Tharuneswar J (PAD21378)
Nikita Shintre (PAD21237)
Raveesh M (PLA20277)
Rishabh Jain (PAD21282)

Credit Card Prediction

OVERVIEW

The project contains a dataset of 5000 points of data of 5000 unique individuals based on their personal details and their interaction with the bank. There are 14 different Variables that totally contribute to the dataset.

GOALS

1. To build a Machine Learning Model that uses the dataset to predict if a customer would buy Credit Card from their bank .
2. Use Classifier Models Such as K Nearest Neighbours and Random Forest as means to run the model.
3. Compare Different Models based on different combinations of features and come up with the most accurate Model.

CLASSIFIERS USED

- K Nearest neighbors
- Random Forest
- Logistic Regression

DATA DESCRIPTION

1. ID - Customer ID
2. Age - Customer's age in completed years
3. Experience - Years of professional experience
4. Income - Annual income of the customer ($000)
5. ZIPCode - Home Address ZIP code.
6. Family - The family size of the customer
7. CCAvg - Avg. spending on credit cards per month ($000)

8. Education - Education Level. 1: Undergrad; 2: Graduate; 3: Advanced/Professional
9. Mortgage - Value of house mortgage if any. ($000)
10. Personal Loan - Did this customer accept the personal loan offered in the last campaign?
11. Securities Account - Does the customer have a securities account with the bank?
12. CD Account - Does the customer have a certificate of deposit (CD) account with the bank?
13. Online - Does the customer use internet banking facilities?
14. CreditCard - Does the customer use a credit card issued by UniversalBank?

Data Visualization

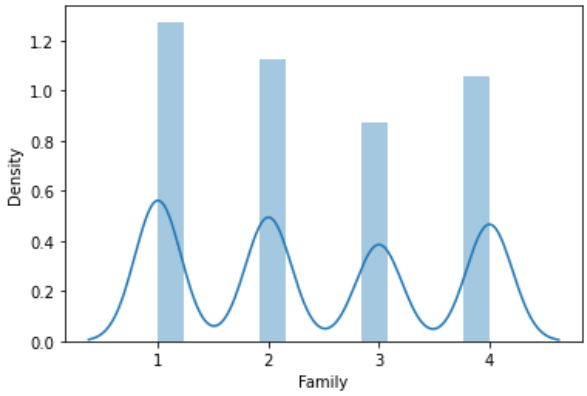
Correlation graph

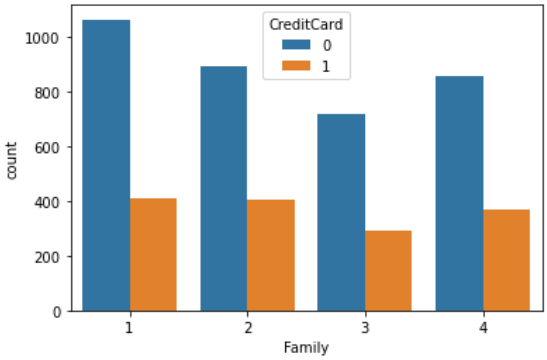
- This can be done by relating the Values of the Credit Card variable with the other values.
- Then the relationship between all the variables can be looked at with the help of a correlation graph and Heatmap.

