Income Classification

Project-Based Internship 2020 Report

Submitted

To

DataRitz Technologies

Duration : Six Weeks

By

Mukund Rastogi
1803210098

Udit Gupta
1803210166

Shatmanyu Gupta
1803210135

ABES Engineering College

APJ Abdul Kalam Technical University (AKTU)

Under the guidance of
Mr. Gopal Gupta
Mr. Shashank Shekhar

TABLE OF CONTENTS

DataRitz Technologies
Enhancing Technology Experience

CERTIFICATE

This is to certify that Project Report entitled “Income Classification” which is submitted by **Mukund Rastogi, Udit Gupta** and **Shatmanyu Gupta** in partial fulfillment of the requirement for the summer internship of **B.Tech** in Department of **CSE** of **ABES Engineering College**, is a record of the candidate own work carried out by him under my/our supervision.

Mr. Gopal Gupta

Mr. Shashank Shekhar

Date: 08/07/2020

ACKNOWLEDGEMENT

It gives us a great sense of pleasure to present the report of the Project Based Internship 2020 undertaken during B.Tech 2nd Year. We owe special debt of gratitude to Mr.Gopal Gupta and Mr. Shashank Shekar, DataRitz Technologies for their constant support and guidance throughout the course of our work. Their constant motivation has been a constant source of inspiration for us. It is only their cognizant efforts that our endeavors have seen light of the day.

We also take the opportunity to acknowledge the contribution of team members of DataRitz Technologies for their full support and assistance during the development of the project.

We also do not like to miss the opportunity to acknowledge the motivation of CSE department of ABES Engineering College to provide us the opportunity to undergo training at DataRitz Technologies.

Name :	*Mukund Rastogi ,*	*Udit Gupta,*	*Shatmanyu Gupta*
Signature:	Mukund	Udit	Shatmanyu
Roll No.:	*1803210098,*	*1803210166,*	*1803210135*

Date : 08 - July - 2020

ABSTRACT

In this project, we have implemented a machine learning model on a real dataset taken from the UCI repository. It is the 1995 US Census dataset which has 14 attributes based on which our machine learning model will work. We are expected to gain experience using a common data-mining and machine learning library.

In this project report, we have a summary of our analysis and exploration of the Adult Census Income Data and we have come up with different and interesting attributes which were present in the dataset. We have visualised the interdependence of the attributes with each other and grasped knowledge about the same. In the end we have performed a predictive task of classification on the income of any individual whether it exceeds a certain amount (50k) per annum or not.

Project Summary

Region/Unit	DataRitz Technologies
Location	Ghaziabad
Program	DataRitz Technologies.<<programcode>>.<<version>>
Project Number	DataRitz Technologies. <<projectcode>>.<<version>>
Project Description	Income Classification using Machine Learning Models

Document Control

Prepared by:	Mukund Rastogi, Udit Gupta, Shatmanyu Gupta
Title:	Income Classification using Machine Learning
College:	ABES Engineering College
Department:	CSE
Location:	Ghaziabad
Version date:	<<date of document completion>>
Status:	**<<Initial Draft**/Consultation Draft/Approved Document/Minor Revision/Major Revision>>

Version history

Version no.	Date	Changed by	Nature of amendment
1.0			
1.1			

Endorsement and Approval

Project Customer

I approve the business requirements specifications in this document.

Name	<<customer name>>		
Position	<<customer position>>		
Signature		Date	

The following officers have **endorsed** this document

Project Sponsor

Name	<<sponsor name>>		
Position	<<sponsor position>>		
Signature		Date	

Project Manager (= Component Project Customer)

Name	Mr. Gopal Gupta / Mr. Shashank Shekhar		
Position	Lead Technical Architect / Project Consultant		
Signature		Date	

Component Project Sponsor

I accept the business requirements specifications in this document.

Name	Dr B P Sharma		
Position	Country Head – Delivery		
Signature		Date	
Comments			

The following officers have **endorsed** this document

Component Program Manager

Name	Mr. Gaurav Kansal		
Position	Chief Operating Officer		
Signature		Date	

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

1.1 : Problem Definition

To classify income of the society in two categories namely above 50k and below 50k through machine learning using the given demographic factors.

The prime goal is to make a machine learning model or to train a binary classifier to predict income of an individual with the help of the given attributes in the primary dataset and to yield the best possible result.

1.2 Motivation

There were several motivation factors for picking this project which are listed as below :

- The world's serging level of economic inequality is the main motivation of making this model which is able to deal with the classification among the society and hence make aware of the gap existing in the society for further upliftment through taking the necessary measures.

