DTSC 5502

Project Title:

Forecasting Sales and Visualizing Ad Campaign Impact on Social Media Platforms using Machine Learning

Team Members:

Name	Email_Id	Student_Id
JHANSI MUPPALLA	jhansimuppalla@my.unt.edu	11735545
TAZBEENA SHAIK	tazbeenashaik@my.unt.edu	11683696
SPOORTHI HASSAN SATHYANARAYANA	spoorthihassansathyanarayana@my.unt.edu	11814524
YOGI VENKATA BALAJI MUTHAKANI	yogivenkatabalajimuthakani@my.unt.edu	11801024

Abstract

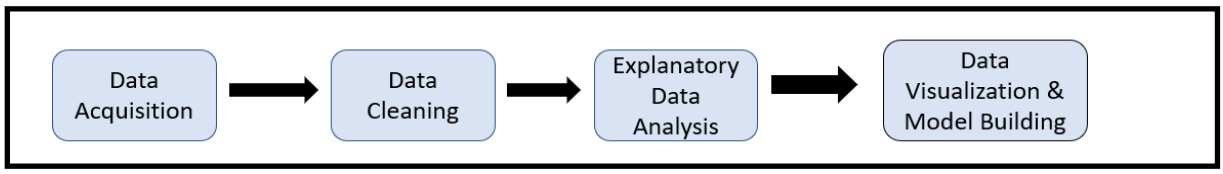
The main focus nalyses social media ad marketing data which helps in future sales optimization and predictions thereby enhancing the ad conversions. Key patterns are identified using the python libraries such as Matplotlib and seaborn from the user demographics, and performance metrics. Final motive of the project is to improve the Return on Investment (ROI), enhancing the performance of ads and outering insights for the campaign. Advanced data visualization techniques are used to create dashboards and bring insights for the marketing teams to assess key performance indicators, click through rates and ad conversion rates.

Our Project comes under the domain of **Digital Marketing.** It is a broad domain and deals with use of digital channels such as Search engines, Social Media, Email marketing and more. This domain majorly relies on a 3-step process. The steps are

I. **Targeting** : Where we first look who is the customer base we want to target for the product. This could be done by a variety of factors such as age, interest, demographics, gender and more.
II. **Reaching Out** : Process of reaching the Target audience and the tools required or means of communication we use to reach the defined Target audience
III. **Conversion Optimization** : Once we are done with reaching out, Now we need to optimise the user experience of the user in such a way that the ad we sent should convert into a Sale.

By following the above defined process, Digital marketers can turn ads into sales and drive real valuable results to their business.

WorkFlow Diagram –



In this project, first we acquire the required data and then do the basic data cleaning and then do the Exploratory Data analysis and find the insights and the correlations in the dataset and then using the Matplotlib library of the python we have done the basic data visualisations and the we used Machine Learning algorithms to train the model.

Related Work –

The background work of the article with title [5] Social Media Advertisement and Its Effects in Sales Prediction- An Analysis by subha explores the effect of social

media advertising on predicting sales. The primary goal of this paper is to explore the impact of social media advertising as a marketing tactic and how it affects forecasting sales. It is important for us to keep in mind that social media has become a more and more popular medium for businesses for marketing their products and services if we want to understand the study's background. Companies can target specific age groups and reach many people at a low cost by using social media advertising. Social media platforms can give companies useful information about consumer behaviour and preferences, which can be used to estimate sales and improve marketing plans. In this study, the author studies data from a few social media platforms to look at the effect of social media advertising on prediction of sales. The article gives details on social media advertising's efficiency and its potential to affect sales forecasting. Businesses must comprehend the link between social media advertising and sales forecasting in order to create effective promotional strategies and maximise their return on investment.

The background work of the article with title [6] (When) Can Social Media Buzz Data Replace Traditional Surveys for Sales Forecasting?" by Shi, Yuying, Karniouchina, Explores the effect of social media buzz as an alternative option for traditional surveys in sales forecasting. Nowadays social media has become a popular source for sales forecasting. The word social media buzz data means how much that product or that services interact in social media platforms. The data can be analysed to predict user behaviour on sales trends of products. Previously the traditional methods of sales forecasting used are gathering customer information and their preferences, but this type of methods are time consumption and the data collected is not accurate. This article determines that this social media survey is an alternative for traditional surveys in order to estimate sales. In this article the author

compares accuracy of sales based on social media and traditional survey and he concluded that social media sales forecasting is best while doing sales prediction.

• Data Abstraction

Dataset (Type and Attributes):

For this project, we have downloaded the dataset from Kaggle. This dataset is taken from Facebook's social media ad campaign.

This dataset is a CSV file. A table format with 11 columns and 1143 data points for each attribute.

The dataset consists of the following columns,

1.) Ad_id: Each ad has a special ID.

2.) xyzcampaignid: an ID linked to every advertising campaign run by XYZ firm.

3.) fbcampaignid, a unique identifier used by Facebook to track each campaign.

4.) age: The age of the viewer of the advertisement.

5.) Gender: The gender of the viewer of the advertisement

6.)Interest: A code that identifies the group to which a person's interests belong (interests are listed on a person's public Facebook profile).

7.)Impressions: the quantity of times the advertisement was viewed.

8.) Clicks: The quantity of times that ad was clicked.

9.) Spent: The sum that company xyz paid Facebook to display that advertisement.

10.) Total conversion: The total amount of individuals that contacted the company about the product after viewing the ad

We can also look at a part of the dataset below.

```
[2] from google.colab import files
    uploaded = files.upload()
```

Choose Files ad_campai...analysis.csv

- **ad_campaign_analysis.csv**(text/csv) - 60522 bytes, last modified: 9/29/2024 - 100% done

Saving ad_campaign_analysis.csv to ad_campaign_analysis.csv

```
[3] data_frame=pd.read_csv("ad_campaign_analysis.csv")
```

Printing the first 5 columns using data.head()

```
[4] data_frame.head()
```

	ad_id	xyz_campaign_id	fb_campaign_id	age	gender	interest	Impressions	Clicks	Spent	Total_Conversion	Approved_Conversion
0	708746	916	103916	30-34	M	15	7350	1	1.43	2	1
1	708749	916	103917	30-34	M	16	17861	2	1.82	2	0
2	708771	916	103920	30-34	M	20	693	0	0.00	1	0
3	708815	916	103928	30-34	M	28	4259	1	1.25	1	0
4	708818	916	103928	30-34	M	28	4133	1	1.29	1	1

We have also made sure there are no null values in the dataset.

Checking for null values

```
[ ] df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1143 entries, 0 to 1142
Data columns (total 11 columns):
 #   Column               Non-Null Count  Dtype
---  ------               --------------  -----
 0   ad_id                1143 non-null   int64
 1   xyz_campaign_id      1143 non-null   int64
 2   fb_campaign_id       1143 non-null   int64
 3   age                  1143 non-null   object
 4   gender               1143 non-null   object
 5   interest             1143 non-null   int64
 6   Impressions          1143 non-null   int64
 7   Clicks               1143 non-null   int64
 8   Spent                1143 non-null   float64
 9   Total_Conversion     1143 non-null   int64
 10  Approved_Conversion  1143 non-null   int64
dtypes: float64(1), int64(8), object(2)
memory usage: 98.4+ KB
```

We can have a look at the shape and description of the dataset.