Index	Age	Experience	Income	ZIP Code	Family	CCAvg	Education	Mortgage	Personal Loan	Securities Account	CD Account	Online	CreditCard
ID	-0.00847259	-0.00832576	-0.0176947	0.0134315	-0.0167972	-0.0246752	0.0214632	-0.0139199	-0.0248012	-0.0169723	-0.0069094	-0.00252841	0.0170282
Age	1	0.994215	-0.0552686	-0.0292163	-0.0464177	-0.0520122	0.0413344	-0.0125386	-0.00772562	-0.000436242	0.00804255	0.0137024	0.00768104
Experience	0.994215	1	-0.0465742	-0.0286255	-0.0525631	-0.0500765	0.0131518	-0.0105816	-0.0074131	-0.00123213	0.0103533	0.0138979	0.00896745
Income	-0.0552686	-0.0465742	1	-0.0164098	-0.157501	0.645984	-0.187524	0.206806	0.502462	-0.0026165	0.169738	0.0142059	-0.00238501
ZIP Code	-0.0292163	-0.0286255	-0.0164098	1	0.0117782	-0.00406068	-0.0173768	0.00738338	0.000107376	0.00470424	0.0199719	0.0169901	0.00769139
Family	-0.0464177	-0.0525631	-0.157501	0.0117782	1	-0.109275	0.0649289	-0.0204449	0.061367	0.0199941	0.0141104	0.010354	0.0115881
CCAvg	-0.0520122	-0.0500765	0.645984	-0.00406068	-0.109275	1	-0.136124	0.109905	0.366889	0.0150863	0.136534	-0.00361101	-0.00668949
Education	0.0413344	0.0131518	-0.187524	-0.0173768	0.0649289	-0.136124	1	-0.0333271	0.136722	-0.010812	0.0139339	-0.0150038	-0.0110141
Mortgage	-0.0125386	-0.0105816	0.206806	0.00738338	-0.0204449	0.109905	-0.0333271	1	0.142095	-0.00541097	0.0893111	-0.0059949	-0.00723092
Personal Loan	-0.00772562	-0.0074131	0.502462	0.000107376	0.061367	0.366889	0.136722	0.142095	1	0.0219539	0.316355	0.00627782	0.00280151
Securities Account	-0.000436242	-0.00123213	-0.0026165	0.00470424	0.0199941	0.0150863	-0.010812	-0.00541097	0.0219539	1	0.317034	0.0126275	-0.0150283
CD Account	0.00804255	0.0103533	0.169738	0.0199719	0.0141104	0.136534	0.0139339	0.0893111	0.316355	0.317034	1	0.17588	0.278644
Online	0.0137024	0.0138979	0.0142059	0.0169901	0.010354	-0.00361101	-0.0150038	-0.0059949	0.00627782	0.0126275	0.17588	1	0.00420966
CreditCard	0.00768104	0.00896745	-0.00238501	0.00769139	0.0115881	-0.00668949	-0.0110141	-0.00723092	0.00280151	-0.0150283	0.278644	0.00420966	1

Credit Card vs Other Variables

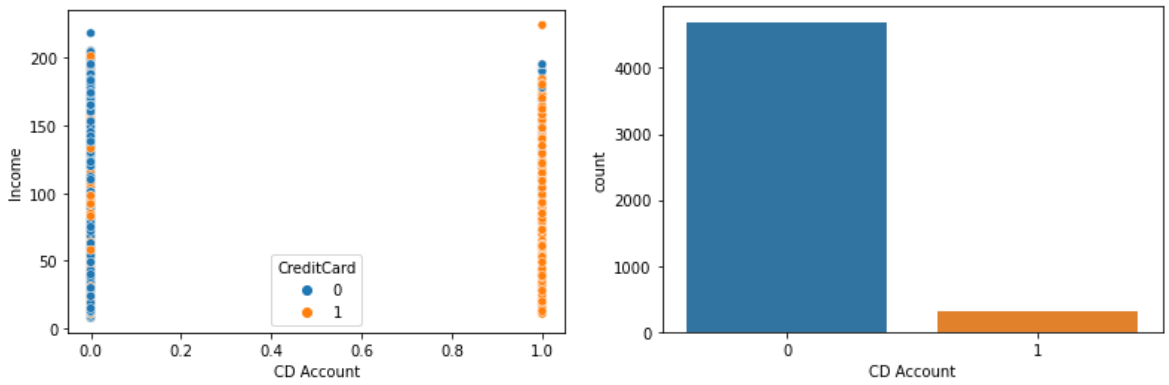
- Analysis of how the Credit card and other variables' values are related to each other variables proves to be an important tool in providing weight to each variable in the regression model.
- A scatterplot with Regression lines is a better way to understand the relationship between credit card values and the other variables.
- Representing them together makes it convenient to determine the weightage of each variable in the regression model.





