NSE talent sprint

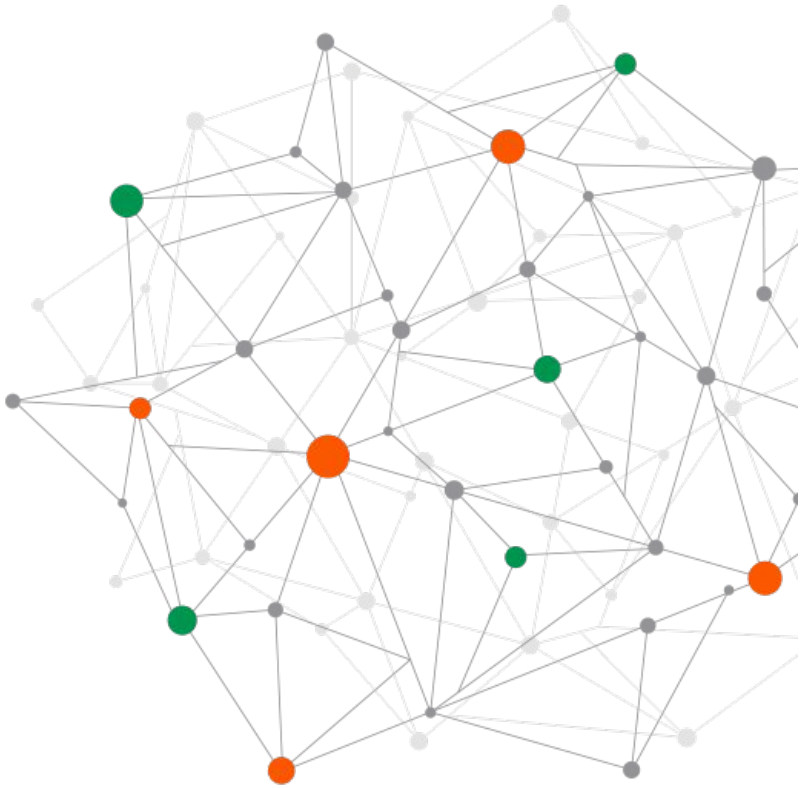
NPCI Cohort 4 Machine Learning Project

Adult Data Set

Presented By : Group 4

Bajjoori Chandu
Mehlam Taheri
Nalla Pavan Kumar Choudhury
Raj Abhishek Maurya
Ravi Jadam

27 October, 2022



NPCI
भारतीय राष्ट्रीय भुगतान निगम
NATIONAL PAYMENTS CORPORATION OF INDIA

Introduction

Problem Statement:

The objective of the dataset is to predict whether income exceeds $50K/yr based on census data. Also known as "Census Income" dataset.

Characteristics :

Data Set Characteristics:	Multivariate	**Number of Instances:**	48842	**Area:**	Social
Attribute Characteristics:	Categorical, Integer	**Number of Attributes:**	14	**Date Donated**	1996-05-01
Associated Tasks:	Classification	**Missing Values?**	Yes	**Number of Web Hits:**	1403336

Approach

Step 1	Understanding the Dataset.
Step 2	Loading Libraries and Dataset.
Step 3	Descriptive Analysis.
Step 4	Exploratory Data Analysis
Step 5	Data Preprocessing.
Step 6	Data Modeling and Evaluation.

Loading Libraries and Dataset.

The dataset contains the labels which we have to predict and the labels are discrete and binary. So the problem we have is a Supervised Classification type.

```
1 # Import libraries
2 import pandas as pd
3 import numpy as np
4 import matplotlib.pyplot as plt
5 import seaborn as sns
6
7 import warnings
8 warnings.filterwarnings('ignore')
```

```
1 # Importing dataset
2 dataset = pd.read_csv('/content/adult.csv')
```

```
1 dataset.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 32561 entries, 0 to 32560
Data columns (total 15 columns):
 #   Column          Non-Null Count  Dtype
---  ------          --------------  -----
 0   age             32561 non-null  int64
 1   workclass       32561 non-null  object
 2   fnlwgt          32561 non-null  int64
 3   education       32561 non-null  object
 4   education.num   32561 non-null  int64
 5   marital.status  32561 non-null  object
 6   occupation      32561 non-null  object
 7   relationship    32561 non-null  object
 8   race            32561 non-null  object
 9   sex             32561 non-null  object
 10  capital.gain    32561 non-null  int64
 11  capital.loss    32561 non-null  int64
 12  hours.per.week  32561 non-null  int64
 13  native.country  32561 non-null  object
 14  income          32561 non-null  int64
dtypes: int64(7), object(8)
memory usage: 3.7+ MB
```

Descriptive Analysis

Checking for NULL values and values with '?'

```
[8]  1 # Check for null values
     2 round((dataset.isnull().sum() /
```

```
age               0.0 %
workclass         0.0 %
fnlwgt            0.0 %
education         0.0 %
education.num     0.0 %
marital.status    0.0 %
occupation        0.0 %
relationship      0.0 %
race              0.0 %
sex               0.0 %
capital.gain      0.0 %
capital.loss      0.0 %
hours.per.week    0.0 %
native.country    0.0 %
income            0.0 %
dtype: object
```

```
[9]  1 # Check for '?' in dataset
     2 round((dataset.isin(['?']).sum()
```

```
age               0.0 %
workclass         5.639 %
fnlwgt            0.0 %
education         0.0 %
education.num     0.0 %
marital.status    0.0 %
occupation        5.66 %
relationship      0.0 %
race              0.0 %
sex               0.0 %
capital.gain      0.0 %
capital.loss      0.0 %
hours.per.week    0.0 %
native.country    1.79 %
income            0.0 %
dtype: object
```

Descriptive Analysis

Checking the counts of label (Income) categories.