Exploratory Data Analysis

```
[ ] df.shape
```

```
(1143, 11)
```

```
[ ] df.describe()
```

	ad_id	xyz_campaign_id	fb_campaign_id	interest	Impressions	Clicks	Spent	Total_Conversion	Approved_Conversion
count	1.143000e+03	1143.000000	1143.000000	1143.000000	1.143000e+03	1143.000000	1143.000000	1143.000000	1143.000000
mean	9.872611e+05	1067.382327	133783.989501	32.766404	1.867321e+05	33.390201	51.360656	2.855643	0.944007
std	1.939928e+05	121.629393	20500.308622	26.952131	3.127622e+05	56.892438	86.908418	4.483593	1.737708
min	7.087460e+05	916.000000	103916.000000	2.000000	8.700000e+01	0.000000	0.000000	0.000000	0.000000
25%	7.776325e+05	936.000000	115716.000000	16.000000	6.503500e+03	1.000000	1.480000	1.000000	0.000000
50%	1.121185e+06	1178.000000	144549.000000	25.000000	5.150900e+04	8.000000	12.370000	1.000000	1.000000
75%	1.121804e+06	1178.000000	144657.500000	31.000000	2.217690e+05	37.500000	60.025000	3.000000	1.000000
max	1.314415e+06	1178.000000	179982.000000	114.000000	3.052003e+06	421.000000	639.949998	60.000000	21.000000

Data Transformation

Modelling

Replacing xyz_campaign_ids again with actual ids for modelling

```
[ ] df["xyz_campaign_id"].replace({"campaign_a":916 ,"campaign_b":936 ,"campaign_c":1178}, inplace=True)
```

Encoding the Labels 'gender' and 'age' for better modelling

```
[ ] #encoding gender
    from sklearn.preprocessing import LabelEncoder
    encoder=LabelEncoder()
    encoder.fit(df["gender"])
    df["gender"]=encoder.transform(df["gender"])
    print(df["gender"])
```

```
0       1
1       1
2       1
3       1
4       1
       ..
1138    0
1139    0
1140    0
```

```
Name: gender, Length: 1143, dtype: int64
```

```
[ ] #encoding age
    encoder.fit(df["age"])
    df["age"]=encoder.transform(df["age"])
    print(df["age"])
```

```
0       0
1       0
2       0
3       0
4       0
       ..
1138    3
1139    3
1140    3
1141    3
1142    3
Name: age, Length: 1143, dtype: int64
```

Along with the above data transformations, we have also encoded the campaign names below.

• Task Abstraction:

The main task of this project is to perform analysis of Facebook ad campaign data and optimise sales conversion. We then use these campaign results to predict future sales. The objective is to analyse the factors that influence sales such as ad impressions, number of clicks and cost for each click etc and create a predictive model that can be applied to optimise future ad campaigns. The project involves tasks like loading the dataset, performing preprocessing, exploratory data analysis and developing machine learning models for sales prediction and then deploying the web application.

Task (Target and Actions):

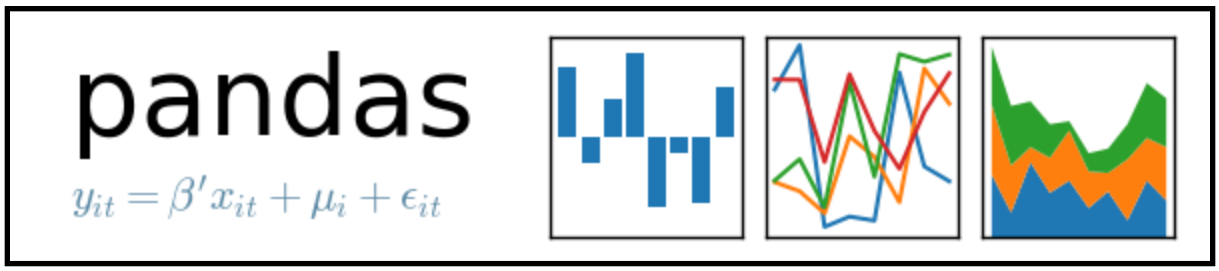
The target of this project is to perform sales prediction based on the results of analysis of Facebook ad campaign dataset.
This task involves the following actions:

- Loading and Preprocessing the dataset
- Performing exploratory data analysis
- Training machine learning models for predicting sales
- Performance evaluation of models using different metrics
- Enhance the model based on the obtained results and gain insights that lead to sales prediction

• Implementation using tools

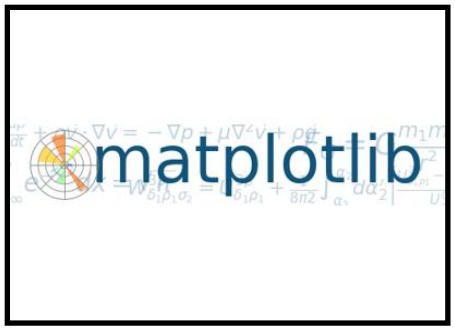
Loading dataset and Exploratory data analysis – Pandas library

- We have used the pandas library for loading the dataset and performing exploratory data analysis. Pandas is a popular python library which is commonly used for data analysis and manipulation. It is also used for tasks like data cleaning and data preprocessing. There are many tools and functions offered by pandas for data manipulation such as dataframes and series. It is used by many users due to its high performance and productivity.



Data visualisation – Matplotlib and Seaborn

- We have used Matplotlib and Seaborn libraries for the purpose of data visualisation. Matplotlib is a popular python library used for creating different kinds of charts, plots and graphs. It offers various visualisation tools making it easier to present the data to the users.
- Seaborn is also a popular data visualisation library built on top of matplotlib. It offers a high level interface for producing attractive and advanced visualisations. It can also be utilised for data exploration. Seaborn offers various functions for visualising different types of data like distribution plots, matrix plots and regression plots.



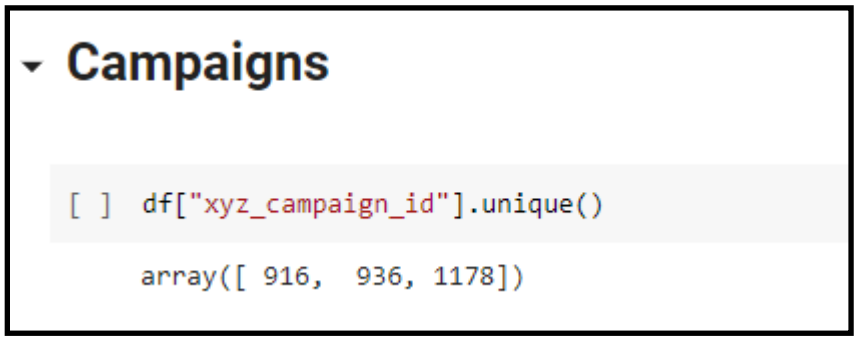
Machine learning modelling – Scikit-learn

- We have used the scikit-learn library for training the machine learning models to predict sales. Scikit-learn is a machine learning library which offers various tools for machine learning tasks such as clustering, classification and regression. It is built on top of Scipy, Numpy and Matplotlib. It also provides tools for data preprocessing, model selection and feature selection.