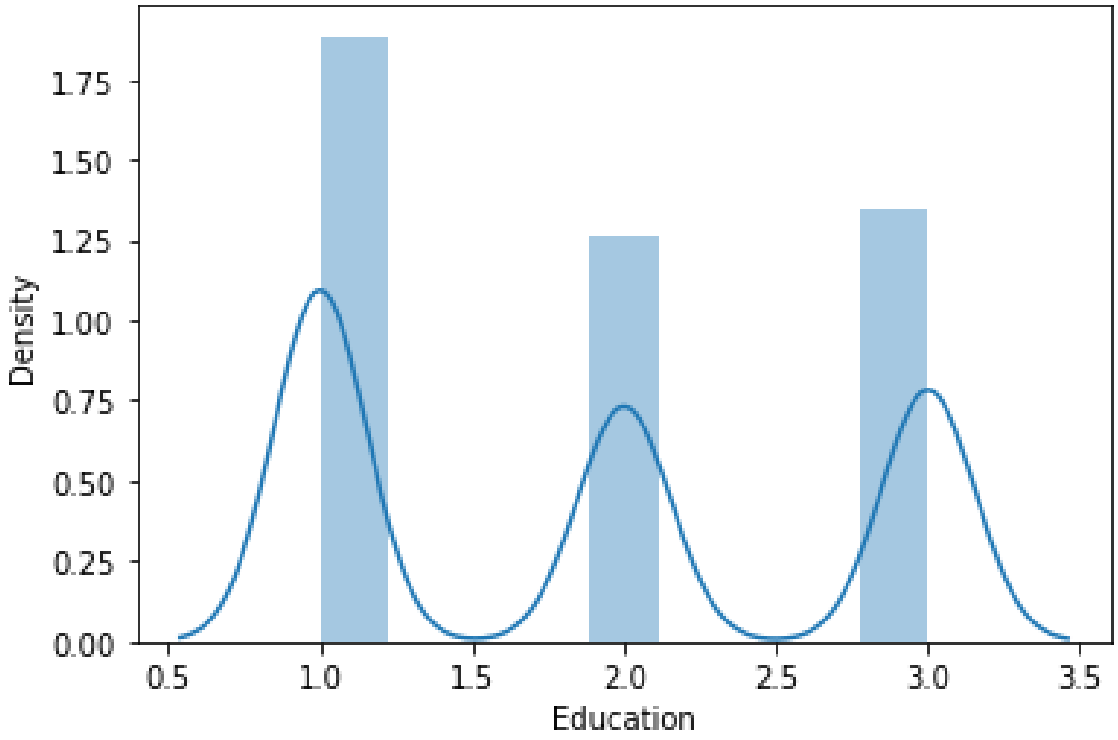
Family	Income
1	78.6223
2	84.2392
3	66.905
4	62.5131

The relationship between Credit Cards and **families** where the graphs don't show major differentiation justifies where family members do not affect in getting a credit card.



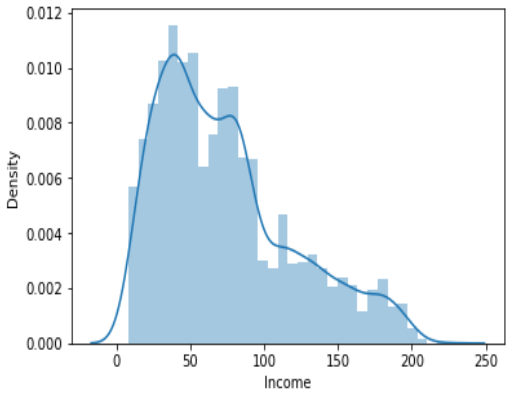
reditCar	CD Account
0	0.0175637
1	0.163265

