**USER SUBSCRIPTION CLASSIFICATION**

*A report submitted in partial fulfillment of the requirements for the Award of Degree of*

**BACHELOR OF TECHNOLOGY**

in

**COMPUTER SCIENCE AND ENGINEERING**

By

**TAMMISETTI VENKATALAKSHMI**

**Reg. No: 23B91A05S9**

Submitted to

**Dr Chigurupati Ravi Swaroop** (Assistant Professor) &

**Dr Hema LathaDendukuri** (Assistant Professor)



**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

**S.R.K.R. ENGINEERING COLLEGE(A)**

SRKR MARG, CHINNA AMIRAM, BHIMAVARAM-534204, A.P

(Recognized by A.I.C.T.E New Delhi) (Accredited by NBA & NAAC) (Affiliated to JNTU, KAKINADA)

SAGI RAMA KRISHNAM RAJU ENGINEERING COLLEGE

(Autonomous)

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**



**CERTIFICATE**

This is to certify that the **ML-XPLORE** Report titled **“USER SUBSCRIPTION CLASSIFICATION**

**”** is the bonafide work done by **Ms. TAMMISETTI VENKATALAKSHMI** bearing **Register Number: 23B91A05S9** in the second year second semester at **SRKR Engineering College, Bhimavaram** from 09 January 2025 to 10 January 2025 in partial fulfillment of the requirements for the award of the degree of Bachelor of Technology in Computer Science and Engineering.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Dr Bh.B.S.R.K.Raju**  **(Professor and Head)** |  | **Dr Chigurupati Ravi Swaroop**  (Assistant Professor) |  | **Dr Hema LathaDendukuri** (Assistant Professor) |  |
|  |  |  |  |  |

**TABLE OF CONTENTS**

1.0 Introduction

1.1 What are the different types of Machine Learning?

1.2 Benefits of using Machine Learning in User Subscription Classification 2.0 User Subscription Classification 3.0 AI / ML Modelling and Results

3.1 Problem Statement – User Subscription Classification 3.2 Data Science Project Life Cycle

3.2.1 Data Exploratory Analysis

3.2.2 Data Pre-processing

3.2.2.1 Managing Duplicate and Low Variation Data

3.2.2.2 Identifying and Addressing Missing Variables

3.2.2.3 Handling Outliers

3.2.2.4 Categorical data and Encoding Techniques

3.2.2.5 Feature Scaling

3.2.3 Selection of Dependent and Independent Variables

3.2.4 Data Balancing

3.2.5 Models used for Development

3.2.5.1 Logistic Regression

3.2.5.2 Decision Tree Model

3.2.5.3 Random Forest Model

3.2.5.4 Extra Trees Classification Model

3.2.5.5 K- Nearest Neighbors Model

3.2.5.6 Light Gradient Boosting Model

3.2.5.7 Gaussian Naive Bayes Model

3.2.5.8 Bagging classifier

3.2.5.9 Gradient Boosting classifier

3.3 AI / ML Models Analysis and Final Results

3.3.1 Combined Code for all Models

3.3.2 Extra Tree Classifier using GridSearchCV– Code

3.3.3 Random Forest using GridSearchCV– Code

3.3.4 Light Gradient Boosting Classifier using GridSearchCV– Code

4.0 Conclusions and Future Work

5.0 References

6.0 Appendices

**ABSTRACT**

In the dynamic landscape of human resources, this machine learning project focuses on the development of a robust predictive model for employee salary estimation. Utilizing a curated dataset encompassing factors such as age, years of experience, education level, and position, our aim is to create a model that not only accurately predicts salaries but also provides actionable insights for organizations.

The project begins with a comprehensive data exploration, addressing missing values and ensuring data quality. Exploratory Data Analysis (EDA) uncovers patterns and relationships, setting the stage for insightful feature engineering. Key variables are extracted, and categorical features are encoded to prepare the data for model training.

Multiple regression models, including linear regression, decision trees and random forest, are implemented and fine-tuned to predict salaries. Evaluation metrics such as Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), and R-squared provide a comprehensive assessment of model performance.

Results and discussions delve into the interpretability of predictions, highlighting the impact of each feature on salary outcomes. Model strengths and areas for improvement are examined, offering a nuanced understanding of its capabilities.