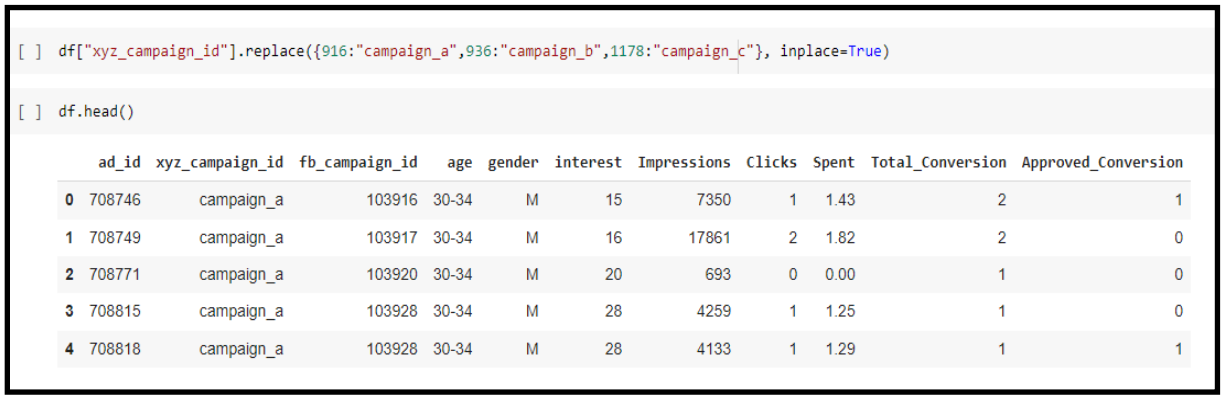
• Preliminary Results for Analysis

In this part of the project we majorly focus on performing Exploratory Data Analysis on the taken dataset. We have used the above mentioned tools to get the insights from the data.

Using the Pandas library and Matplotlib we have generated the following maps. Firstly we analysed the no. of campaigns.



We can observe that the xyz corporation has three separate advertising campaigns here. For better visualisation, we'll now change their names to a,b,c.



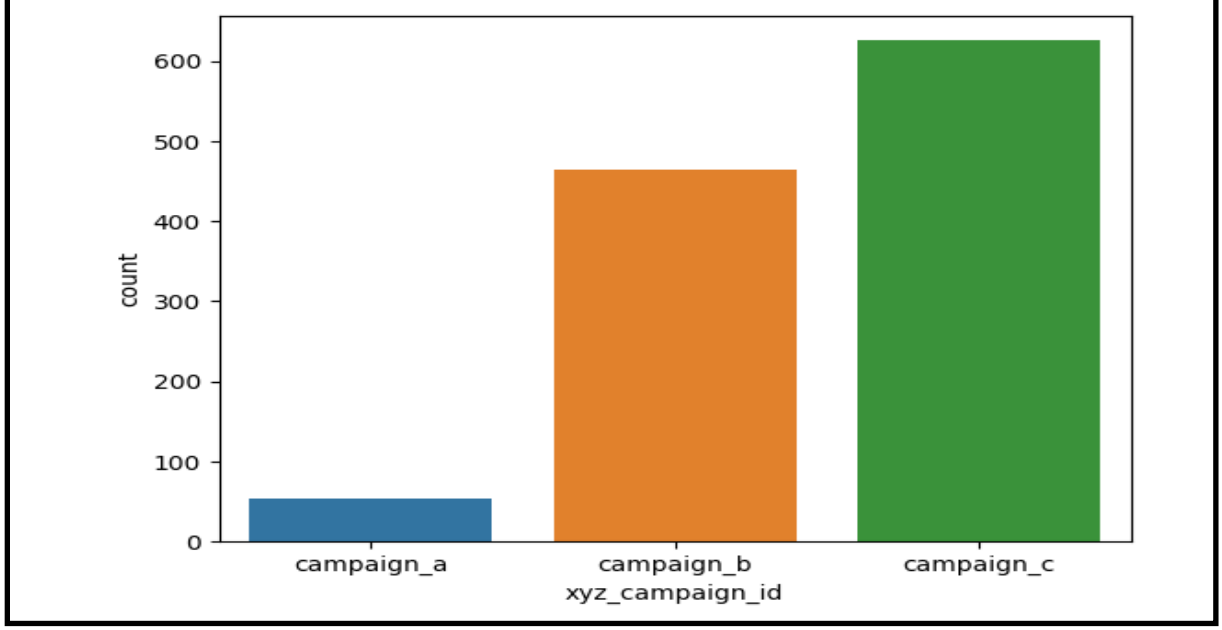
```
[ ] df["xyz_campaign_id"].replace({916:"campaign_a",936:"campaign_b",1178:"campaign_c"}, inplace=True)

[ ] df.head()
```

	ad_id	xyz_campaign_id	fb_campaign_id	age	gender	interest	Impressions	Clicks	Spent	Total_Conversion	Approved_Conversion
0	708746	campaign_a	103916	30-34	M	15	7350	1	1.43	2	1
1	708749	campaign_a	103917	30-34	M	16	17861	2	1.82	2	0
2	708771	campaign_a	103920	30-34	M	20	693	0	0.00	1	0
3	708815	campaign_a	103928	30-34	M	28	4259	1	1.25	1	0
4	708818	campaign_a	103928	30-34	M	28	4133	1	1.29	1	1

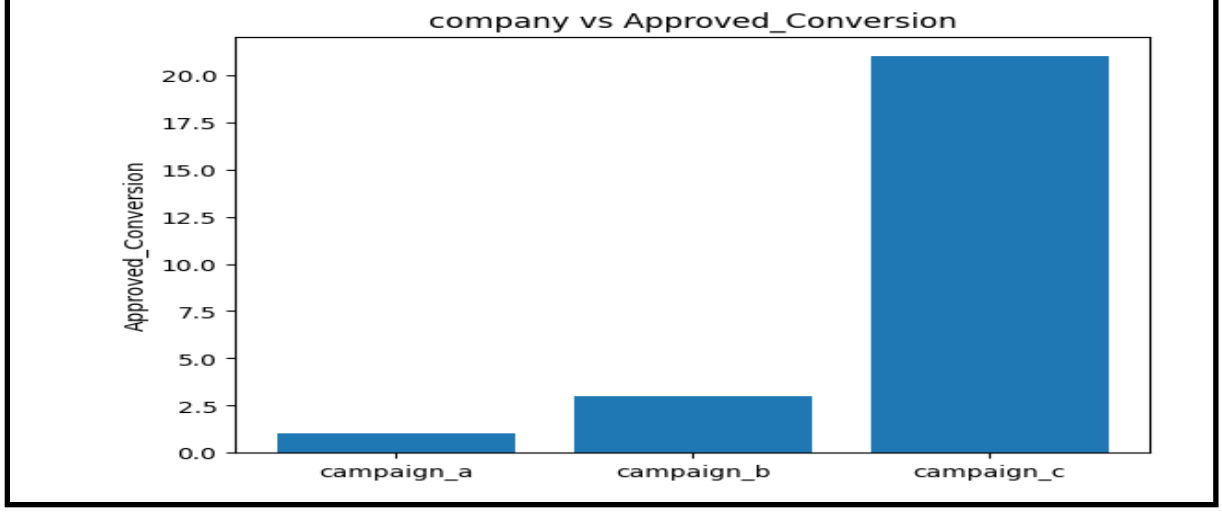
This is how the rendered dataframe looks after we replace the campaign names.

```
[ ]  # count plot on single categorical variable
     sns.countplot(x ='xyz_campaign_id', data = df)
     # Show the plot
     plt.show()
```



A simple count plot to plot the no of ad_campaings per each campaign. From the above plot we can infer that Campaign_c has most no_of ads.