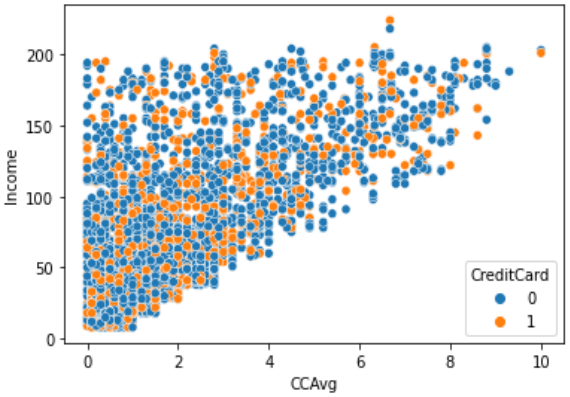
The relationship between Credit Card and a **CD account** states that the person who has a Certificate of Deposit from the bank finds it easiest to buy a credit card from the bank.



Educatio	Income	CCAvg
1	85.5864	2.26083
2	64.3136	1.68509
3	66.1226	1.72339

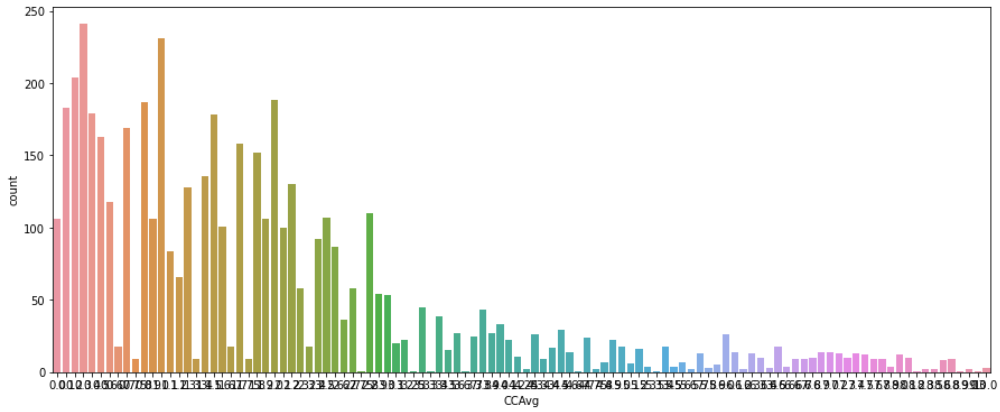
The relationship between **Education** and Credit cards states that the person with or without credit is similar and does not contribute to the possibility of the person getting a Credit Card

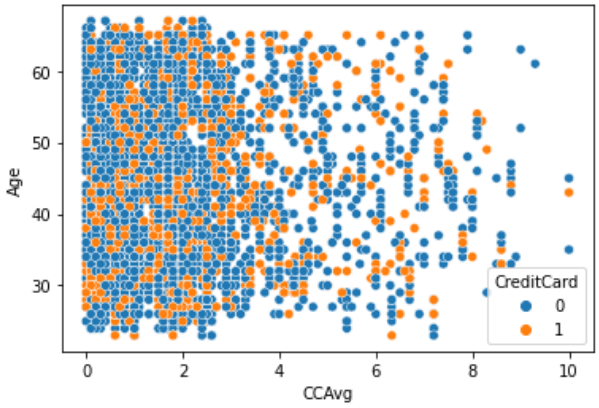




CreditCar	Income
0	73.845
1	73.6041

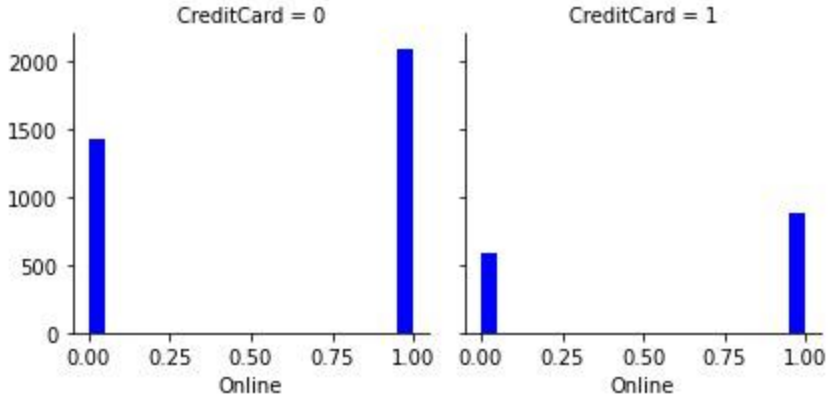
The graph on the right shows people with an **income** higher than 100$. This shows that lower-income people buy credit cards more



