DATA TECHCON COMPANY

EXECUTIVE SUMMARY REPORT FOR PREDICTING APP USER CLICKS

METHODOLOGY AND PROJECT OUTLINE

We will use the CRISP-DM (Cross-Industry Standard Process for Data Mining) framework to execute this end to end data science project as outlined below

- Business Understanding
- Data Understanding
- Data preparation

- Modeling
- Evaluation

UNDERSTANDING THE PROBLEM

Data techcon (DTC) is a cohort-based training institute that specializes in training students on data science and analytics. They launched a marketing campaign through different media apps and are interested in investigating if there is a correlation between marketing spend and user engagement clicks. Analyzing clicks will help to understand consumer behavior and patterns, subsequently providing an insight on the effectiveness of the marketing campaign. The effectiveness of the marketing campaign on potential customers will be studied by measuring marketing spend and user engagement clicks.

I will also build a machine model to help predict the future outcome of DTC app user clicks.

BUSINESS GOAL

The goal is to build a regression model to predict DTC app user clicks to help the senior product manager make informed decisions on how to allocate marketing budget.

DATA SOURCING

The data set was obtained from Data Techcon in excel csv file format. The excel file was then imported to Python for data analysis.

DATA SELECTION

- Excel csv file
- PYTHON

APPROACH

- Import Python libraries
- Data loading and cleaning
- Show the relationships between variables using scatter plots.
- Build a predictive model using Linear Regression algorithm using scikit learn.

DATA LOADING AND VALIDATION

Import Python libraries necessary for the project.

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline

# Pandas for exploratory data analysis.
# Numpy for arithmetic operations.
# Matplotlib for data visualization.
# Seaborn for high level interface data visualization.
```

Load the csv data into a dataframe named “df” and validate the data in Python

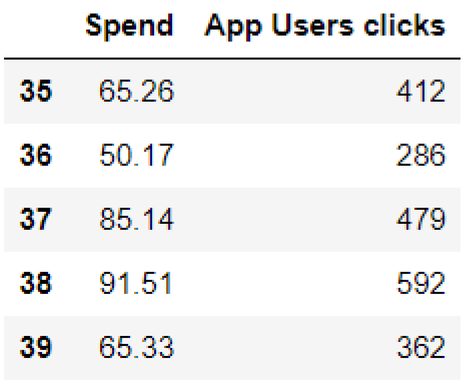
The df.head() and df.tail() functions show the top 5 rows and bottom 5 rows of the data respectively.

```
# Load data into python
df=pd.read_csv('C:/Users/loruf/Downloads/DTC/asswk8/App_user_engagement.csv
')
```

```
# Data validation: Top 5 rows
df.head()
```

	Spend	App Users clicks
0	22.61	165
1	37.28	228
2	55.57	291
3	45.42	247
4	50.22	290

```
# Data validation: Bottom 5 rows
df.tail()
```

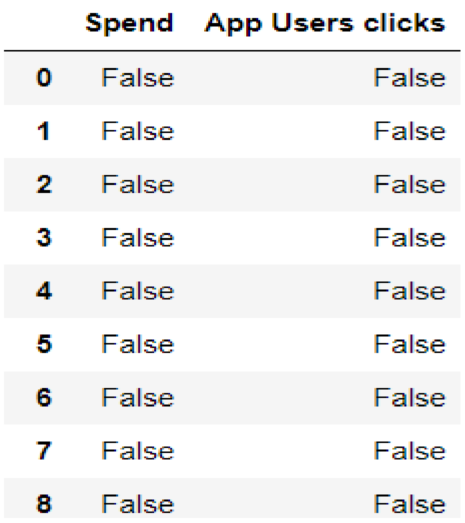


	Spend	App Users clicks
35	65.26	412
36	50.17	286
37	85.14	479
38	91.51	592
39	65.33	362

DATA CLEANING

This involves checking the data for missing value, data type format, outliers and duplicate entries. There were no missing values and duplicates in the data.