1. **Introduction**

Building a machine learning model for predicting employee salaries involves a systematic process. The first step is to collect relevant data, encompassing employee profiles, job roles, education levels, experience, and skills. Subsequently, the data undergoes preprocessing to handle missing values, outliers, and conversion of categorical variables. Feature engineering may enhance the model's predictive power by creating new relevant features. Exploratory Data Analysis (EDA) aids in understanding relationships within the data. Following this, an appropriate regression algorithm, such as linear regression, decision trees, or gradient boosting, is chosen for model selection. The dataset is then split into training and testing sets to train and evaluate the model. Evaluation metrics like Mean Absolute Error (MAE) or Mean Squared Error (MSE) assess the model's accuracy. Hyperparameter tuning optimizes the model's performance, and once satisfied, the model is deployed for predictions in a production environment. Continuous monitoring and updates are crucial for maintaining accuracy over time, and ethical considerations address biases and fairness in the salary prediction process. It's important to interpret model predictions as aids to decision-making rather than absolute certainties.

**1.1 What are the different types of Machine Learning?**

Machine Learning is a subset of Artificial Intelligence (AI). Basically, machine learning automates the prediction of data by learning the trends of data and improving performance.

Machine Learning comprises of set of algorithms which has their own unique way of learning the data and predicting the outcomes. This Machine Learning is then classified into four types.

* + 1. **Supervised Machine Learning**

Supervised machine learning is based on supervision in which we are subjected to train labelled dataset. The main goal of the supervised learning it to map the input variables with the output variables and provide an optimal prediction.

The following are the two types of problems in supervised learning:

1. Regression
2. Classification
   * 1. **Unsupervised Machine Learning**

In unsupervised machine learning the machine interprets the unlabeled dataset and trains and predicts the output without supervision.

In this type of machine learning the model is trained with the data which is neither classified nor labelled and groups the unsorted dataset based on their patterns and similarities with differences under consideration.

Unsupervised learning is again classified into the following types

1. Clustering
2. Association
   * 1. **Semi-Supervised Machine Learning**

Semi-Supervised machine learning is one of the machine learning algorithms which lies between Supervised and Unsupervised machine learning. To overcome the complications and drawbacks of Supervised and unsupervised machine learning can be overcome by using Semi-supervised machine learning technique.

**1.1.4 Reinforcement Machine Learning**

Reinforcement learning is an area of Machine Learning. It is about taking suitable action to maximize reward in a particular situation. It is employed by various software and machines to find the best possible behavior or path it should take in a specific situation.

Reinforcement learning is all about making decisions sequentially. In simple words, we can say that the output depends on the state of the current input and the next input depends on the output of the previous input

**1.2 Benefits of using Machine Learning in Employee Salary Prediction.**

The progressively growing range of applications of machine learning permits us to glimpse at a future where data, analysis, and innovation work hand-in-hand to assist organizations in optimizing salary structures without employees ever realizing the underlying complexity. Soon, it will be quite common to find ML-based applications embedded with real-time employee data available from different organizational systems, thereby increasing the accuracy and efficiency of salary predictions that were unfeasible before.

Keeping salary structures competitive and fair is an extensive process, and while technology has simplified data collection and analysis, many processes still take a long time to complete. The primary role of machine learning in salary prediction is to facilitate these processes, saving time, effort, and money. Regression models, clustering techniques, and supervised learning algorithms are slowly gaining traction for predicting fair and competitive employee salaries.

One of the most important applications of machine learning in salary prediction is its role in analyzing multiple factors such as experience, skills, industry trends, and location. This also includes advanced techniques such as natural language processing for analyzing resumes and clustering methods to group employees with similar profiles, helping organizations determine appropriate salary ranges. For example, ML techniques involve supervised learning that predicts salaries by recognizing patterns in historical data. Developed by Microsoft, tools like Azure Machine Learning use ML-based technologies for several business initiatives, including resource planning and salary optimization.

ML applications in salary prediction also extend to identifying discrepancies and biases in pay structures that might otherwise go unnoticed. This can include anything from identifying gender pay gaps to analyzing regional differences. IBM Watson Analytics is an excellent example of how integrating machine learning with organizational data can aid in creating transparent and equitable salary frameworks.

**2.0 User Subscription Classification**

**1. Data Collection:**

Gather a dataset containing information about employees and their salaries. This dataset should ideally have features like years of experience, education level, job title, location, etc.

**2. Data Preprocessing:**

Clean the data by handling missing values, encoding categorical variables (like job titles or locations), and scaling numerical features.

