**Centre for Development of Advanced Computing**

**(CDAC),** **Noida**

**POST GRADUATE DIPLOMA IN BIG DATA ANALYTICS (DBDA)**

****

**A REPORT ON**

**“Directing Customers To Subscription Through Financial App Behaviour Analysis”**

**SUBMITTED BY**

Kaushal Jeena (220920525019)

Vedant Choubey (220920525058)

Vijay Kumar (220920525059)

**UNDER THE GUIDANCE OF**

Mr. Shivam Pandey

(Academic Year: **Sep 2022 - Mar 2023**)

**CONTENTS**

1. Certificate

2. Acknowledgement

3. Abstract

4. Introduction

5. Exploratory data analysis (EDA)

6. Know about dataset

7. Data Vizualization

8. Data preprocessing

9. Feature Scaling

10.Data Model

11.Conclusion

12.Evaluation

**CERTIFICATE**

This is to certify that Kaushal Jeena, Vedant Choubey and Vijay Kumar, students of PGDBDA, have successfully completed the project entitled “Directing Customers To Subscription Through Financial App Behaviour Analysis” .The project has been submitted to CDAC, Noida in fulfilment of the requirements for the award of the diploma.

Date: 13/03/2023

Place: Noida(U.P.)

Signature:

Kaushal Jeena (220920525019)

Vedant Choubey (2209205250)

Vijay Kumar (220920525059)

PGDBDA - CDAC Noida(U.P.)

**ACKNOWLEDGEMENT**

I would like to express my deepest gratitude to all the individuals who have supported me in completing this project.

Firstly, I would like to thank my supervisor Mr. Shivam Pandey, for his invaluable guidance and constructive feedback throughout the project. His encouragement, suggestions and insights have been instrumental in shaping the direction of this research.

I would also like to extend my sincere appreciation to the all faculty members of CDAC Noida who have provided me with a conducive environment for learning. Their passion and commitment to teaching have inspired me to pursue excellence in my academic pursuits.

In addition, I am grateful to my family and friends for their unwavering support and understanding during my academic journey. Their encouragement and belief in me have been a constant source of motivation.

**ABSTRACT**

The Financial Technology company (Fin-Tech Company) launches their mobile app. This app is used for financial purposes like bank loans, savings, etc. in one place. It has two versions free and premium. The free version app contains basic features and customer wants to use the premium feature then they must pay some amount to unlock it.

The main goal of the company is to sell the premium version app with low advertisement cost but they don’t know how to do it. That’s a reason they are provided the premium feature in the free version app for 24 hours to collect the customer’s behaviour. After that, the company hired the Machine Learning Engineer to find insight from the collected data (customer’s behaviour).

The job of the ML engineer is to find or predict new customer who is interested to buy the product or not. If the customers will buy a product anyway so no need to give an offer to that customer and loss the business. Only give offers to those customers who are interested to use premium version app but they can’t afford its cost. So the company will give offers to those customers and earn more money.

**1.INTRODUCTION:**

In today's digital age, subscription-based business models have become increasingly popular for companies offering digital products and services. Subscriptions provide a reliable and recurring revenue stream, while also allowing businesses to build lasting relationships with their customers. However, getting users to sign up for a subscription can be challenging, especially in a highly competitive market.