```
# Check for missing values
df.isnull()
```



	Spend	App Users clicks
0	False	False
1	False	False
2	False	False
3	False	False
4	False	False
5	False	False
6	False	False
7	False	False
8	False	False

```
#Check for duplicate values
df.duplicated()
```

```
0     False
1     False
2     False
3     False
4     False
5     False
6     False
7     False
8     False
9     False
10    False
11    False
12    False
```

FINDINGS

- Dataset contains 40 observations and 2 columns
- No evidence of missing data.
- No duplicate entries.

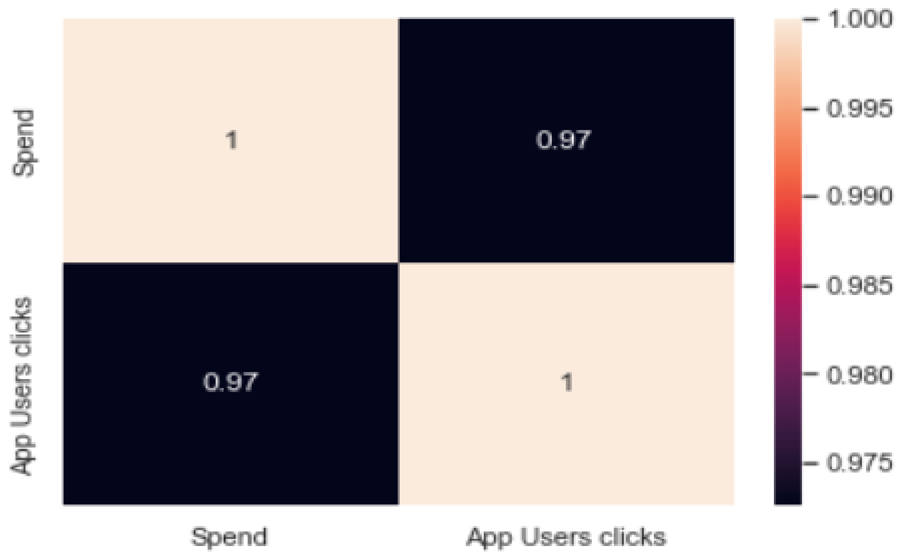
EXPLORATORY DATA ANALYSIS (EDA)

Correlation Analysis

```
#correlation table between spend and app users clicks
df.corr()
```

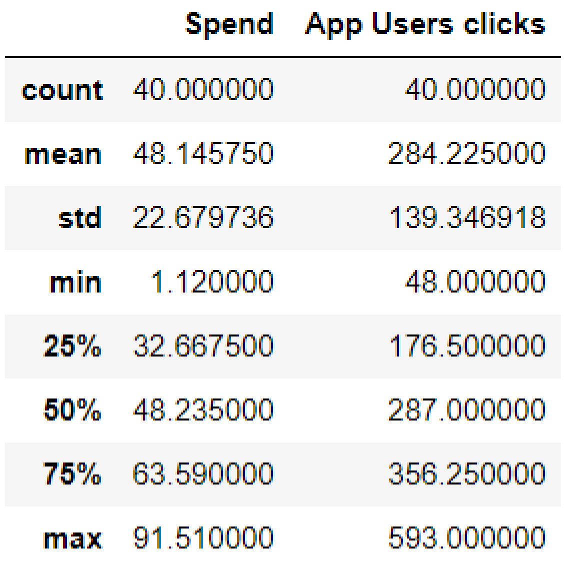
	Spend	App Users clicks
Spend	1.000000	0.972575
App Users clicks	0.972575	1.000000

```
#heatmap showing the visualization of correlation
import seaborn as sns
import matplotlib.pyplot as plt
sns.heatmap(df[['Spend','App Users clicks']].corr(),annot=True)
plt.show()
```



Insights: App User Clicks and ads spend are highly correlated (Strong linear relationship between the 2 variables).

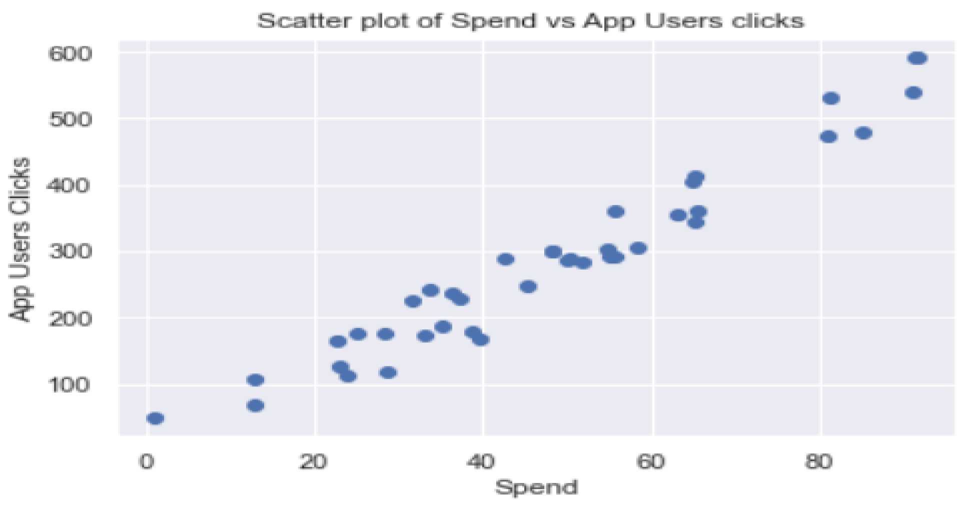
Descriptive Statistics



	Spend	App Users clicks
count	40.000000	40.000000
mean	48.145750	284.225000
std	22.679736	139.346918
min	1.120000	48.000000
25%	32.667500	176.500000
50%	48.235000	287.000000
75%	63.590000	356.250000
max	91.510000	593.000000

Scatter plot of App User Clicks and ads spend