**3. Feature Selection/Engineering:**

Identify the most relevant features that affect salary. You might want to engineer new features like creating dummy variables for categorical data or deriving new features from existing ones.

**4. Splitting the Data:**

Divide the dataset into training and testing sets. The training set is used to train the model, while the testing set evaluates its performance.

**5. Model Selection and Training:**

Choose a suitable machine learning algorithm for regression tasks. Algorithms like Linear Regression, Random Forest, Gradient Boosting, or Neural Networks are common choices. Train the model using the training dataset.

**6. Model Evaluation:**

Evaluate the model's performance on the testing dataset using metrics like Mean Absolute Error (MAE), Mean Squared Error (MSE), or R-squared score to understand how well the model predicts salaries.

**7. Hyperparameter Tuning (Optional):**

Optimize the model's performance by tuning its hyperparameters. This process involves adjusting parameters that are not learned by the model but affect its learning process.

**8. Prediction:**

Once the model is trained and evaluated, use it to predict salaries for new employee data.

**3.0 AI / ML Modelling and Results**

**3.1 Problem Statement –User Subscription Classification**

To predict whether a user will subscribe to a service based on various

factors such as age, gender, income, and past activity. The goal is to classify users into two

categories: those likely to subscribe and those who are unlikely to subscribe. By analyzing

these features, businesses can target the right customers with personalized offers and

services.

**1. Data Collection:**

Gather a dataset containing information about users, including features like:

User ID

Age

Gender

Country

Subscription\_Type

Income

Device

Last\_Activity

Days\_Since\_Last\_Activity

Will\_Subscribe

Any other relevant information

**2. Data Preprocessing:**

Clean the data by handling missing values, outliers, encoding categorical variables, and scaling numerical features if needed.

**3. Feature Engineering:**

Create new features if necessary. For instance, converting categorical data into numerical form using techniques like one-hot encoding or label encoding.

**4. Splitting Data:**

Divide the dataset into training and testing sets. Commonly, a 70-30 or 80-20 split is used.

**5. Choosing a Model:**

Select appropriate machine learning models for regression, considering algorithms like:

Linear Regression

Decision Trees

Random Forests

Gradient Boosting Regressors

Support Vector Machines (SVM)

Neural Networks (if dealing with a complex dataset)

**6. Training the Model:**

Train the chosen models on the training data.

**7. Model Evaluation:**

Evaluate the models using appropriate metrics like Accuracy Score, Confusion Matrix on the test dataset.

**8. Hyperparameter Tuning:**

Optimize the model performance by tuning hyperparameters using techniques like GridSearchCV or RandomizedSearchCV.

**9. Predictions:**

Use the trained model to predict salaries for new or unseen data.

**10. Model Deployment:**

Deploy the model into a production environment if needed, making sure it can handle real-time predictions.

**3.2 Data Science Project Life Cycle**

Data Science is a multidisciplinary field of study that combines programming skills, domain expertise and knowledge of the statistics and mathematics to extract useful insights and knowledge from data.

**3.2.1 Data Exploratory Analysis**

Exploratory data analysis has been done on the data to look for the relationship and correlation between different variables and to understand how they impact the target variable.

The exploratory analysis is done for the Base Salary Prediction with different parameters and all the charts are presented in Appendices 6.1 and 6.2.

**3.2.2 Data Pre-processing**

The variables which do not affect the target variable (Base Salary) as they add noise and also increase our computation time are removed. The data is checked for anomalous data points and outliers. Only the variables which directly affect the target variable are considered viable and are considered for prediction and analysis.

**3.2.2.1 Managing Duplicate and Low Variation Data**

The data set comprises of the Base Salary of outcomes of has no Duplicates since those are the records of individual and this has been verified using the appropriate code.

**3.2.2.2. Identifying and Addressing Missing Variables**

The missing values in this variable are handled by filling it with the average age at which a according to the records of annual health survey which is 24 years. Every other Variable Every other missing value in each of the columns are filled with the most frequently repeated value in its respective columns since they are all categorical data.

**3.2.2.3 Handling Outliers**

An outlier is a data point that is noticeably different from the rest. They represent errors in measurement, bad data collection, or simply show variables not considered when collecting the data. By applying the boxplot for every variable, it is observed that no outliers are present in this case.

