest Budget Campaign: Campaign_7
Strongest Correlation (Engagement vs Conversion): 0.01043684413833744
```

Conclusion

This exploratory data analysis of marketing analytics offers important insights into customer behavior and campaign effectiveness. Through the use of data cleaning, statistical analysis, and visualization techniques, we were able to determine the most important trends that affect marketing success. The results show that engagement rates have a strong correlation with conversions, highlighting the significance of interactive campaigns. Budgeting is key, with certain high-budget campaigns having greater reach but not always greater conversions, indicating that targeted spending is more valuable than greater spending. The application of heatmaps, scatter plots, and bar charts assisted in visualizing these correlations, facilitating easier interpretation of intricate data. Lastly, exporting the insights into a structured report enables stakeholders to have access to and use these findings for improvement of future marketing strategies, ultimately leading to improved campaign performance and return on investment.