```
#scatter plot App users click vs spend
plt.scatter(df['Spend'], df['App Users clicks'])
plt.xlabel("Spend")
plt.ylabel("App Users Clicks")
plt.title("Scatter plot of Spend vs App Users clicks ")
plt.show()
```



Insights: Clearly, there is a linear relationship between ads spend and App user clicks.
Recommendation: Increase marketing budget to increase conversion rate.

BUILD MACHINE LEARNING MODEL: LINEAR REGRESSION

The library sklearn was used to import training-test package, linear model and metrics functions. The data was split into 80% train data and 20% test data.

```
#Import necessary libraries
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn import metrics
%matplotlib inline

#Validate data
df.head()

# Check the info of datasets
```

```
df.info()
```

```
# input and predictor variable
X=df.drop(['App Users clicks'],axis=1)
Y=df['App Users clicks']
```

MODEL TRAINING

Dataset was split into 80% training and 20% testing.

```
# fit training data to model
regressor=LinearRegression()
regressor.fit(X_train,Y_train)
```

```
LinearRegression()
```

```
print (regressor.intercept_)
print (regressor.coef_)
```

```
-4.863875223198363
[5.9848097]
```

```
#Check prediction score
regressor.score(X_test,Y_test)
```

```
0.93172722814936
```

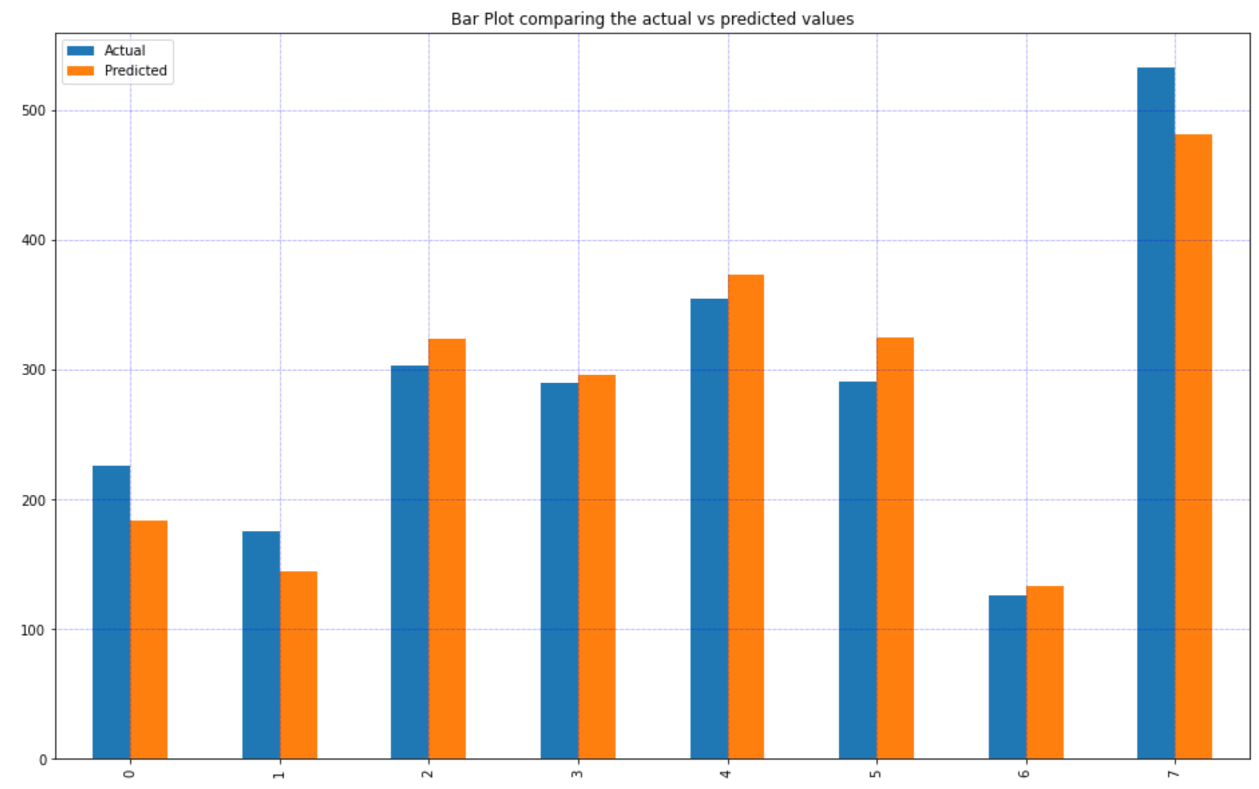
The prediction score is 0.932 which is 93% and good. Above 50% is good.