**3.2.2.4 Categorical data and Encoding Techniques**

Categorical data is a collection of information that is divided into groups. This data is called categorical because it may be grouped according to the variables present in the collection of Information. Categorical data can take on numerical values (such as “1” indicating Yes and “2” indicating No), but those numbers don’t have mathematical meaning. One can neither add them together nor subtract them from each other.

Categorical data is of two types as following

**Nominal Data**

This is a type of data used to name variables without providing any numerical value

**Ordinal Data**

This is a data type with a set order or scale to it. However, this order does not have a standard scale on which the difference in variables in each scale is measured.

Encoding in categorical data involves converting the unique elements of the variables which belong to a set assigned an unique number. This results in alphabetically assigned values to each value in the variable.

LabelEncoding is one of the encoding techniques which arranges and replaces each value with their unique number.

In this project, at total 80 of 82 variables are categorical variables.

**3.2.2.5 Feature Scaling**

Feature scaling involves converting the variable values using scaling techniques by calculating the variance and ranging them from 0-1.

In this project “age” of employee and “years of experience” are the only columns out of all 82 columns which are continuous and are subjected to MinMaxScaler.

**3.2.3 Selection of Dependent and Independent Variables**

The dependent or target variable here is Base Salary which describes the birth of the baby as live birth or abortion based on all other variables.

The independent variables are selected after doing exploratory data analysis and we used Boruta to select which variables are most affecting our target variable.

**3.2.4 Data Balancing**

The data we have is completely well balanced and the major to minor class ratio is 4:1 which portrays that it is balanced and this is a good sign that the accuracy will be good in the models.

**3.2.5 Models Used for Development**

**3.2.5.1 Logistic Regression**

A statistical analysis method called logistic regression uses previous observations from a dataset to predict a binary outcome, like yes or no. By examining the correlation between one or more already present independent variables, a logistic regression model forecasts a dependent data variable. The mathematics for determining how many factors have an effect on a certain result are simplified by the use of logistic regression.

**3.2.5.2 Decision Tree Model**

A statistical analysis method called logistic regression uses previous observations from a dataset to predict a binary outcome, like yes or no. By examining the correlation between one or more already present independent variables, a logistic regression model forecasts a dependent data variable. The mathematics for determining how many factors have an effect on a certain result are simplified by the use of logistic regression.

**3.2.5.3 Random Forest Model**

A classification system made up of several decision trees is called the random forest. It attempts to produce an uncorrelated forest of trees whose forecast by committee is more accurate than that of any individual tree by using bagging and feature randomness when generating each individual tree. We require features with at least some predictive capability. The forest's trees, and more crucially, their forecasts, must not be connected (or at least have low correlations with each other).

**3.2.5.4 Extra Trees Classification Model**

Extremely Randomized Trees Classifiers (also known as Extra Trees Classifiers) are an ensemble learning technique that combine the output of various de-correlated decision trees gathered in a "forest" to produce a single classification result. The only way it differs conceptually from a Random Forest Classifier is in how the decision trees in the forest are built. The initial training sample is used to build each decision tree in the Extra Trees Forest. The optimal feature to divide the data according to some mathematical criteria (usually the Gini Index) must then be chosen by each decision tree from a random sampling of k features from the feature-set at each test node.

**3.2.5.5 K- Nearest Neighbors Model**

The k-nearest neighbors’ algorithm, sometimes referred to as KNN or k-NN, is a supervised learning classifier that employs proximity to produce classifications or predictions about the grouping of a single data point. Although it can be applied to classification or regression issues, it is commonly employed as a classification algorithm because it relies on the idea that comparable points can be discovered close to one another.

**3.2.5.6 Light Gradient Boosting Model**

LightGBM is a gradient boosting framework built on decision trees that improves model performance while using less memory. It employs two innovative techniques, Exclusive Feature Bundling (EFB), which overcomes the restrictions of the histogram-based algorithm that is largely employed in all GBDT (Gradient Boosting Decision Tree) frameworks, and Gradient-based One Side Sampling, which replaces them. Gradient-based One Side Sampling Technique for LightGBM, in which the information gain is computed using several data instances in different ways.

**3.2.5.7 Gaussian Naive Bayes Model**

A variation of Naive Bayes that supports continuous data and adheres to the Gaussian normal distribution is called Gaussian Naive Bayes. A class of supervised machine learning classification methods built on the Bayes theorem are known as naive bayes. Although it is a straightforward categorization method, it is highly functional. When the inputs are highly dimensional, they are useful. The Naive Bayes Classifier can be used to solve complex classification issues as well.