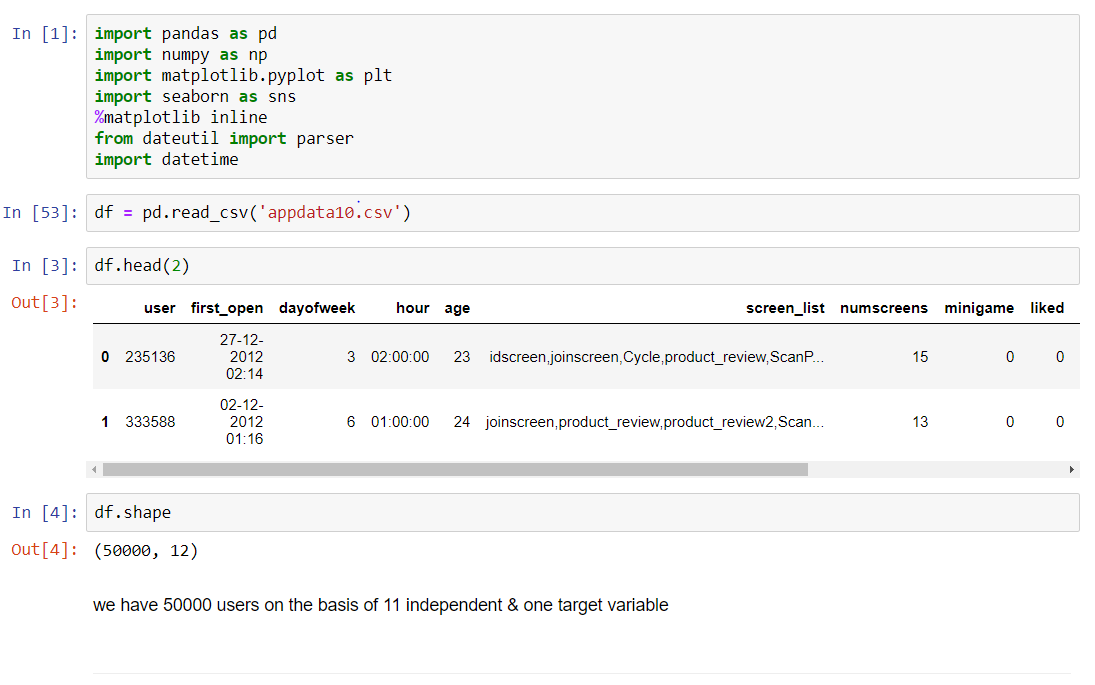
To tackle this challenge, companies are turning to app behavior analysis to direct their customers towards subscription products. By analysing user behaviour, companies can identify patterns that suggest what motivates users to subscribe, and use this information to create targeted marketing campaigns aimed at increasing subscription rates.

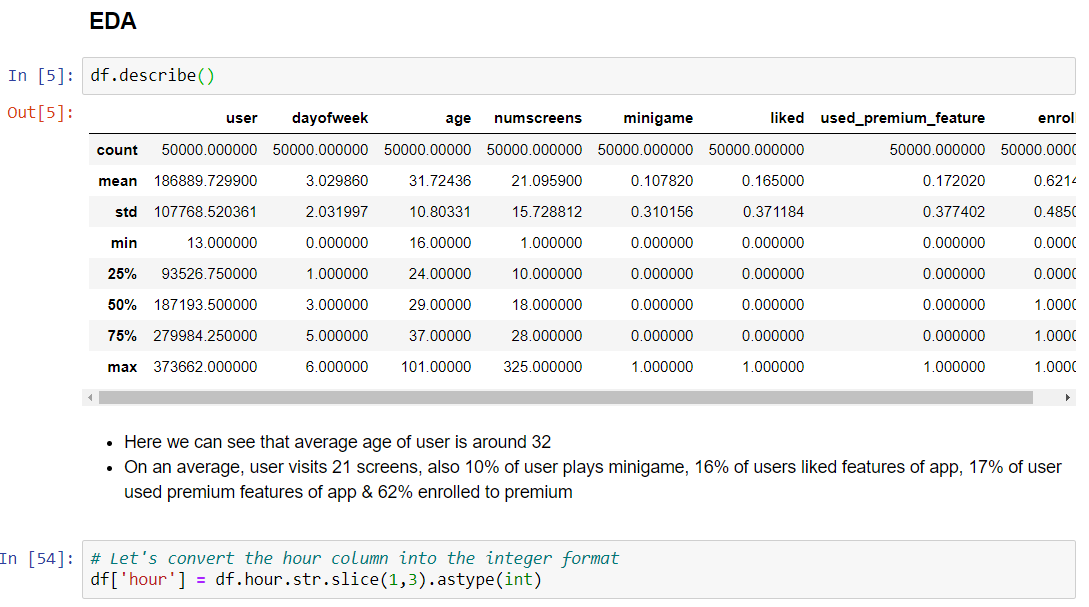
App behaviour analysis involves analysing data about user behaviour, such as how often they use the app, what features they interact with, and when they use the app. This data can be used to gain insights into the user's preferences and behaviour, and to identify which marketing strategies are most likely to be successful.