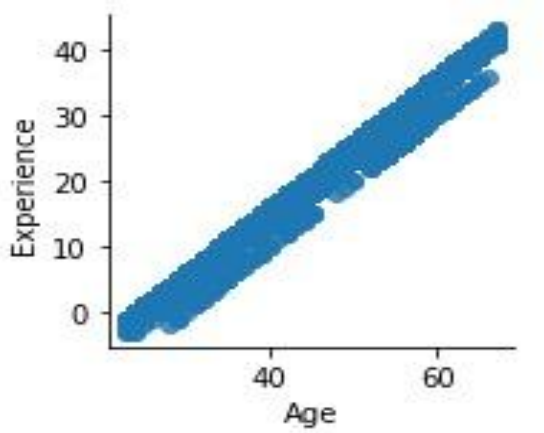


reditCar	CCAvg
0	1.94548
1	1.91982

The people who spend using credit cards are those who have **credit cards** already, thus making this column not useful.



People who use **internet banking** are more so this provides a positive impact on the purchase of credit cards.



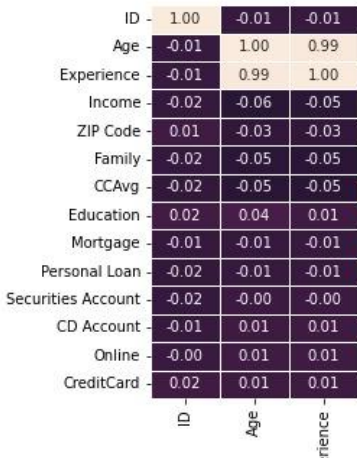
	ID	Age	Experience
ID	1.00	-0.01	-0.01
Age	-0.01	1.00	0.99
Experience	-0.01	0.99	1.00
Income	-0.02	-0.06	-0.05
ZIP Code	0.01	-0.03	-0.03
Family	-0.02	-0.05	-0.05
CCAvg	-0.02	-0.05	-0.05
Education	0.02	0.04	0.01
Mortgage	-0.01	-0.01	-0.01
Personal Loan	-0.02	-0.01	-0.01
Securities Account	-0.02	-0.00	-0.00
CD Account	-0.01	0.01	0.01
Online	-0.00	0.01	0.01
CreditCard	0.02	0.01	0.01

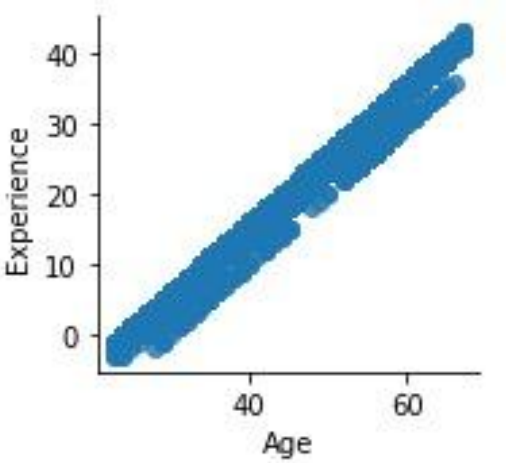
Index	Age	xperienc
count	5000	5000
mean	45.3384	20.3276
std	11.4632	11.253
min	23	0
25%	35	11
50%	45	20
75%	55	30
max	67	43

The graph on the above shows that people with the least **work experience** and the most experience tend to buy credit cards less.