**3.2.5.8 Bagging classifier**

An ensemble meta-estimator called a bagging classifier fits base classifiers one at a time to random subsets of the original dataset, and then it aggregates the individual predictions (either by voting or by averaging) to provide a final prediction. By adding randomization to the process of building a black-box estimator (such a decision tree), a meta-estimator of this kind can often be used to lower the variance of the estimator

**3.2.5.9 Gradient Boosting Classifier**

A class of machine learning techniques known as gradient boosting classifiers combines a number of weak learning models to produce a powerful predicting model. Gradient boosting frequently makes use of decision trees. Due to their success in classifying complicated datasets, gradient boosting models are gaining popularity.

**3.3 AI / ML Models Analysis and Final Results**

The pre-processed dataset is split into two parts in the ration 80:20 in which the first part is test data and the second part is test data to check the accuracy and performance of the models.

I’ve used Confusion matrix to check the Accuracy, Precision, Recall and F1 Score of all models and compare and select the best model for given dataset.

**3.3.1 Combined Code for all Models**

#importing the libraries

import pandas as pd

import numpy as np

#importing plotting

from matplotlib import pyplot as plt

%matplotlib inline

import seaborn as sns

#ignore the harmless warnings

import warnings

warnings.filterwarnings('ignore')

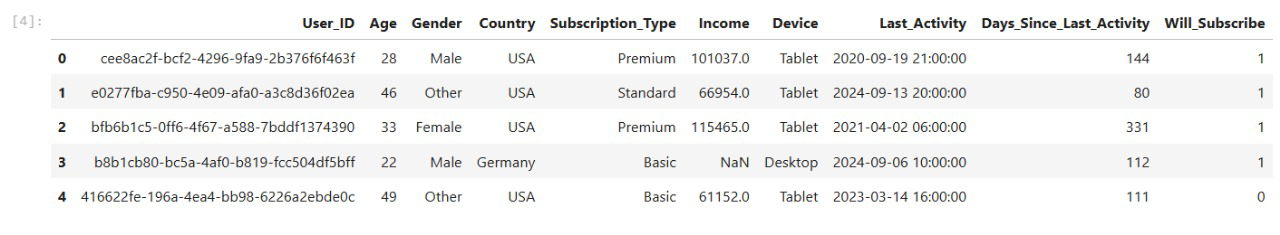
#set to display all cols in dataset

pd.set\_option('display.max\_columns',None)

# Load the pro dataset

user\_dataset=pd.read\_csv(r"C:\Users\swath\Desktop\ml\User\_Subscription\_Classification.csv")

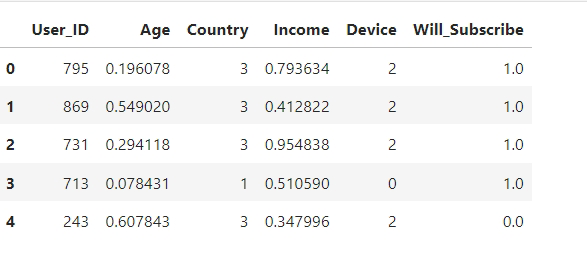
user\_dataset.head()



# Deleting Subscription\_type,Gender,Days\_Since\_LastActivity,Last\_Activity column as it has no influence on target variable

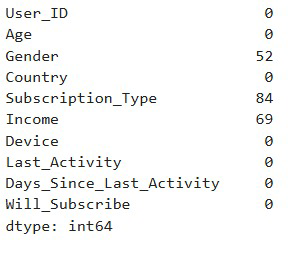
user\_dataset.drop(columns=['Gender','Subscription\_Type','Last\_Activity','Days\_Since\_Last\_Activity'],inplace=True)

user\_dataset.head()



# Checking for count of null values in the dataset in every column

user\_dataset.isnull().sum()

****

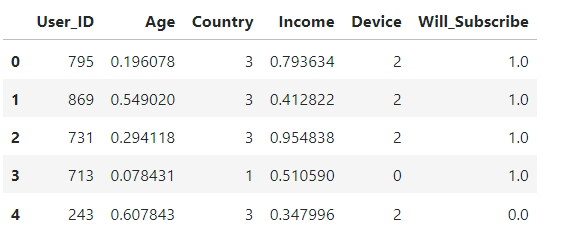
from sklearn.preprocessing import LabelEncoder

label\_encoder = LabelEncoder()

for column in categorical\_features:

    user\_dataset[column]= label\_encoder.fit\_transform( user\_dataset[column])

user\_dataset.head()