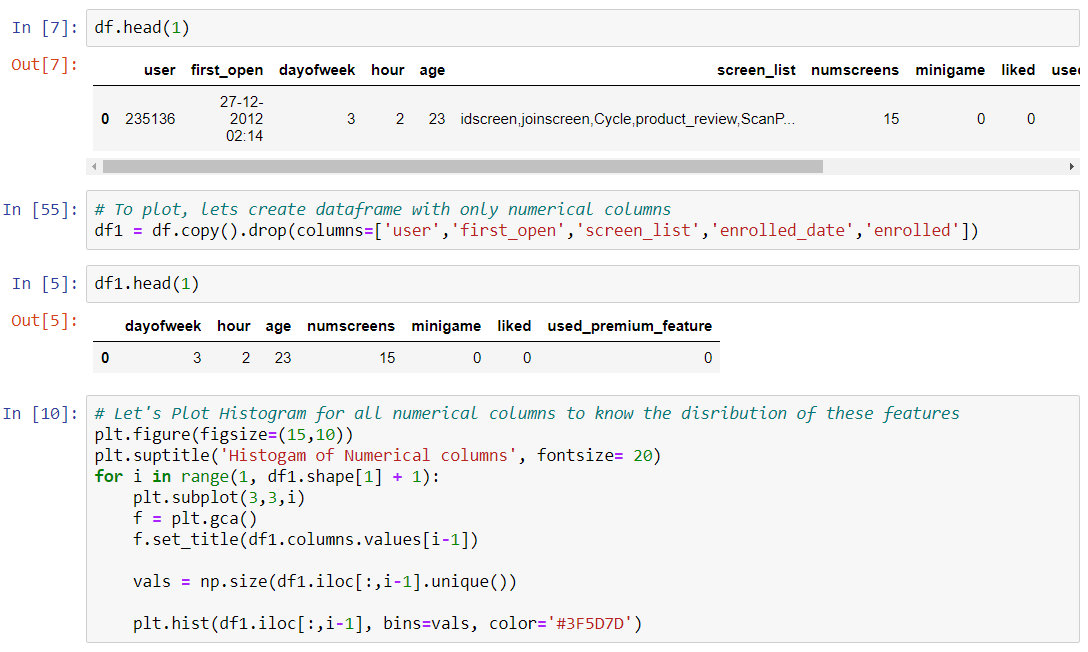
Overall, directing customers to subscription products through app behaviour analysis is an effective strategy for companies looking to increase their subscription rates. By leveraging data analytics to better understand their customers' behaviour, companies can create more effective marketing campaigns and drive higher subscription rates, resulting in a more successful subscription-based business.

Overall, directing customers to subscription products through app behaviour analysis is an increasingly important strategy for companies looking to build successful subscription-based businesses. By leveraging the power of data analysis, companies can gain a deep understanding of user behaviour and create marketing campaigns that are more likely to convert users into paying subscribers.

**2. Exploratory data analysis (EDA) with snapshots:**







**3.** **Know about dataset:**

As you can see in **df**DataFrame, there are 50,000 users data with 12 different features. Let’s know each and every feature in brief.

**1. user**: Unique ID for each user.

**2. first\_open**: Date *(yy-mm-dd)* and time *(Hour:Minute:Seconds:Milliseconds)* of login on app first time.

**3. dayofweek**: On which day user logon.

* 0: Sunday
* 1: Monday
* 2: Tuesday
* 3: Wednesday
* 4: Thursday
* 5: Friday
* 6: Saturday

**4. Hour**: Time of a day in 24-hour format customer logon. It is correlated with **dayofweek**column.

**5. age**: The age of the registered user.

**6. screen\_list**: The name of multiple screens seen by customers, which are separated by a comma.

**7. numscreens**: The total number of screens seen by customers.

**8. minigame**: Tha app contains small games related to finance. If the customer played mini-game then 1 otherwise 0.

**9. used\_premium\_feature**: If the customer used the premium feature of the app then 1 otherwise 0.

**10. enrolled**: If the user bought a premium feature app then 1 otherwise 0.

**11. enrolled\_date**: On the date (yy-mm-dd) and time (Hour:Minute:Seconds:Milliseconds) the user bought a premium features app.