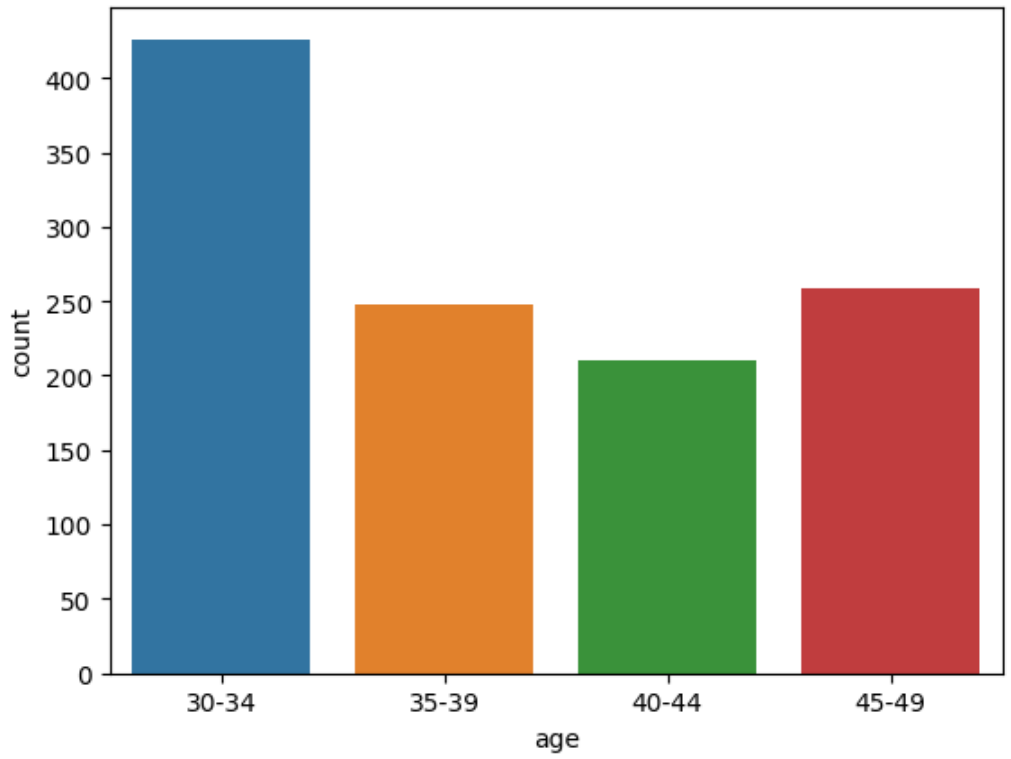
```
[ ]  #Approved_Conversion
     # Creating our bar plot
     plt.bar(df["xyz_campaign_id"], df["Approved_Conversion"])
     plt.ylabel("Approved_Conversion")
     plt.title("company vs Approved_Conversion")
     plt.show()
```



We tried to plot a countplot for no. Approved_conversion in each campaign. From the above plot we can deduce that Campaign_c has the best conversion rate i.e, most products were bought in campaign-c.

Age

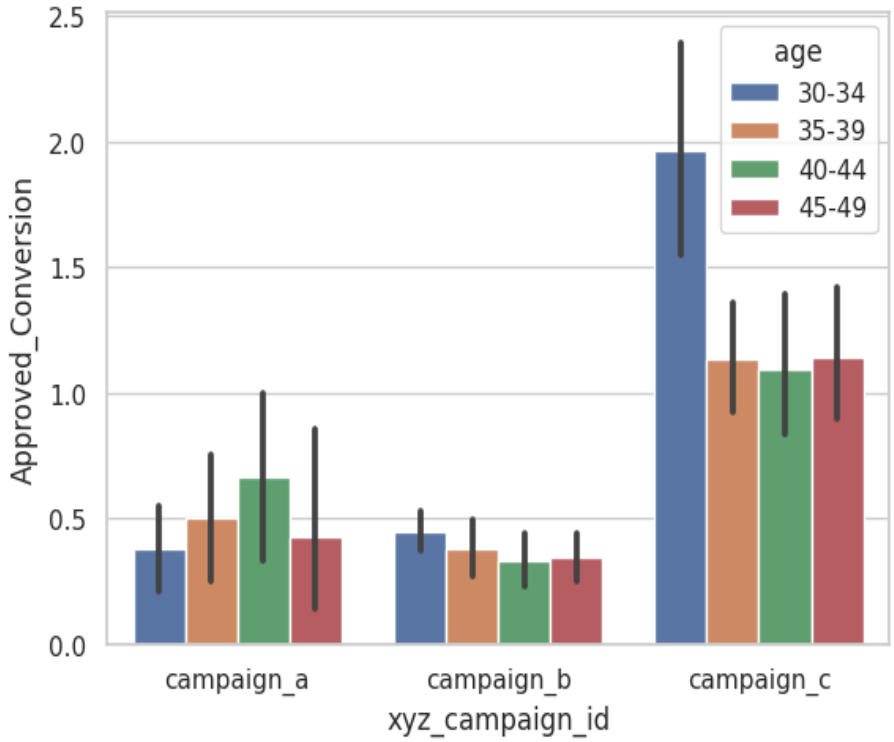
```
[ ]  # count plot on single categorical variable
     sns.countplot(x ='age', data = df)
     # Show the plot
     plt.show()
```



We tried to plot the most targeted age-group from all the campaigns. We can conclude from the above countplot that the age group 30-34 is the most target age-group.

```
import seaborn as sns
sns.set(style="whitegrid")
tips = sns.load_dataset("tips")
sns.barplot(x=df["xyz_campaign_id"], y=df["Approved_Conversion"], hue=df["age"], data=tips)
```

```
<Axes: xlabel='xyz_campaign_id', ylabel='Approved_Conversion'>
```

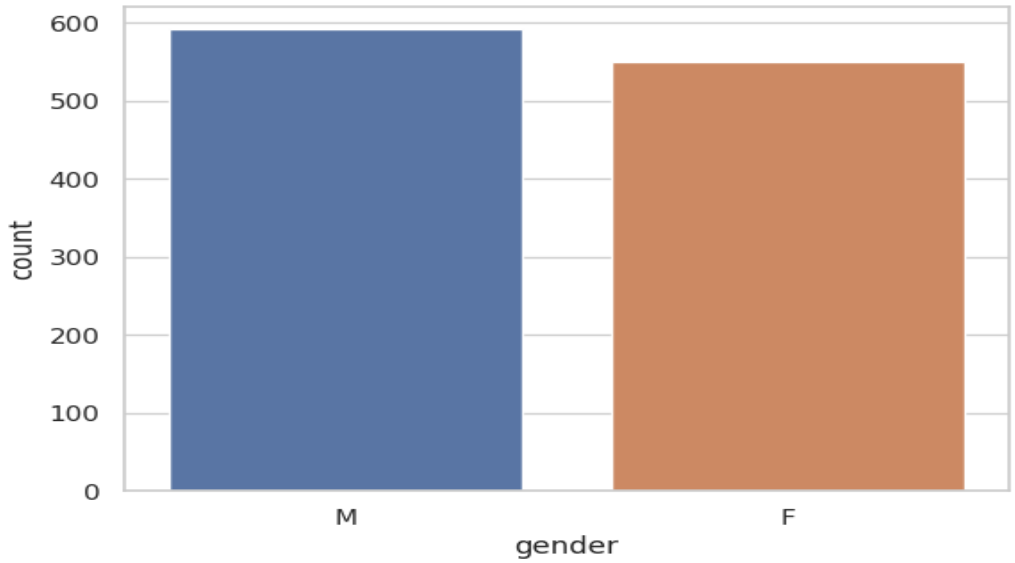


We used seaborn's barplot function to create a grouped bar plot that shows the relationship between campaign ID, approved conversions, and age.