The similarity in Age and Experience :

- Observing the Similarities between Age and Experience there is a similarity in their graphs when plotted with Credit card values.
- So only Experience is considered in this model for quicker computation and accuracy of the model.





Preparing Test and Train Values:

```
# Splitting the dataset into the Training set 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 = 0)

# Feature Scaling
from sklearn.preprocessing import StandardScaler
sc_X = StandardScaler()
X_train = sc_X.fit_transform(X_train)
X_test = sc_X.transform(X_test)
```

- The dataset is divided into test and train datasets with 80% of the entries being used to train the data and the rest 20% is used to test the data.
- The Values are then plotted as X and Y in which Y is the BMI dataset and X is a combination of the other variables in the dataset. X train and Y train values are used to train the model and the outcome is then plotted with Y test values to find the accuracy of the model.

Features in order of relation to Credit Card :

1. Securities account
2. Family
3. Education
4. Experience
5. Securities account
6. Mortgage
7. CC average
8. Online
9. Personal Loan
10. Income

Random Forest Classifier :

Fitting Training Data :

```
from sklearn.ensemble import RandomForestClassifier
Clf_RF =  RandomForestClassifier(max_depth=2, random_state=0)
Clf_RF.fit(X_train, y_train)
```

Prediction :

```
# Prediction
y_pred_RF = Clf_RF.predict(X_test)
```

K Nearest Neighbours Classifier:

Fitting Training Data :

```
from sklearn.neighbors import KNeighborsClassifier
Clf_KNN =  KNeighborsClassifier(n_neighbors=3)
Clf_KNN.fit(X_train, y_train)
```

Prediction :

```
# Prediction
y_pred_KNN = Clf_KNN.predict(X_test)
```

Logistic Regression:

Fitting Training Data :

```
from sklearn.linear_model import LogisticRegression
Clf_LR =  LogisticRegression(random_state=0)
Clf_LR.fit(X_train, y_train)
```

Prediction :

```
# Prediction
y_pred_LR = Clf_LRR.predict(X_test)
```

Model:

Features = All Features Except - ID, ZIP Code , Experience:

```
-------------------------
-------------------------
LOGISTIC REGRESSION
-------------------------
-------------------------
--------------------
| Confusion Matrix |
--------------------
[[697   7]
 [245  51]]
-------------------------
| Classifiction Report |
-------------------------
              precision    recall  f1-score   support

           0       0.74      0.99      0.85       704
           1       0.88      0.17      0.29       296

    accuracy                           0.75      1000
   macro avg       0.81      0.58      0.57      1000
weighted avg       0.78      0.75      0.68      1000

-------------------------
| Accuracy Score |
-------------------------
0.748
```

```
-------------------------
-------------------------
KNN CLASSIFIER
-------------------------
-------------------------
--------------------
| Confusion Matrix |
--------------------
[[697   7]
 [245  51]]
-------------------------
| Classifiction Report |
-------------------------
              precision    recall  f1-score   support

           0       0.74      0.99      0.85       704
           1       0.88      0.17      0.29       296

    accuracy                           0.75      1000
   macro avg       0.81      0.58      0.57      1000
weighted avg       0.78      0.75      0.68      1000

-------------------------
| Accuracy Score |
-------------------------
0.748
-------------------------
| Cross Validation Score |
-------------------------
0.6195999999999999
```

```
-------------------------
-------------------------
RANDOM FORST CLASSIFIER
-------------------------
-------------------------
--------------------
| Confusion Matrix |
--------------------
[[698   6]
 [247  49]]
-------------------------
| Classifiction Report |
-------------------------
              precision    recall  f1-score   support

           0       0.74      0.99      0.85       704
           1       0.89      0.17      0.28       296

    accuracy                           0.75      1000
   macro avg       0.81      0.58      0.56      1000
weighted avg       0.78      0.75      0.68      1000

-------------------------
| Accuracy Score |
-------------------------
0.747
-------------------------
| Cross Validation Score |
-------------------------
0.7414
```