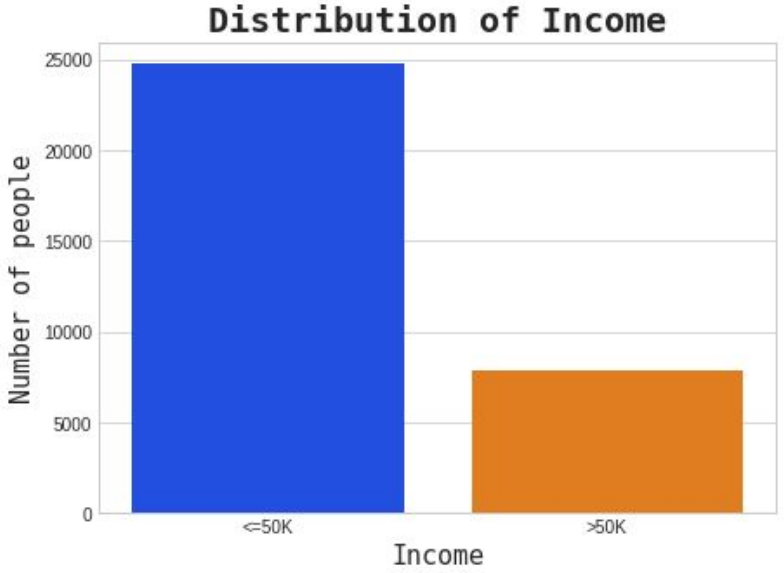
```
[10]  1 # Checking the counts of label(income) categories
      2 income = dataset['income'].value_counts(normalize=True)
      3 round(income * 100, 3).astype('str') + ' %'
```

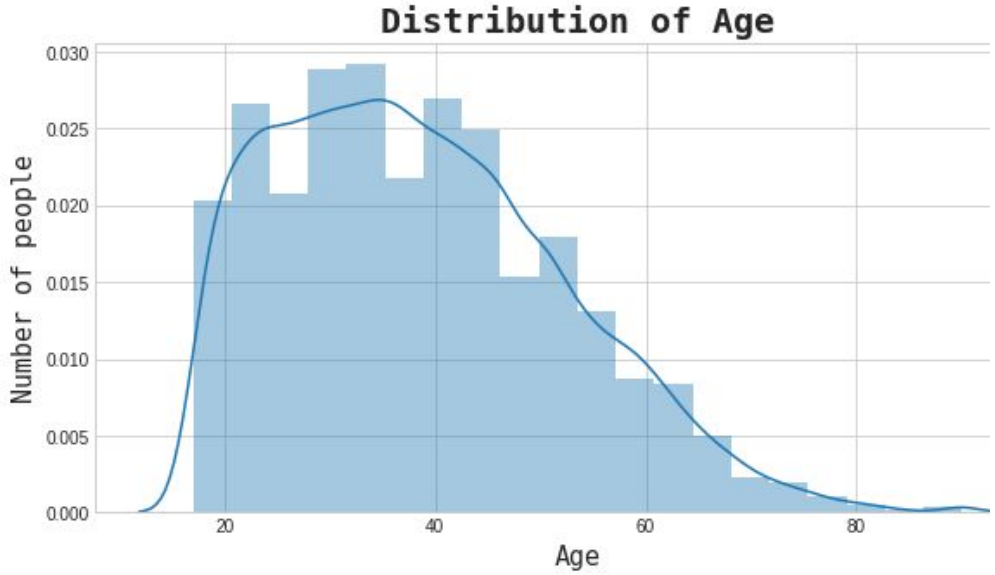
```
<=50K     75.919 %
>50K      24.081 %
Name: income, dtype: object
```

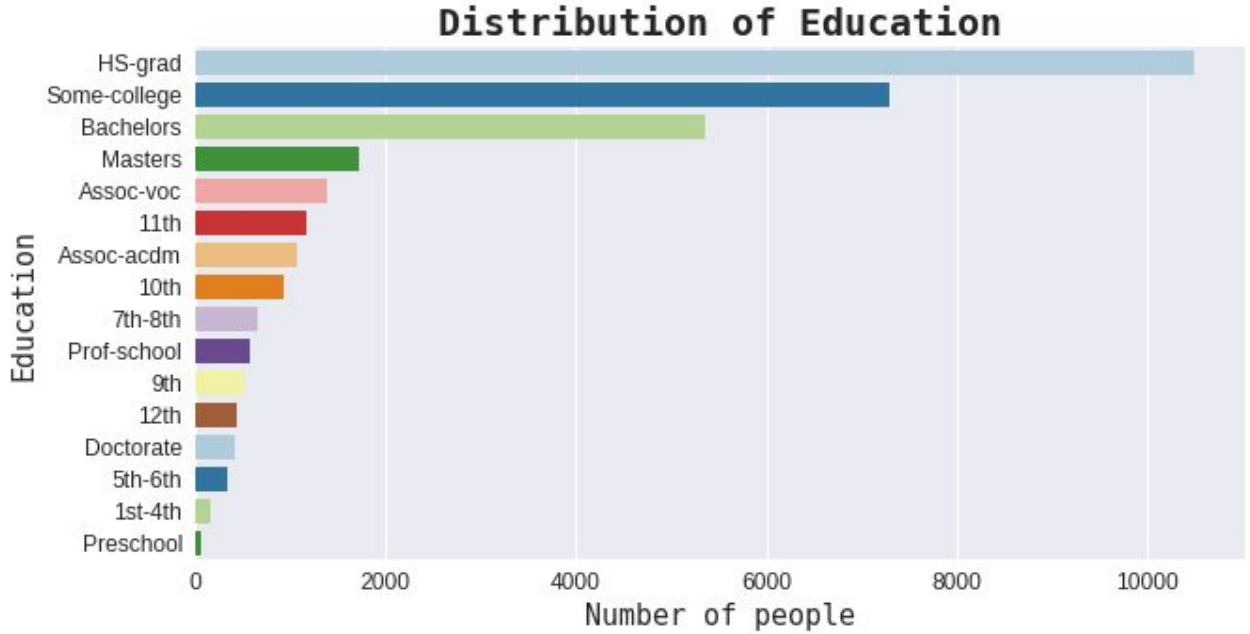
OBSERVATIONS :