**12. liked**: The each screen of the app has a like button if the customer likes it then 1 otherwise 0.

Find the null value in DataFrame using **DataFrame.isnull()** method and take summation by **sum()** method.

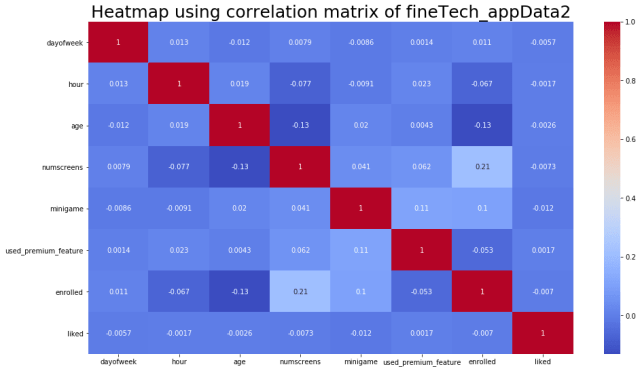
**4.Data visualization:**

**Heatmap using the correlation matrix**

**Heatmap used to find the correlation between each and every features using the correlation matrix.**

Text

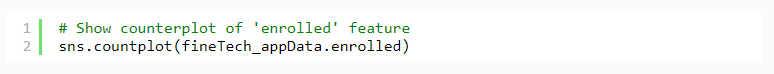
Description automatically generated

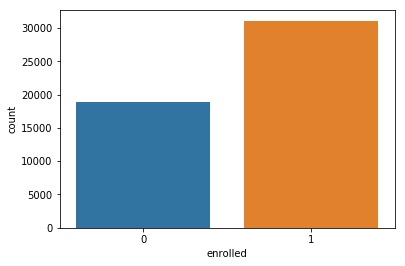


In the dataset there **is no strong correlation between any features**. There is **little correlation between ‘numscreens’ and ‘enrolled’**. It means that those customers saw more screen they are taking premium app. There is a slight correlation between ‘minigame’ with ‘enrolled’ and ‘used\_premium\_feature’. The slightly negative correlation between ‘age’ with ‘enrolled’ and ‘numscreens’. It means that older customers do not use the premium app and they don’t see multiple screens.

**We visualize the counterplot of enrolled data.**

**Count plot of enrolled**





Text

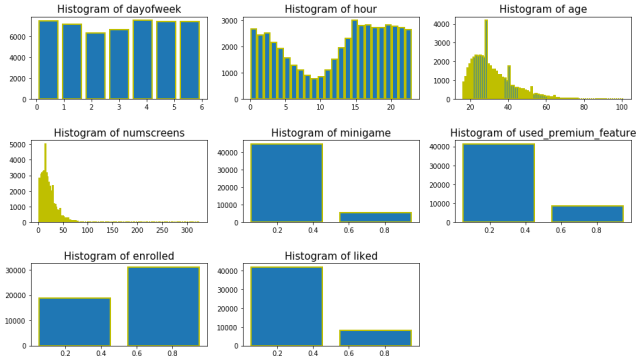
Description automatically generated

Graphical user interface, text, application

Description automatically generated

Text

Description automatically generated



In the above histogram, we can see minigame, used\_primium\_feature, enrolled, and like they have only two values and how they distributed.

The histogram of ‘dayofweek’ shows, on Tuesday and Wednesday slightly fewer customer registered the app.

The histogram of ‘hour’ shows the less customer register on the app around 10 AM.

The ‘age’ histogram shows, the maximum customers are younger.

The ‘numsreens’ histogram shows the few customers saw more than 40 screens.