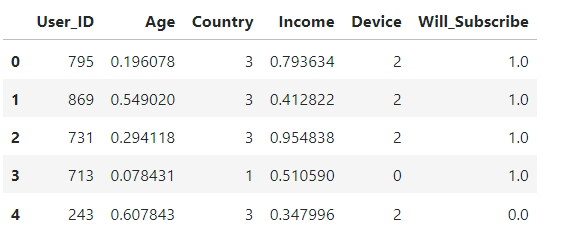


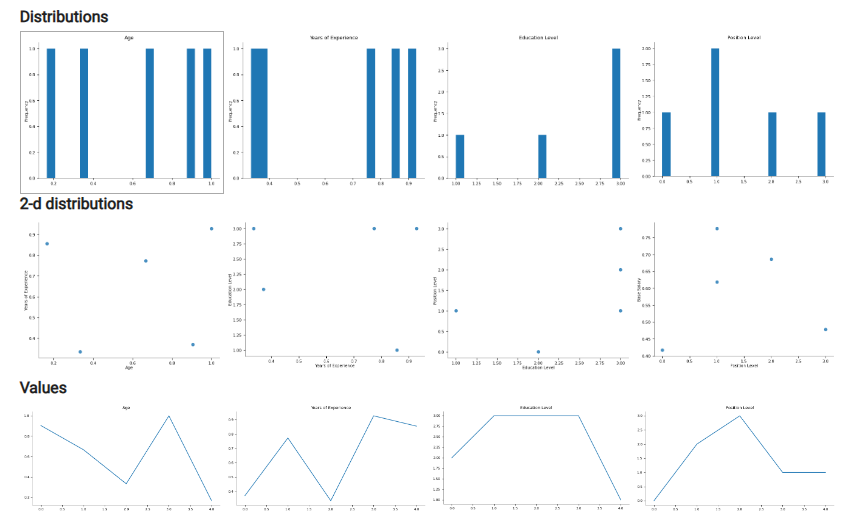
from sklearn.preprocessing import MinMaxScaler

scaler = MinMaxScaler()

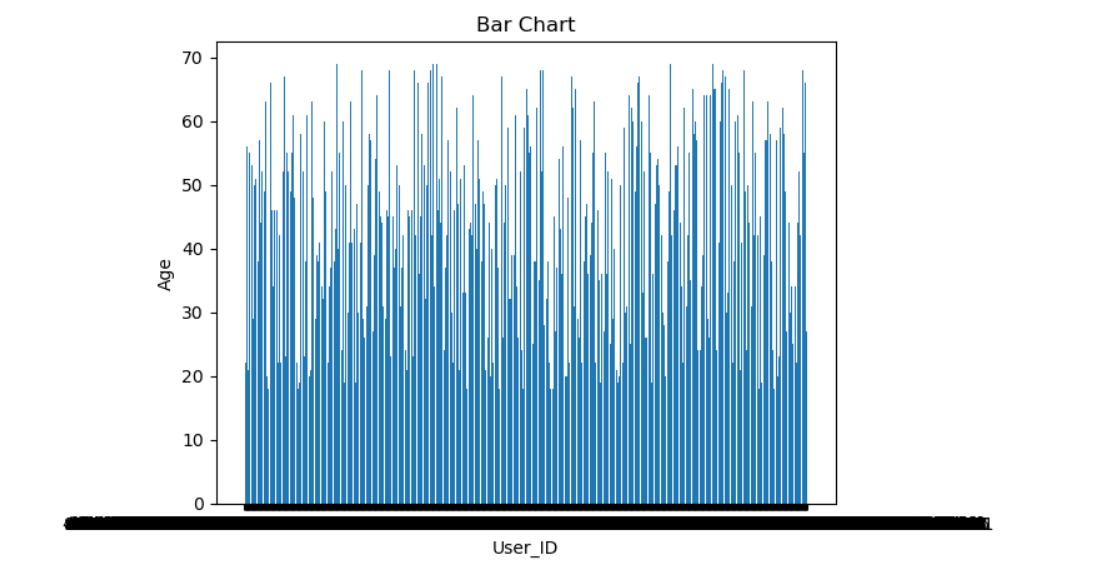
user\_dataset[continuous\_features] = scaler.fit\_transform(user\_dataset[continuous\_features])

user\_dataset.head()