1. Dataset does not have any NULL Values, but it contain missing values in form of '?'
2. The dataset is unbalanced, as the dependent feature 'income' contains 75.92% values have income less than 50k$ and 24.08% have income more than 50k$

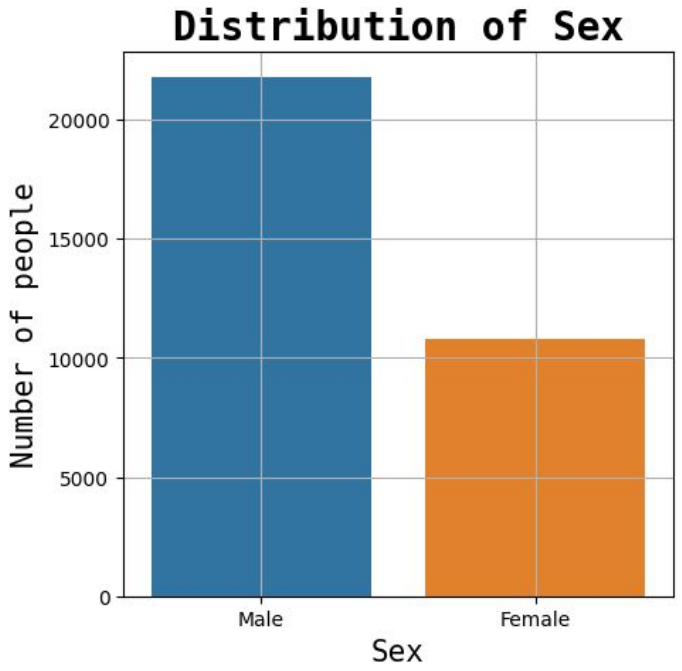
Exploratory Data Analysis : Univariate Analysis