**Correlation barplot with ‘enrolled’ feature**

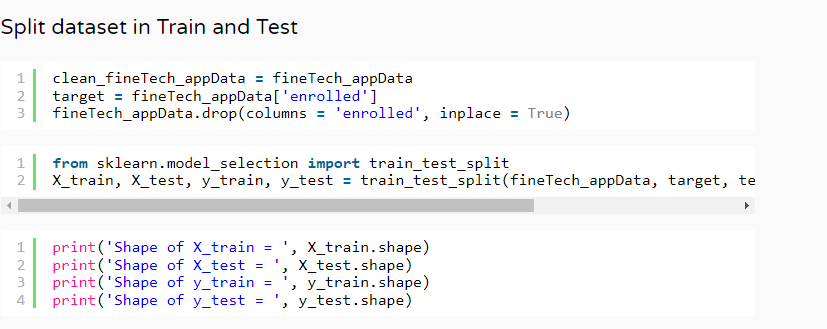
Now we are trying to know which feature is strongly correlated with ‘enrolled’ feature with positive or negative through barplot.

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8 | # show corelation barplot    sns.set() # set background dark grid  plt.figure(figsize **=** (14,5))  plt.title("Correlation all features with 'enrolled' ", fontsize **=** 20)  fineTech\_appData3 **=** fineTech\_appData2.drop(['enrolled'], axis **=** 1) # drop 'enrolled' feature  ax **=**sns.barplot(fineTech\_appData3.columns,fineTech\_appData3.corrwith(fineTech\_appData2.enrolled)) # plot barplot  ax.tick\_params(labelsize**=**15, labelrotation **=** 20, color **=**"k") # decorate x & y ticks font |

Text

Description automatically generated with medium confidence

### **5.Data preprocessing:**



Text

Description automatically generated

**6.Feature Scaling:**

Graphical user interface, text, application

Description automatically generated

**7. Data Model:**

### **A.Machine Learning Model Building**

The target variable is categorical type 0 and 1, so we have to use supervised classification algorithms.

To build the best model, we have to train and test the dataset with multiple Machine Learning algorithms then we can find the best ML model. So let’s try.

First, we import the required packages.

|  |  |
| --- | --- |
| 1  2 | # import required packages  **from** sklearn.metrics **import** confusion\_matrix, classification\_report, accuracy\_score |

#### **B. Decision Tree Classifier**

|  |  |
| --- | --- |
| 1  2  3  4  5  6 | # Decision Tree Classifier  **from** sklearn.tree **import** DecisionTreeClassifier  dt\_model **=** DecisionTreeClassifier(criterion**=** 'entropy', random\_state**=**0)  dt\_model.fit(X\_train, y\_train)  y\_pred\_dt **=** dt\_model.predict(X\_test)  accuracy\_score(y\_test, y\_pred\_dt) |

**Output >>>**0.6936

|  |  |
| --- | --- |
| 1  2  3  4  5 | # train with Standert Scaling dataset  dt\_model2 **=** DecisionTreeClassifier(criterion**=** 'entropy', random\_state**=**0)  dt\_model2.fit(X\_train\_sc, y\_train)  y\_pred\_dt\_sc **=** dt\_model2.predict(X\_test\_sc)  accuracy\_score(y\_test, y\_pred\_dt\_sc) |

**Output >>>**0.6932

#### **> C. K – Nearest Neighbor Classifier:**

|  |  |
| --- | --- |
| 1  2  3  4  5  6 | **from** sklearn.neighbors **import** KNeighborsClassifier  knn\_model **=** KNeighborsClassifier(n\_neighbors**=**5, metric**=**'minkowski', p**=**2,)  knn\_model.fit(X\_train, y\_train)  y\_pred\_knn **=** knn\_model.predict(X\_test)    accuracy\_score(y\_test, y\_pred\_knn) |

**Output >>>**0.6994

|  |  |
| --- | --- |
| 1  2  3  4  5  6 | # train with Standert Scaling dataset  knn\_model2 **=** KNeighborsClassifier(n\_neighbors**=**5, metric**=**'minkowski', p**=**2,)  knn\_model2.fit(X\_train\_sc, y\_train)  y\_pred\_knn\_sc **=** knn\_model2.predict(X\_test\_sc)    accuracy\_score(y\_test, y\_pred\_knn\_sc) |