It's interesting to note that in campaign_c and campaign_b, the age group of 30-34 shows more interest, whereas in campaign_a the age group of 40-44 shows more interest.

Gender

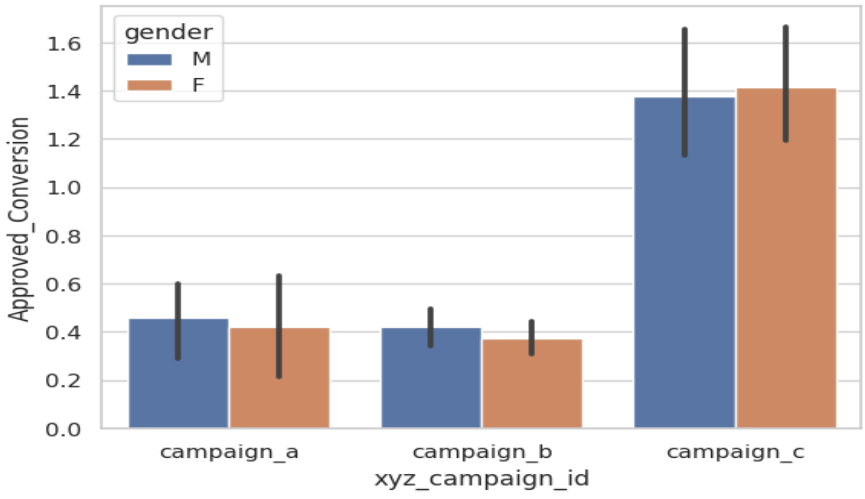
```
[ ]  # count plot on single categorical variable
     sns.countplot(x ='gender', data = df)
     # Show the plot
     plt.show()
```



We tried to create a countplot between the gender and we can infer from the graph that there are more male audiences than the female audience.

```
[ ]  import seaborn as sns
     sns.set(style="whitegrid")
     tips = sns.load_dataset("tips")
     sns.barplot(x=df["xyz_campaign_id"], y=df["Approved_Conversion"], hue=df["gender"], data=tips)
```

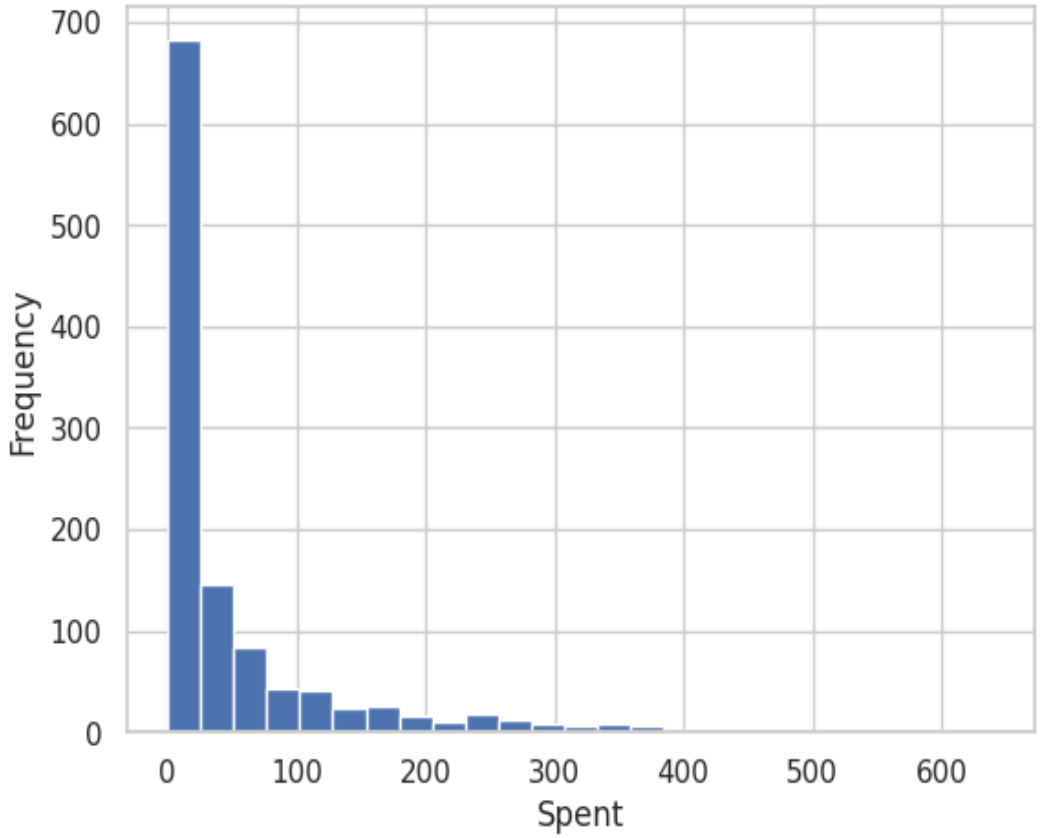
```
<Axes: xlabel='xyz_campaign_id', ylabel='Approved_Conversion'>
```



We created a grouped bar plot that shows the relationship between campaign ID, approved conversions, and gender. This allows us to see the gender distribution along each campaign. Both genders show almost similar interest in each campaign.

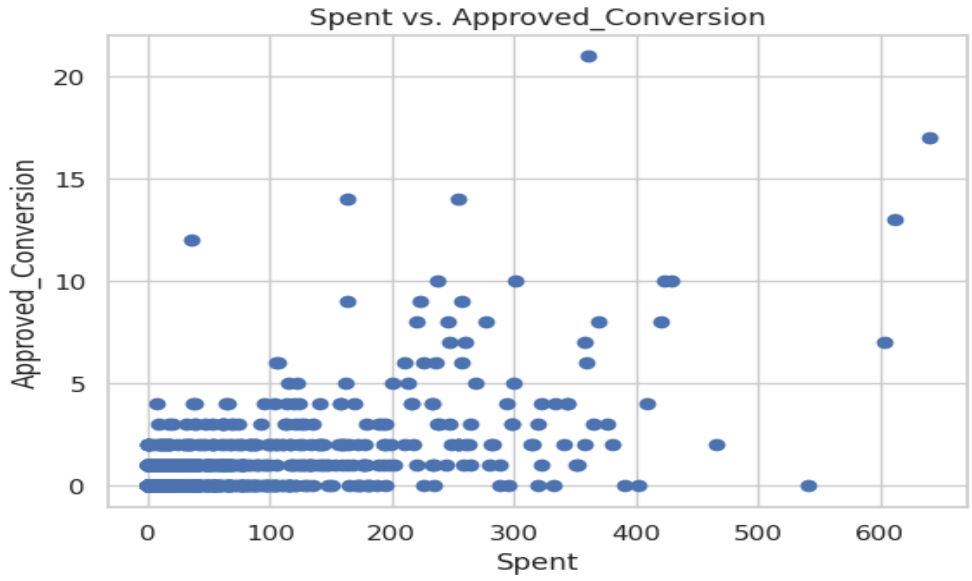
Spent

```
[ ]  plt.hist(df['Spent'], bins = 25)
     plt.xlabel("Spent")
     plt.ylabel("Frequency")
     plt.show()
```