```
#Get predicted y
y_pred=regressor.predict(X_test)

# Check predictions where accuracy levels is achieved
y_test=np.array(list(Y_test))
y_pred=np.array(y_pred)
df=pd.DataFrame({'Actual':y_test.flatten(),'Predicted':y_pred.flatten()})
df
```

	Actual	Predicted
0	226	183.777327
1	175	144.876064
2	303	323.881722
3	290	295.693268
4	355	373.256402
5	291	325.078684
6	126	133.026140
7	533	481.581458

```
#Visual with 1st 25 values
df1=df.head(25)
df1.plot(kind='bar',figsize=(16,10))
plt.grid(which='major',linestyle='-',linewidth='0.5',color='blue')
plt.grid(which='major',linestyle=':',linewidth='0.5',color='blue')
plt.title("Bar Plot comparing the actual vs predicted values")
plt.show()
```



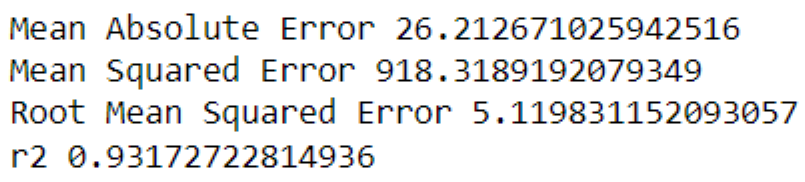
Key Insight from visual

There is a low variance in the model. The error rate is low.

CHECKING THE MODEL PERFORMANCE(Regression Model)

The model performance was measured using Mean Absolute error, Mean squared error , Root Mean squared error and r squared.

```
# Calculate error of the model
print('Mean Absolute Error', metrics.mean_absolute_error(y_test,y_pred))
print('Mean Squared Error', metrics.mean_squared_error(y_test,y_pred))
print('Root Mean Squared Error',
np.sqrt(metrics.mean_absolute_error(y_test,y_pred)))
print('r2', metrics.r2_score(y_true=y_test,y_pred=y_pred))
```



```
Mean Absolute Error 26.212671025942516
Mean Squared Error 918.3189192079349
Root Mean Squared Error 5.119831152093057
r2 0.93172722814936
```

INTERPRETATION

- **The Mean Absolute Error** (MAE) is a measure of the average absolute difference between the predicted and actual values. In this case, the MAE is 26.213, which means that on average, the model's predictions are off by 26.213 units.
- **The Mean Squared Error** (MSE) is a measure of the average squared difference between the predicted and actual values. In this case, the MSE is 918.319, which is higher than the MAE. This suggests that the model is more sensitive to larger errors.
- **The Root Mean Squared Error** (RMSE) is the square root of the MSE and is a commonly used measure of regression model accuracy. In this case, the RMSE is 5.120, which means that the average prediction error is approximately 5.120 units.
- **The r2 score** is a measure of how well the model fits the data, with higher values indicating a better fit. The r2 score in this case is 0.932, which means that the model explains approximately 93% of the variance in the data.

RECOMMENDATIONS

The r2 score in this case is 0.932, which means that the model explains approximately 93% of the variance in the data. This is above average which means the model is reliable.

CONCLUSION

In this article, I have been able to walk you through a step by step approach to building a predictive model to App Users click to enable them to make informed decisions on how to allocate marketing budget.