**Output >>>**0.7314

#### **D. Naive Bayes Classifier**

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7 | # Naive Bayes  **from** sklearn.naive\_bayes **import** GaussianNB  nb\_model **=** GaussianNB()  nb\_model.fit(X\_train, y\_train)  y\_pred\_nb **=** nb\_model.predict(X\_test)    accuracy\_score(y\_test, y\_pred\_nb) |

**Output >>>**0.7114

|  |  |
| --- | --- |
| 1  2  3  4  5  6 | # train with Standert Scaling dataset  nb\_model2 **=** GaussianNB()  nb\_model2.fit(X\_train\_sc, y\_train)  y\_pred\_nb\_sc **=** nb\_model2.predict(X\_test\_sc)    accuracy\_score(y\_test, y\_pred\_nb\_sc) |

**Output >>>**0.7114

#### **E. Random Forest Classifier**

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7 | # Random Forest Classifier  **from** sklearn.ensemble **import** RandomForestClassifier  rf\_model **=** RandomForestClassifier(n\_estimators**=**10, criterion**=**'entropy', random\_state**=**0)  rf\_model.fit(X\_train, y\_train)  y\_pred\_rf **=** rf\_model.predict(X\_test)    accuracy\_score(y\_test, y\_pred\_rf) |

**Output >>>**0.7621

|  |  |
| --- | --- |
| 1  2  3  4  5  6 | # train with Standert Scaling dataset  rf\_model2 **=** RandomForestClassifier(n\_estimators**=**10, criterion**=**'entropy', random\_state**=**0)  rf\_model2.fit(X\_train\_sc, y\_train)  y\_pred\_rf\_sc **=** rf\_model2.predict(X\_test\_sc)    accuracy\_score(y\_test, y\_pred\_rf\_sc) |

**Output >>>** 0.7616

#### **Logistic Regression**

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7 | # Logistic Regression  **from** sklearn.linear\_model **import** LogisticRegression  lr\_model **=** LogisticRegression(random\_state **=** 0, penalty **=** 'l1')  lr\_model.fit(X\_train, y\_train)  y\_pred\_lr **=** lr\_model.predict(X\_test)    accuracy\_score(y\_test, y\_pred\_lr) |

**Output >>>** 0.7684

|  |  |
| --- | --- |
| 1  2  3  4  5  6 | # train with Standert Scaling dataset  lr\_model2 **=** LogisticRegression(random\_state **=** 0, penalty **=** 'l1')  lr\_model2.fit(X\_train\_sc, y\_train)  y\_pred\_lr\_sc **=** lr\_model2.predict(X\_test\_sc)    accuracy\_score(y\_test, y\_pred\_lr\_sc) |

**Output >>>** 0.7681

#### **Support Vector Classifier**

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7 | # Support Vector Machine  **from** sklearn.svm **import** SVC  svc\_model **=** SVC()  svc\_model.fit(X\_train, y\_train)  y\_pred\_svc **=** svc\_model.predict(X\_test)    accuracy\_score(y\_test, y\_pred\_svc) |

**Output >>>** 0.7616

|  |  |
| --- | --- |
| 1  2  3  4  5  6 | # train with Standert Scaling dataset  svc\_model2 **=** SVC()  svc\_model2.fit(X\_train\_sc, y\_train)  y\_pred\_svc\_sc **=** svc\_model2.predict(X\_test\_sc)    accuracy\_score(y\_test, y\_pred\_svc\_sc) |

**Output >>>** 0.779

#### **XGBoost Classifier**

|  |  |
| --- | --- |
| 1  2  3  4  5  6 | # XGBoost Classifier  **from** xgboost **import** XGBClassifier  xgb\_model **=** XGBClassifier()  xgb\_model.fit(X\_train, y\_train)  y\_pred\_xgb **=** xgb\_model.predict(X\_test)  accuracy\_score(y\_test, y\_pred\_xgb) |

**Output >>>** 0.7748

|  |  |
| --- | --- |
| 1  2  3  4  5  6 | # train with Standert Scaling dataset  xgb\_model2 **=** XGBClassifier()  xgb\_model2.fit(X\_train\_sc, y\_train)  y\_pred\_xgb\_sc **=** xgb\_model2.predict(X\_test\_sc)    accuracy\_score(y\_test, y\_pred\_xgb\_sc) |