We tried to plot a histogram, to see the number of data points in the campaigns spent at each spend rate.

```
# No. of products bought vs the acquistion cost
plt.scatter(df["Spent"], df["Approved_Conversion"])
plt.title("Spent vs. Approved_Conversion")
plt.xlabel("Spent")
plt.ylabel("Approved_Conversion")
plt.show()
```

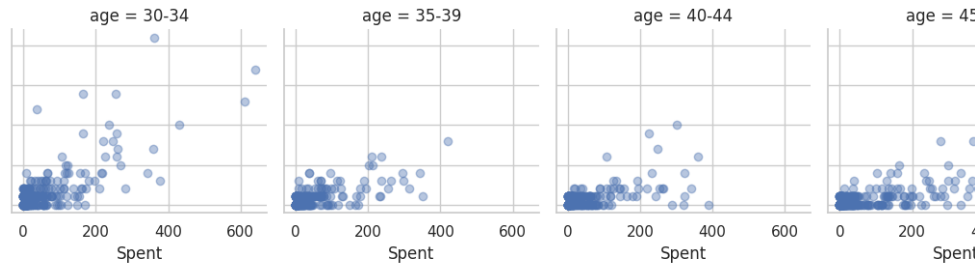


We can see, as the amount of money spent increases, the number of products bought increases.

```
# Create a FacetGrid with age as the column variable
g = sns.FacetGrid(df, col="age")

# Map a scatter plot of Spent versus Approved_Conversion for each age group
g.map(plt.scatter, "Spent", "Approved_Conversion", alpha=.4)

# Add a legend to the plot
g.add_legend()
```



We created a grid of scatter plots(facet grid), with each plot showing the relationship between “Spent” and “Approved_Conversion” for all age groups.

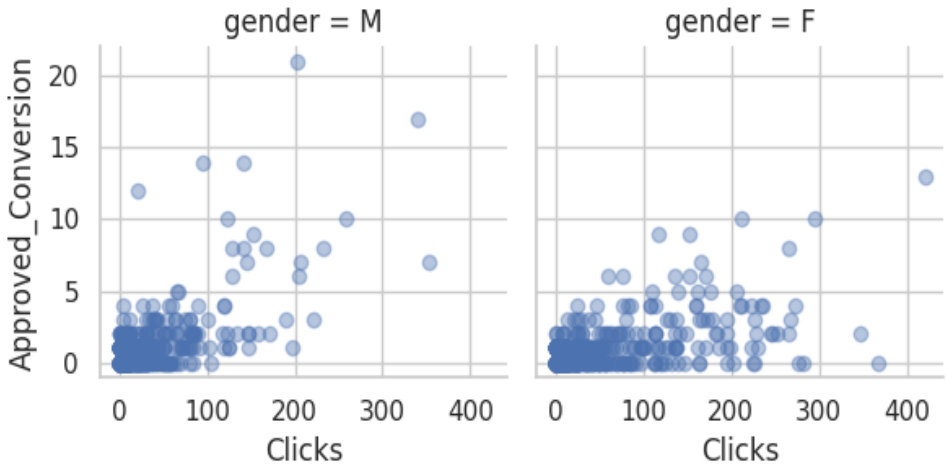
Now moving on to the vital attributes, “People who actually bought the product”.

People who actually bought the product

After Clicking the ad ?

Let's see people who actually went from clicking to buying the product.

```
[ ]  g = sns.FacetGrid(df, col="gender")
     g.map(plt.scatter, "Clicks", "Approved_Conversion", alpha=.4)
     g.add_legend();
```



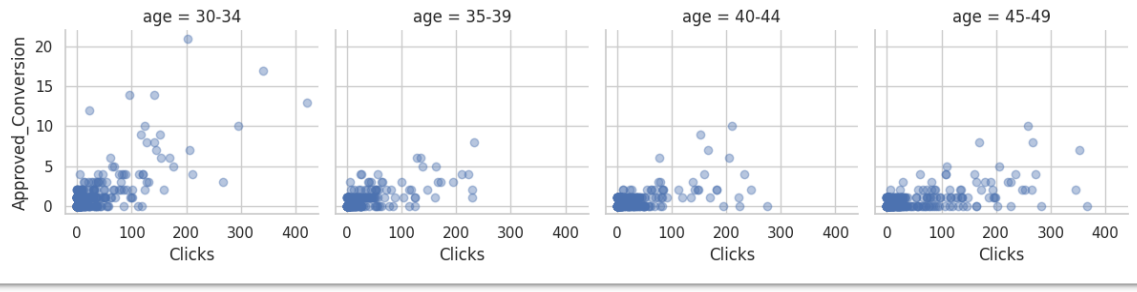
Men appear to click on ads more often than women, but after clicking on the ad, women tend to make more purchases.

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

# Create a FacetGrid with age as the column variable
g = sns.FacetGrid(df, col="age")

# Map a scatter plot of Clicks versus Approved_Conversion for each age group
g.map(plt.scatter, "Clicks", "Approved_Conversion", alpha=.4)

# Add a legend to the plot
g.add_legend()
```



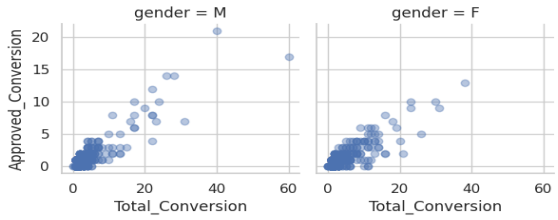
We created a grid of scatter plots, with each plot showing the relationship between “Clicks” and “Approved_Conversion” for a different age group.

People in the 30-34 age group have a greater tendency to buy products after clicking the ad.

After enquiring the product?

Let's see people who actually went from enquiring to buying the product.

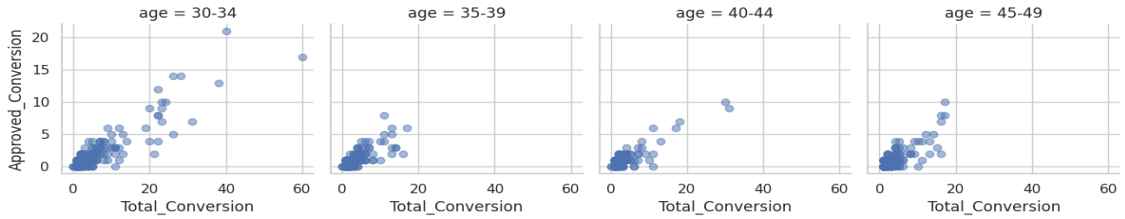
```
[ ]  g = sns.FacetGrid(df, col="gender")
     g.map(plt.scatter, "Total_Conversion", "Approved_Conversion", alpha=.4)
     g.add_legend();
```



It seems women buys more products than men after enquiring the product. However men tends to enquire more about the product.