****





X=user\_dataset.drop('Will\_Subscribe',axis=1)

y=user\_dataset['Will\_Subscribe']

]

#importing ML processes

from sklearn.linear\_model import LinearRegression

from sklearn.tree import DecisionTreeRegressor

from sklearn.ensemble import RandomForestRegressor

from sklearn.metrics import confusion\_matrix

from sklearn.metrics import classification\_report

# Splitting the dataset into train and test set

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)

from sklearn.linear\_model import LogisticRegression

logreg\_model = LogisticRegression( class\_weight='balanced')

logreg\_model.fit(X\_train, y\_train)

from sklearn.metrics import accuracy\_score, confusion\_matrix

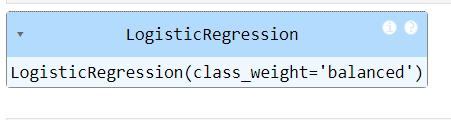
logreg\_pred = logreg\_model.predict(X\_test)

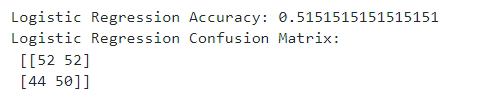
logreg\_accuracy = accuracy\_score(y\_test, logreg\_pred)

logreg\_conf\_matrix = confusion\_matrix(y\_test, logreg\_pred)

print("Logistic Regression Accuracy:", logreg\_accuracy)

print("Logistic Regression Confusion Matrix:\n", logreg\_conf\_matrix)





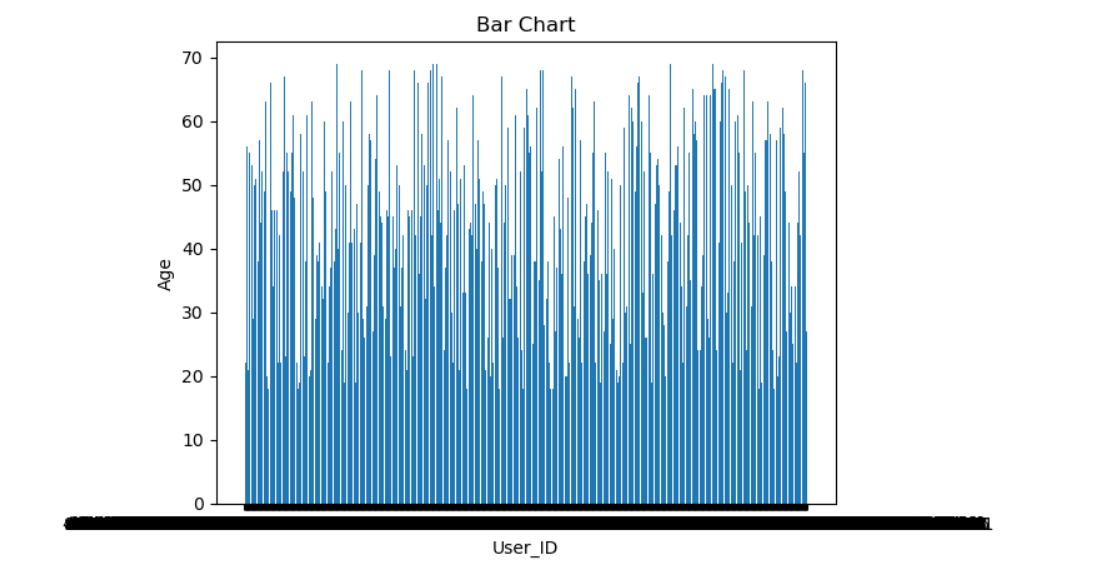
**5.0 References**

* Association Of Computer Engineers SRKR

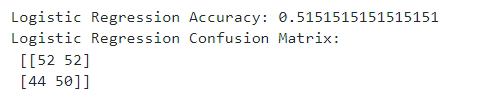
**6.0 Appendices**

**6.1 List of charts**

**Chart 01: 6.1.1 Bar Chart**

****

**6.2 Overall Results**

****

**Conclusion :**

By analysing ,

1.Logistic Regression

2.Random Forest Classifier

We concluded that “Logistic Regression” has more Accuracy than others which is 51%

So, we can use Logistic Regression model for “prediction” for getting good prediction