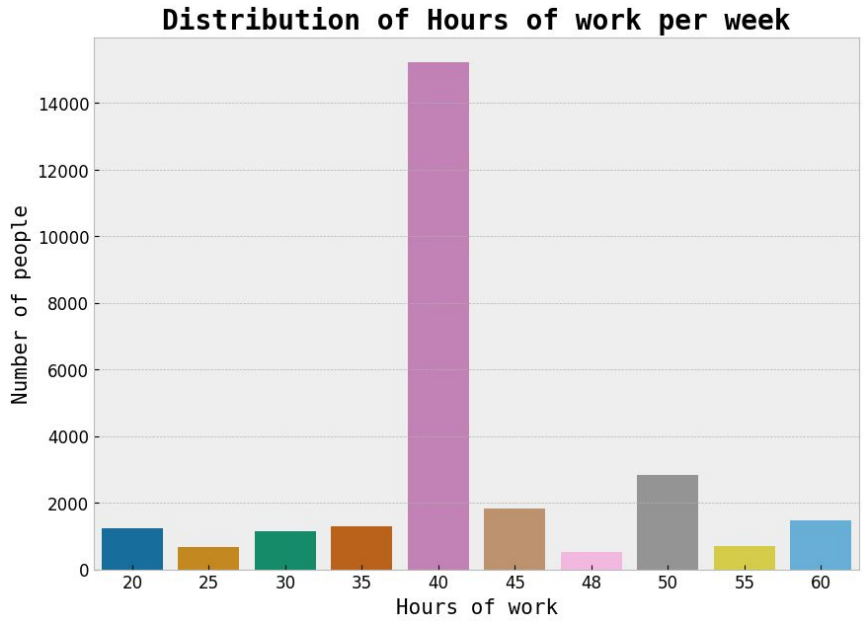


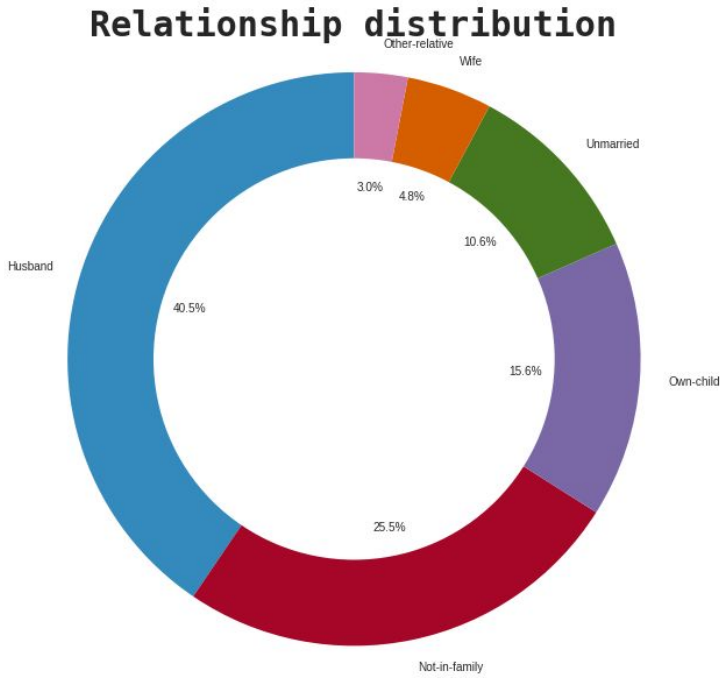




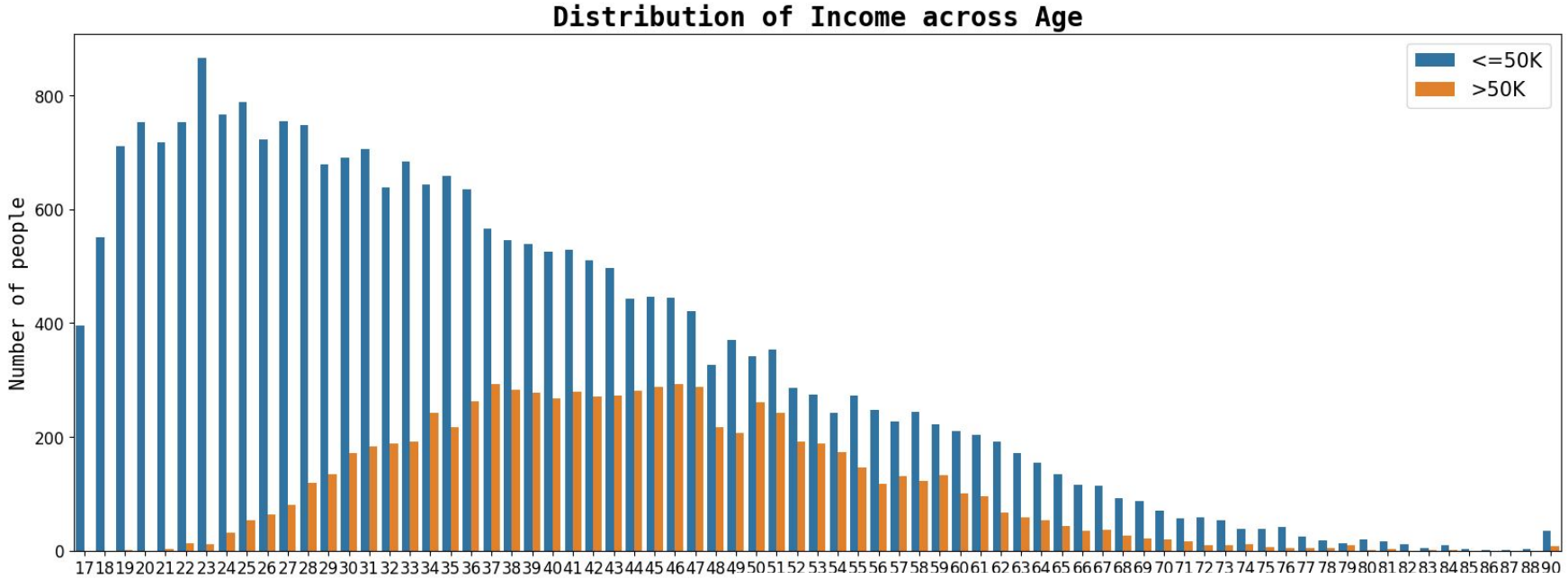
Exploratory Data Analysis : Univariate Analysis