```
[ ]  g = sns.FacetGrid(df, col="age")
     g.map(plt.scatter, "Total_Conversion", "Approved_Conversion",alpha=.5)
     g.add_legend()
```

```
<seaborn.axisgrid.FacetGrid at 0x7fef34eb5700>
```



It seems people in age group 30-34 are more likely to buy the product after enquiring the product.

Regression model and analysis

Before doing the Regression model we are replacing **xyz_campaign_ids** with actual id's and encoding the labels for age and gender.

Encoding the Labels 'gender' and 'age' for better modelling

0s

```
#encoding gender
from sklearn.preprocessing import LabelEncoder
encoder=LabelEncoder()
encoder.fit(df["gender"])
df["gender"]=encoder.transform(df["gender"])
print(df["gender"])
```

```
#encoding age
encoder.fit(df["age"])
df["age"]=encoder.transform(df["age"])
print(df["age"])
```

Splitting the data into train and test with proportion 80:20

splitting Data into testset and trainset

```
[54] from sklearn.model_selection import train_test_split
     x_train, x_test, y_train, y_test = train_test_split(x,y,test_size=0.2, random_state=42)
```

We are predicting Total conversion by using random forest regression. After the evaluation we got the mean absolute error is 0.99 and 75.3% of data fit into the regression model.

```
[59] mae

     0.9912663755458515
```

The mean absolute error achieved is 0.99.

```
[61] #R-squred value
     r2_score

     0.7530816415210646
```

• Milestones

• Project Management

• Responsibility (Task, Person)

Name	Task Completed
Tazbeena Shaik	Data collection. Data Understanding, Data Preprocessing, ppt
Jhansi Muppala	EDA, Feature engineering, report
Spoorthi Hassan Sathyanarayana	Machine Learning Algorithms, Model selection, model building, ppt
Yogi Venkata Balaji Muthakani	ML models, Visualizations, Results, Report, PPT

Appendix:

```
import numpy as np
import pandas as pd

from google.colab import files
uploaded = files.upload()

data_frame=pd.read_csv("ad_campaign_analysis.csv")
data_frame.head()
```

```
data_frame.info()
data_frame.shape
data_frame.describe()
# Importing the Libararies for visualization
import matplotlib.pyplot as plt
import seaborn as sns
corr=sns.heatmap(data_frame[["Impressions","Clicks","Spent","Total_Conve
rsion","Approved_Conversion"]].corr(),annot=True ,fmt=".2f",
cmap="coolwarm")
data_frame["xyz_campaign_id"].unique()
data_frame["xyz_campaign_id"].replace({916:"CAMPAIGN_A",936:"CAMPAIG
N_B",1178:"CAMPAIGN_C"}, inplace=True)
data_frame.head()

# visualizing count plot on single categorical variable
sns.countplot(x ='xyz_campaign_id', data = data_frame)
# Generating the plot
plt.show()

#Company vs Approved_Conversion
# Generating bar plot
plt.bar(data_frame["xyz_campaign_id"], data_frame["Approved_Conversion"])
plt.ylabel("Approved_Conversion")
plt.title("Company vs Approved_Conversion")
plt.show()
```

```
# count plot on single categorical variable
sns.countplot(x ='age', data = data_frame)
# Generating the plot
plt.show()

import seaborn as sns
sns.set(style="whitegrid")
tips = sns.load_dataset("tips")
sns.barplot(x=data_frame["xyz_campaign_id"],
y=data_frame["Approved_Conversion"], hue=data_frame["age"], data=tips)

# count plot on single categorical variable
sns.countplot(x ='gender', data = data_frame)
# Generating the plot
plt.show()

import seaborn as sns
sns.set(style="whitegrid")
tips = sns.load_dataset("tips")
sns.barplot(x=data_frame["xyz_campaign_id"],
y=data_frame["Approved_Conversion"],                hue=data_frame["gender"],
data=tips)
```

```
# count plot on single categorical variable
fig_dims = (15,6)
fig, ax = plt.subplots(figsize=fig_dims)
sns.countplot(x ='interest', data = data_frame)
# Generating the plot
plt.show()

plt.scatter(data_frame["interest"], data_frame["Approved_Conversion"])
plt.title("interest vs. Approved_Conversion")
plt.xlabel("interest")
plt.ylabel("Approved_Conversion")
plt.show()

p = sns.FacetGrid(data_frame, col="gender")
p.map(plt.scatter, "interest", "Approved_Conversion", alpha=.4)
p.add_legend();

p = sns.FacetGrid(data_frame, col="age")
p.map(plt.scatter, "interest", "Approved_Conversion", alpha=.4)
p.add_legend();

plt.hist(data_frame['Spent'], bins = 25)
plt.xlabel("Spent")
```

```
plt.ylabel("Frequency")
plt.show()
# spent vs approved conversion
plt.scatter(data_frame["Spent"], data_frame["Approved_Conversion"])
plt.title("Spent vs. Approved_Conversion")
plt.xlabel("Spent")
plt.ylabel("Approved_Conversion")
plt.show()

p = sns.FacetGrid(data_frame, col="gender")
p.map(plt.scatter, "Spent", "Approved_Conversion", alpha=.4)
p.add_legend();

# Creating a FacetGrid with the column variable as age
p = sns.FacetGrid(data_frame, col="age")

# mapping Scatter plot for Spent vs Approved Conversion for each age group
p.map(plt.scatter, "Spent", "Approved_Conversion", alpha=.4)

# Giving legend for plot
p.add_legend()


plt.hist(data_frame['Impressions'], bins = 25)
plt.xlabel("Impressions")
```

```
plt.ylabel("Frequency")
plt.show()
plt.scatter(data_frame["Impressions"], data_frame["Approved_Conversion"])
plt.title("Impressions vs. Approved_Conversion")
plt.xlabel("Impressions")
plt.ylabel("Approved_Conversion")
plt.show()

p = sns.FacetGrid(data_frame, col="gender")
p.map(plt.scatter, "Clicks", "Approved_Conversion", alpha=.4)
p.add_legend();

import seaborn as sns
import matplotlib.pyplot as plt

# Creating a FacetGrid with the column variable as age
p = sns.FacetGrid(data_frame, col="age")

# Mapping scatter plot of Clicks vs Approved_Conversion for each age group
p.map(plt.scatter, "Clicks", "Approved_Conversion", alpha=.4)

# Giving legend for plot
p.add_legend()

p = sns.FacetGrid(data_frame, col="gender")
```

```
p.map(plt.scatter, "Total_Conversion", "Approved_Conversion", alpha=.4)
p.add_legend();
p = sns.FacetGrid(data_frame, col="age")
p.map(plt.scatter, "Total_Conversion", "Approved_Conversion",alpha=.5)
p.add_legend()

c_a=[]
c_b=[]
c_c=[]
for i,j,k in zip(data_frame.xyz_campaign_id, data_frame.fb_campaign_id,
data_frame.Approved_Conversion):
  if i=="CAMPAIGN_C":
    c_a.append(i),c_b.append(j),c_c.append(k)

z={'campaign_name':c_a, 'fb_campaign_id':c_b, 'Approved_Conversion':c_c}
c_campaign=pd.DataFrame(z)
c_campaign.head()

plt.figure(figsize=(20,5))
plt.scatter(c_campaign["fb_campaign_id"],
c_campaign["Approved_Conversion"])
plt.title("fb campaign id vs Approved Conversion for CAMPAIGN_C")
plt.xlabel("fb campaign id")
plt.ylabel("Approved Conversion")
plt.show()
```