**Output >>>** 0.7748

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19 | # XGB classifier with parameter tuning  xgb\_model\_pt1 **=** XGBClassifier(   learning\_rate **=**0.01,   n\_estimators**=**5000,   max\_depth**=**4,   min\_child\_weight**=**6,   gamma**=**0,   subsample**=**0.8,   colsample\_bytree**=**0.8,   reg\_alpha**=**0.005,   objective**=** 'binary:logistic',   nthread**=**4,   scale\_pos\_weight**=**1,   seed**=**27)    xgb\_model\_pt1.fit(X\_train, y\_train)  y\_pred\_xgb\_pt1 **=** xgb\_model\_pt1.predict(X\_test)    accuracy\_score(y\_test, y\_pred\_xgb\_pt1) |

**Output >>>** 0.7887

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  20 | # XGB classifier with parameter tuning  # train with Stander Scaling dataset  xgb\_model\_pt2 **=** XGBClassifier(   learning\_rate **=**0.01,   n\_estimators**=**5000,   max\_depth**=**4,   min\_child\_weight**=**6,   gamma**=**0,   subsample**=**0.8,   colsample\_bytree**=**0.8,   reg\_alpha**=**0.005,   objective**=** 'binary:logistic',   nthread**=**4,   scale\_pos\_weight**=**1,   seed**=**27)    xgb\_model\_pt2.fit(X\_train\_sc, y\_train)  y\_pred\_xgb\_sc\_pt2 **=** xgb\_model\_pt2.predict(X\_test\_sc)    accuracy\_score(y\_test, y\_pred\_xgb\_sc\_pt2) |

**8. Conclusion:**

In conclusion, directing customers to subscription products through app behaviour analysis is a powerful strategy for companies looking to increase their subscription rates. By analysing user behaviour, companies can gain valuable insights into what motivates users to subscribe, and use this information to create targeted marketing campaigns that are more likely to convert users into paying subscribers.

App behaviour analysis involves analysing data about user behaviour, such as how often they use the app, what features they interact with, and when they use the app. By identifying patterns in this data, companies can create targeted marketing campaigns that speak directly to users' interests and preferences, increasing the likelihood of converting users into paying subscribers.

With the increasing popularity of subscription-based business models, directing customers to subscription products through app behaviour analysis is becoming a critical strategy for businesses. By leveraging the power of data analytics, companies can gain a deep understanding of user behaviour and create marketing campaigns that are more likely to drive subscription rates and lead to a successful subscription-based business.

We have completed the **Machine learning Project** successfully with **78.87%** accuracy which is great for **‘Directing Customers to Subscription Through Financial App Behaviour Analysis’** project.

**9.Evaluation:**

* Directing customers to subscription products through app behaviour analysis is a powerful strategy for companies looking to increase their subscription rates. By analysing user behaviour and identifying patterns, companies can gain insights into what motivates users to subscribe and use this information to create targeted marketing campaigns. This approach has several advantages:
* Targeted marketing: By using app behaviour analysis, companies can create marketing campaigns that speak directly to users' interests and preferences, increasing the likelihood of conversion.
* Cost-effective: This approach can be cost-effective as it involves analysing existing data to identify patterns and create targeted marketing campaigns, rather than relying on more traditional methods that may require significant resources.
* Data-driven decisions: By relying on data analytics, companies can make more informed decisions based on concrete evidence rather than relying on assumptions.
* Improved user experience: By using app behaviour analysis, companies can gain insights into how users interact with the app and use this information to improve the overall user experience, leading to higher customer satisfaction and retention rates.
* However, there are some limitations to this approach:
* Limited sample size: App behaviour analysis relies on data from existing users, which may not be representative of the wider target audience.
* Privacy concerns: Collecting and analysing user data may raise privacy concerns, and companies need to be transparent about their data collection and usage practices.
* Changing user behaviour: User behaviour is not static, and patterns may change over time, requiring continuous analysis and adaptation of marketing strategies.
* Overall, directing customers to subscription products through app behaviour analysis is an effective strategy for companies looking to increase their subscription rates. However, companies need to be aware of the limitations and ethical considerations associated with this approach.