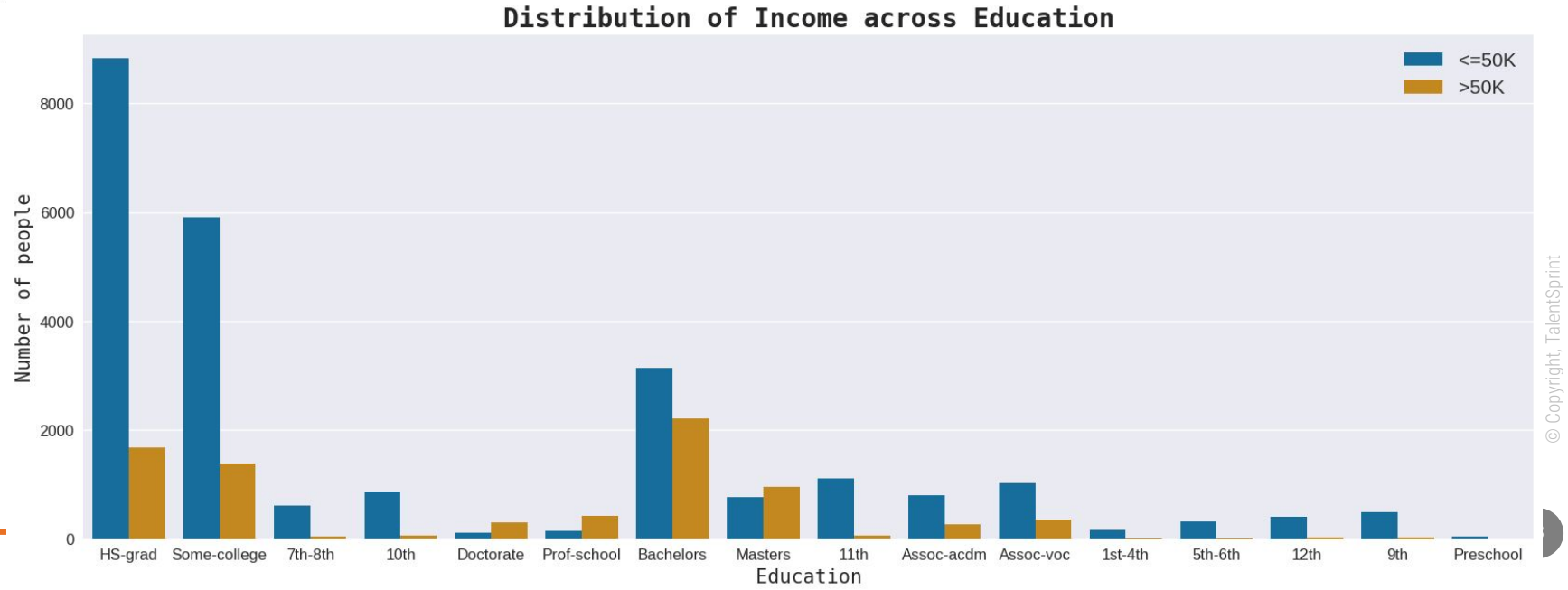


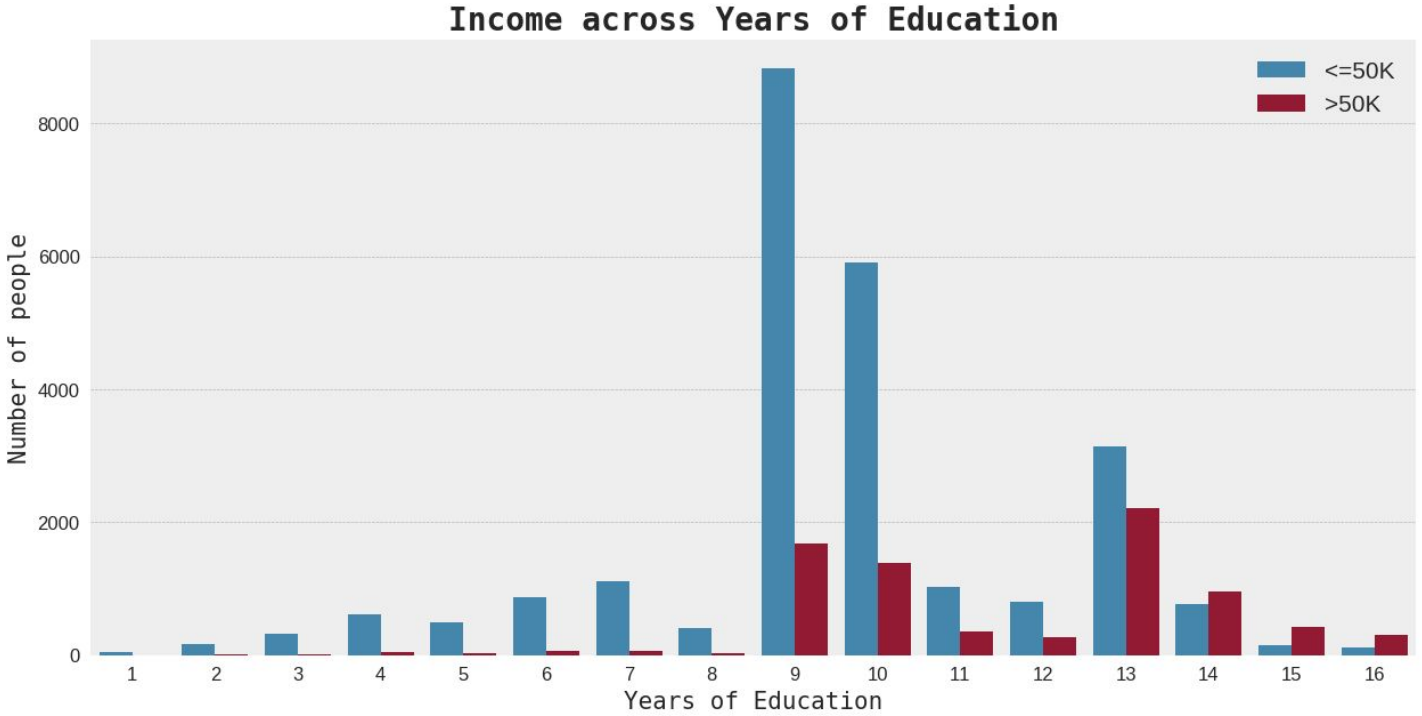


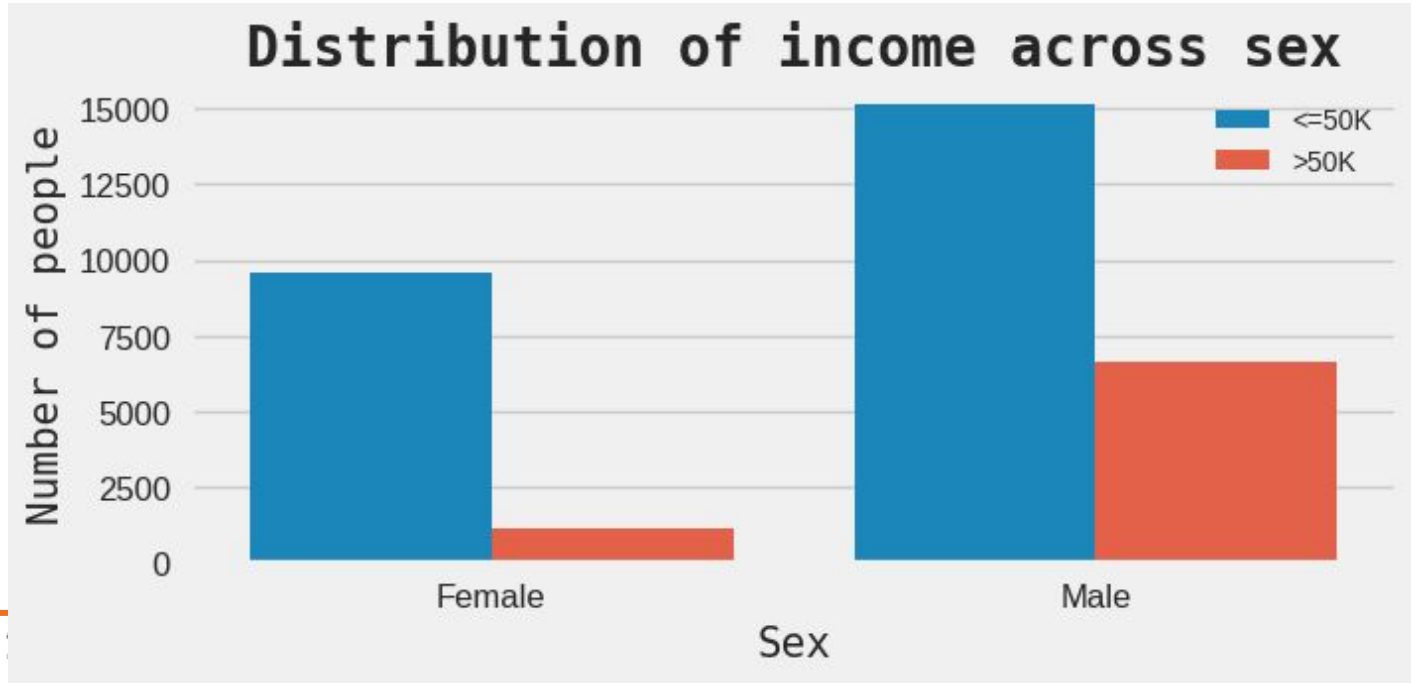


Exploratory Data Analysis : Bivariate Analysis





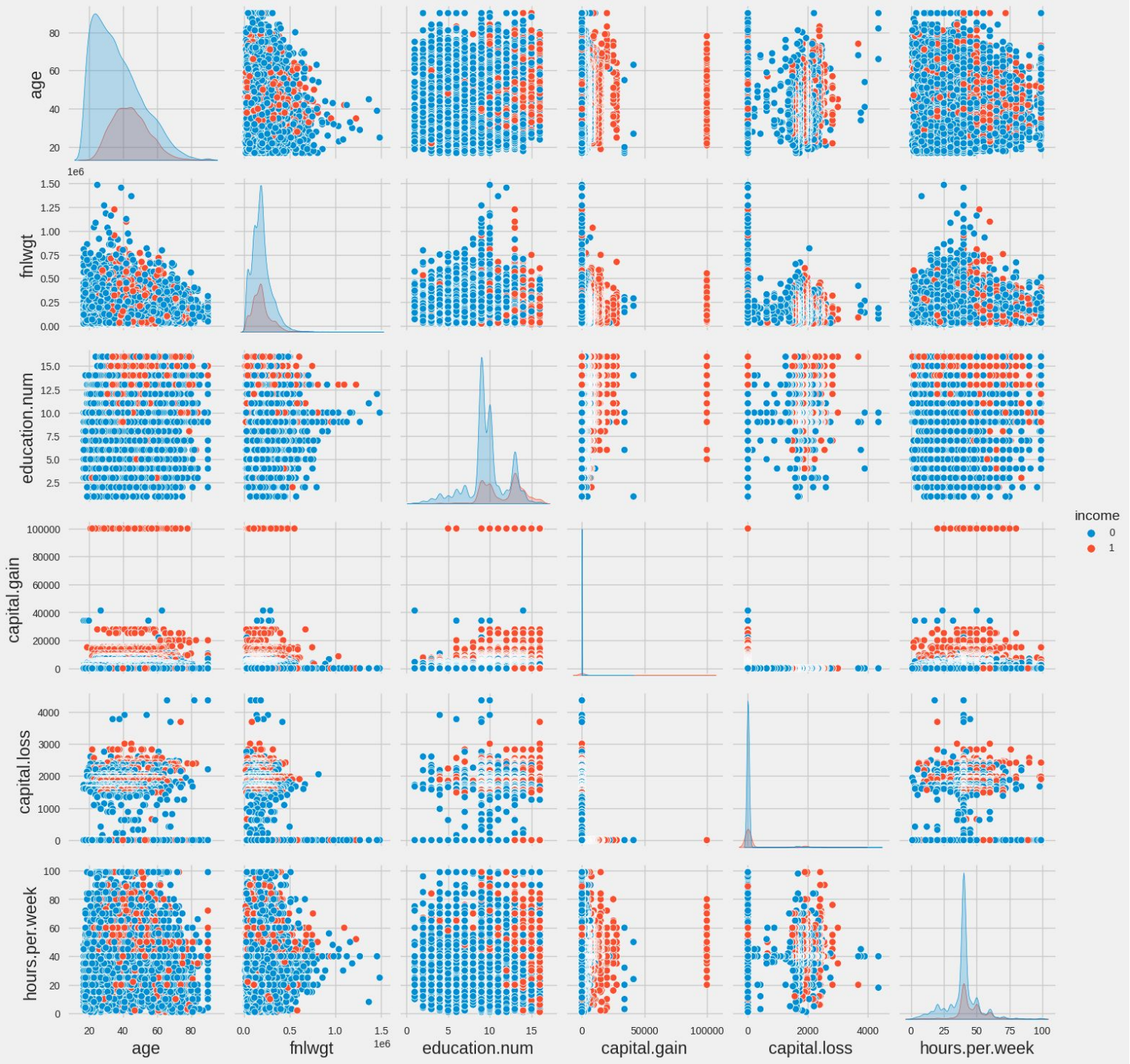


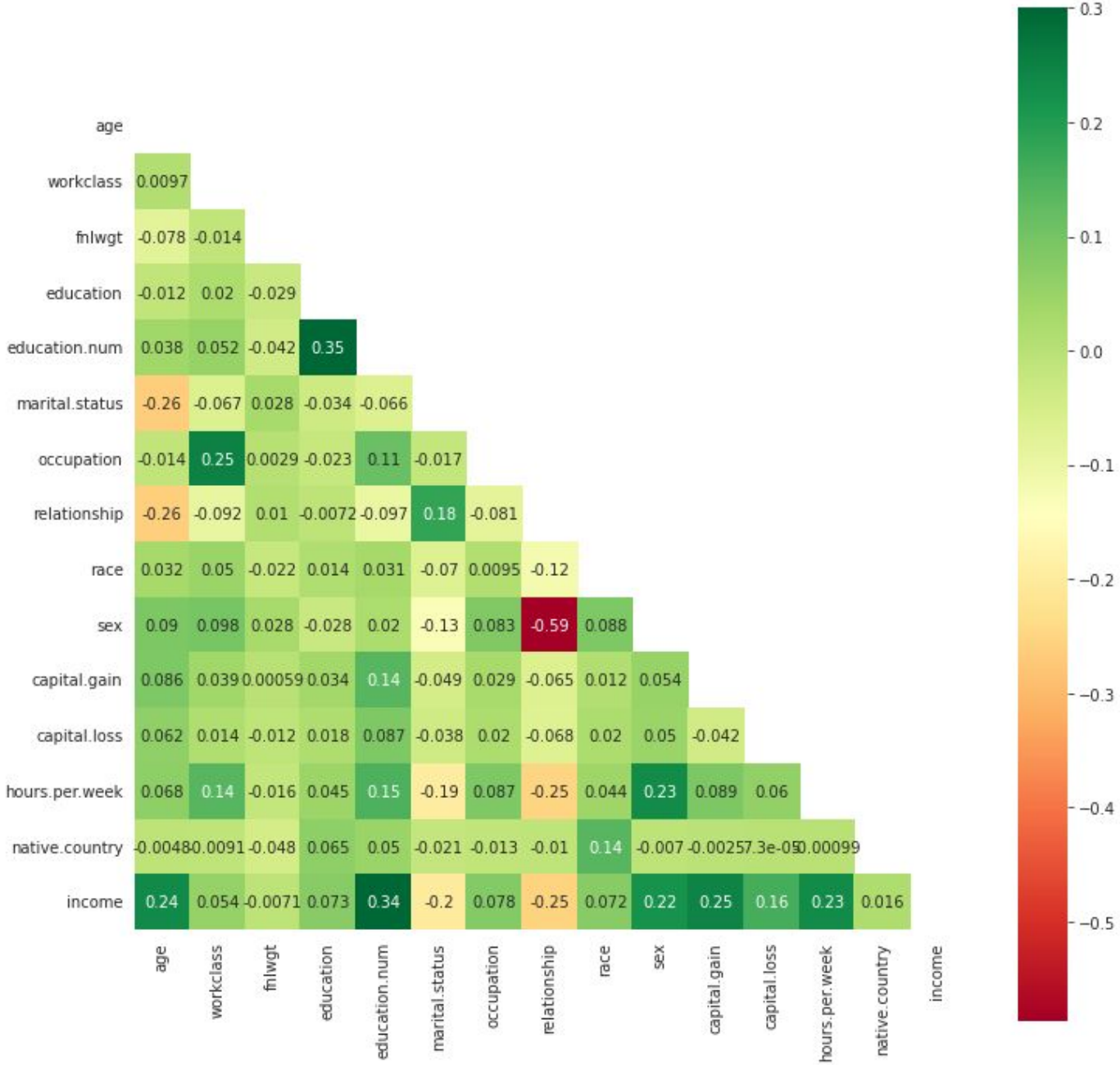


Exploratory Data Analysis :

Multivariate Analysis

Creating a Pairplot of Dataset





Exploratory Data Analysis : Multivariate Analysis

Correlation Heatmap.

OBSERVATIONS :

1. In this dataset most no. of people are young, male, high school graduates with 9 to 10 years of education and work 40 hours per week.
2. From the correlation heatmap we can see that dependent feature ‘income’ is highly correlated with age, numbers of year of education, capital gain and no. of hours per week.