```
data_frame["xyz_campaign_id"].replace({"CAMPAIGN_A":916
,"CAMPAIGN_B":936 ,"CAMPAIGN_C":1178}, inplace=True)

#encoding the genders
from sklearn.preprocessing import LabelEncoder
encoder=LabelEncoder()
encoder.fit(data_frame["gender"])
data_frame["gender"]=encoder.transform(data_frame["gender"])
print(data_frame["gender"])

#encoding the age
encoder.fit(data_frame["age"])
data_frame["age"]=encoder.transform(data_frame["age"])
print(data_frame["age"])

data_frame.head()

x=np.array(data_frame.drop(labels=["Approved_Conversion","Total_Conversi
on"], axis=1))
y=np.array(data_frame["Total_Conversion"])

y
y=y.reshape(len(y),1)
y
```

```
from sklearn.preprocessing import StandardScaler
x_sc= StandardScaler()
x = x_sc.fit_transform(x)

from sklearn.model_selection import train_test_split
x_train, x_test, y_train, y_test = train_test_split(x,y,test_size=0.3, random_state=42)

from sklearn.ensemble import RandomForestClassifier
random_fr = RandomForestClassifier(n_estimators = 10, random_state = 0)
random_fr.fit(x_train, y_train)

# Make predictions on the test set
y_pred = random_fr.predict(x_test)

y_pred

from sklearn.metrics import r2_score,mean_squared_error,mean_absolute_error,accuracy_score
mean_abs_error=mean_absolute_error(y_test, y_pred)
mean_sqr_error=mean_squared_error(y_test, y_pred)
r_mean_sqr_error=np.sqrt(mean_sqr_error)
r2_s=r2_score(y_test, y_pred)
a_s=accuracy_score(y_test, y_pred)
```

```
mean_abs_error

#R-squred value
r2_s

a_s

from sklearn.tree import DecisionTreeClassifier
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score
from sklearn.metrics import precision_score
import pandas as pd
from sklearn import tree
decision_tree_clf= DecisionTreeClassifier()

# Train the decision tree classifier
decision_tree_clf.fit(x_train, y_train)

# Make predictions on the test set
y_pred =decision_tree_clf.predict(x_test)
y_pred
```

```
# Calculate the accuracy of the classifier
accuracy = accuracy_score(y_test, y_pred)
print("Accuracy:", accuracy)
#precision = precision_score(y_test,y_pred)
#print(precision)
```

1.Defining the Project Goals and Objectives:

Identified the problem of optimizing social media ad campaigns for better sales forecasting.

Set clear objectives to analyze, visualize, and predict sales based on social media ad data.

2. Dataset Collection and Understanding:

Sourced the dataset from Kaggle containing Facebook ad campaign data.

Explored the dataset structure (11 columns, 1143 data points).

Understood key attributes such as impressions, clicks, spending, and total conversions.

3.Data Preprocessing and Cleaning:

Cleaned the dataset, ensuring no missing or null values were present.

Encoded categorical variables like campaign names and demographic attributes for easier analysis.

4.Exploratory Data Analysis (EDA):

Utilized Pandas, Matplotlib, and Seaborn to perform detailed EDA.

Identified key insights such as:

Campaign C had the highest conversion rates.

Age group 30-34 showed the most engagement across campaigns.

Men clicked on ads more frequently, but women had a higher purchase rate after clicking.

5.Data Visualization:

Created intuitive visualizations like count plots, bar plots, scatter plots, and histograms to better understand relationships between variables.

Highlighted patterns such as the correlation between ad spending and product purchases.

6.Model Development and Training:

Implemented a Random Forest Regression model to predict total conversions.

Split the dataset into training and testing sets (80:20 ratio).

Encoded labels for age and gender to ensure compatibility with the model.

7.Model Evaluation:

Achieved a mean absolute error of 0.99 and a model accuracy of 75.3%.

Refined the model based on evaluation metrics to improve predictive capabilities.

8.Insights and Conclusions:

Determined factors influencing conversions, such as higher spending leading to more product purchases.

Highlighted age groups and demographics for future targeting.

Future Scope:

The future scope of the Facebook Ad Campaign Optimization project is vast, offering several avenues for expansion and refinement. One key area is improving the predictive models by experimenting with advanced machine learning techniques such as Gradient Boosting or XGBoost, along with hyperparameter tuning to enhance accuracy. Additionally, integrating real-time data through APIs and developing dynamic dashboards for live performance tracking would allow marketers to make immediate adjustments to campaigns. Advanced visualizations, like heatmaps or time-series analyses, could further enrich campaign insights. Incorporating Natural Language Processing (NLP) to analyze ad content and user sentiment could optimize messaging for better engagement. Expanding the project to include cross-platform data integration from platforms like Google Ads or Instagram would provide a holistic view of campaign performance. Furthermore, implementing A/B testing and multivariate analysis would help fine-tune ad variations, while segmentation techniques and customer lifetime value predictions could drive more targeted ad strategies. On a broader scale, scaling the project to handle larger campaigns through cloud-based solutions and ensuring data privacy compliance would make it a robust tool for real-world business environments. Ultimately, the project’s potential for integration, personalization, and optimization

could transform it into a powerful asset for enhancing ad performance across various digital channels.

References/Bibliography:

1. McKinney, W. (2017). Python for Data Analysis: Data Wrangling with Pandas, NumPy, and IPython. O'Reilly Media.
2. VanderPlas, J. (2016). Python Data Science Handbook: Essential Tools for Working with Data. O'Reilly Media.

3. Raschka, S., & Mirjalili, V. (2017). Python Machine Learning. Packt Publishing.
4. Hastie, T., Tibshirani, R., & Friedman, J. (2009). The Elements of Statistical Learning: Data Mining, Inference, and Prediction (2nd Ed.). Springer.
5. Subha, B. “Social Media Advertisement and Its Effect in Sales Prediction - An Analysis.” Shanlax International Journal of Management, vol. 8, no. 2, 2020, pp. 40–44.
6. Shi, Yuying and Karniouchina, Ekaterina and Uslay, Can, (When) Can Social Media Buzz Data Replace Traditional Surveys for Sales Forecasting? (April 1, 2020). Rutgers Business Review, Vol. 5, No. 1, 2020, pp.43-60.
7. ilad Dehghani, Mojtaba Khorram Niaki, Iman Ramezani, Rasoul Sali,Evaluating the influence of YouTube advertising for attraction of young customers,Computers in Human Behavior,Volume 59,2016,Pages 165-172,ISSN 0747-5632.
8. Jong Uk Kim, Woong Jin Kim, Sang Cheol Park,Consumer perceptions on web advertisements and motivation factors to purchase in the online shopping,Computers in Human Behavior,Volume 26, Issue 5,2010,Pages 1208-1222,ISSN 0747-5632.
9. Milad Dehghani, Mustafa Tumer,A research on effectiveness of Facebook advertising on enhancing purchase intention of consumers,Computers in Human Behavior,Volume 49,2015,Pages 597-600,ISSN 0747-5632.
10. Hung-Pin Shih,An empirical study on predicting user acceptance of e-shopping on the Web,Information & Management,Volume 41, Issue 3,2004,Pages 351-368,ISSN 0378-7206.