Data Preprocessing

```
[ ]  1 dataset = dataset.replace('?', np.nan)
```

```
[ ]  1 # Checking null values
     2 round((dataset.isnull().sum() / dataset.shape[0]) * 100, 2).astype(str) + ' %'
```

```
age               0.0 %
workclass        5.64 %
fnlwgt            0.0 %
education         0.0 %
education.num     0.0 %
marital.status    0.0 %
occupation       5.66 %
relationship      0.0 %
race              0.0 %
sex               0.0 %
capital.gain      0.0 %
capital.loss      0.0 %
hours.per.week    0.0 %
native.country   1.79 %
income            0.0 %
dtype: object
```

```
[ ]  1 columns_with_nan = ['workclass', 'occupation', 'native.country']
```

```
[ ]  1 for col in columns_with_nan:
     2     dataset[col].fillna(dataset[col].mode()[0], inplace=True)
```

Fixing '?' values in the dataset.

Rechecking for NULL Values

Replacing NULL values with Mode value of the particular column.

Data Preprocessing

```
[ ]  1 for col in dataset.columns:
     2     if dataset[col].dtypes == 'object':
     3         encoder = LabelEncoder()
     4         dataset[col] = encoder.fit_transform(dataset[col])
```

```
[ ]  1 dataset.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 32561 entries, 0 to 32560
Data columns (total 15 columns):
 #   Column          Non-Null Count  Dtype
---  ------          --------------  -----
 0   age             32561 non-null  int64
 1   workclass       32561 non-null  int64
 2   fnlwgt          32561 non-null  int64
 3   education       32561 non-null  int64
 4   education.num   32561 non-null  int64
 5   marital.status  32561 non-null  int64
 6   occupation      32561 non-null  int64
 7   relationship    32561 non-null  int64
 8   race            32561 non-null  int64
 9   sex             32561 non-null  int64
 10  capital.gain    32561 non-null  int64
 11  capital.loss    32561 non-null  int64
 12  hours.per.week  32561 non-null  int64
 13  native.country  32561 non-null  int64
 14  income          32561 non-null  int64
dtypes: int64(15)
```

Label Encoding

Data Modelling

4.1: Logistic Regression

```
[ ]  1 from sklearn.linear_model import LogisticRegression
     2 log_reg = LogisticRegression(random_state=42)
```

```
[ ]  1 log_reg.fit(X_train, Y_train)
```

```
LogisticRegression(random_state=42)
```

```
[ ]  1 Y_pred_log_reg = log_reg.predict(X_test)
```

4.2: KNN Classifier

```
[ ]  1 from sklearn.neighbors import KNeighborsClassifier
     2 knn = KNeighborsClassifier()
```

```
[ ]  1 knn.fit(X_train, Y_train)
```

```
KNeighborsClassifier()
```

```
[ ]  1 Y_pred_knn = knn.predict(X_test)
```

4.5: Decision Tree Classifier

```
     1 from sklearn.tree import DecisionTreeClassifier
     2 dec_tree = DecisionTreeClassifier(random_state=42)
```

```
[ ]  1 dec_tree.fit(X_train, Y_train)
```

```
DecisionTreeClassifier(random_state=42)
```

```
[ ]  1 Y_pred_dec_tree = dec_tree.predict(X_test)
```

4.6: Random Forest Classifier

```
[ ]  1 from sklearn.ensemble import RandomForestClassifier
     2 ran_for = RandomForestClassifier(random_state=42)
```

```
[ ]  1 ran_for.fit(X_train, Y_train)
```

```
RandomForestClassifier(random_state=42)
```

```
[ ]  1 Y_pred_ran_for = ran_for.predict(X_test)
```

```
print('Logistic Regression:')
print('Accuracy score:', round(accuracy_score(Y_test, Y_pred_log_reg) * 100, 2))
print('F1 score:', round(f1_score(Y_test, Y_pred_log_reg) * 100, 2))
```

```
Logistic Regression:
Accuracy score: 68.74
F1 score: 66.28
```

```
print('KNN Classifier:')
print('Accuracy score:', round(accuracy_score(Y_test, Y_pred_knn) * 100, 2))
print('F1 score:', round(f1_score(Y_test, Y_pred_knn) * 100, 2))
```

```
KNN Classifier:
Accuracy score: 73.13
F1 score: 75.13
```

```
print('Decision Tree Classifier:')
print('Accuracy score:', round(accuracy_score(Y_test, Y_pred_dec_tree) * 100, 2))
print('F1 score:', round(f1_score(Y_test, Y_pred_dec_tree) * 100, 2))
```

```
Decision Tree Classifier:
Accuracy score: 91.59
F1 score: 92.01
```

```
print('Random Forest Classifier:')
print('Accuracy score:', round(accuracy_score(Y_test, Y_pred_ran_for) * 100, 2))
print('F1 score:', round(f1_score(Y_test, Y_pred_ran_for) * 100, 2))
```

```
Random Forest Classifier:
Accuracy score: 93.25
F1 score: 93.53
```

Model Evaluation

Conclusion :

1. In this project we build various models like Logistic Regression, KNN Classifier, Decision Tree Classifier and Random Forest Classifier
2. Random Forest Classifier gives the highest accuracy score of 93.25 and f1 score of 93.53

Challenges

Selecting the appropriate column names.

Exploration of each column.

Understanding multivariate analysis with target columns.

Statistics

Number of Lines of Code

220

Miscellaneous

Libraries Used :

1. Pandas
2. Numpy
3. Matplotlib
4. Seaborn
5. Scikit- learn

Algorithms Used :

1. Logistic Regression
2. KNN Classifier
3. Decision Tree Classifier
4. Random Forest Classifier

NSE talent sprint

NPCI
भारतीय राष्ट्रीय भुगतान निगम
NATIONAL PAYMENTS CORPORATION OF INDIA