- For improving the living standards of the middle class in a developing country like India. It is evident the middle class usually has access to all the factors which are necessary for the upliftment in their living standards i.e. income but they are usually unaware of it.

- Get an estimation of income classification after the pandemic outbreak of coronavirus and get the suggestive approach in upliftment.

- In improving the decision making of students who are stuck in the phase to pursue higher education or not by giving their predicted income through the attributes. This can be helpful when the student lacks in economic factors and needs to make an effective decision for the time being.

1.3 Objective of the Project:

- The main objective of this project is to make a classification model on the basis of income of the society and also classify the income of an individual above 50k or below 50k.

- Improve the income level of an individual and hence the society by implementing the “development of one development of all” technique.

- To improve Government’s decision making in implementing different yojanas by providing a classification of the demographic region and hence stating the group(s) it will be affected.

- to ensure sustainable development and improve the economic stability of a nation.

- However the main objective of this project is to introduce ourselves to the domain of machine learning.

- Through the project we shall be able to rigorously apply machine learning to our problems.

- After the project we shall have all the necessary background for fundamental machine learning research.

1.4 Scope of the Project

- Data Collection: The suitable data needs to be collected from several open source platforms or public repositories like UCI repository or some dataset provider or a website like Kaggle.

- Preprocessing Of Data: Address and assess the different attributes in the dataset and to understand their role and importance in the income.

- Removing insignificant columns: To remove the attributes and related data which is insignificant in building the model. These attributes prove to be redundant in the further data processing so they are to be removed in the beginning through human analysis.

- Removing missing data: The data fields (rows) with inconsistent as well as missing data needs to be removed before processing of data. The missing values will make our model inconsistent and reduce the accuracy so it is better to compromise a row of data for the missing values.

- Building a model: A machine learning mode is to be created using one or more classifiers like decision tree. We will later do the feature inspection in our model and generalize the capability of the same.

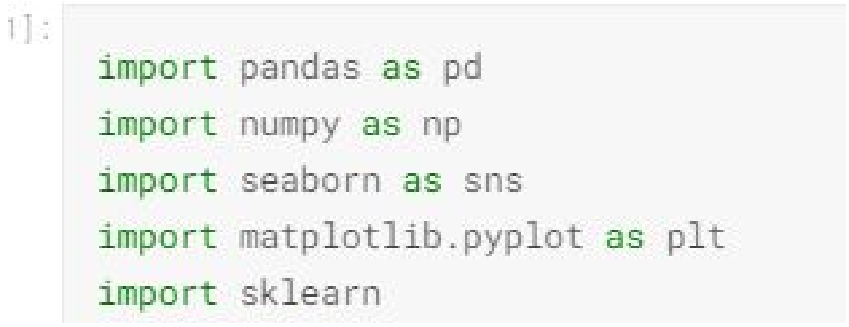
- Why machine learning? : We are applying machine learning because the methodologies have proven to be of great use and practical value over a variety of applicable domains and real life situations.

- How will Machine Learning be applied: The first and the foremost requirement of any model is to get a good dataset after that we shall analyse the data fields and replace the different string annotations with meaningful and justified numeric values for a better training and visualisation of our model.

- Evaluation of model: The model needs to be evaluated by testing it against the test data and checking its accuracy.

1.5 Need of Work

We will be working on the dataset cleaning , data visualisation, data encoding, applying machine learning models and then selecting the best working model.



```
1]:
    import pandas as pd
    import numpy as np
    import seaborn as sns
    import matplotlib.pyplot as plt
    import sklearn
```

Figure 1.5.1 : Getting started with packages

CHAPTER 2

RELATED WORK

Several efforts have been made earlier in this field of machine learning to classify the income based on the demographic factors and some were on our chosen dataset too.

Some of the researchers who have been working in the past on this problem statement are mentioned below and referenced at the end of this report:

- Chockalingam et. al. [1] explored and analysed the Adult Dataset and used several Machine Learning Models like Logistic Regression, Stepwise Logistic Regression, Naive Bayes, Decision Trees, Extra Trees, k-Nearest Neighbor, SVM, Gradient Boosting and 6 configurations of Activated Neural Network. They also drew a comparative analysis of their predictive performances

- Bekena [2] implemented the Random Forest Classifier algorithm to predict income levels of individuals.

- Topiwalla [3] made the usage of complex algorithms like XGBOOST, Random Forest and stacking of models for prediction tasks including Logistic Stack on XGBOOST and SVM Stack on Logistic for scaling up the accuracy.

- Deepajyothi et. al. [5] tried to replicate Bayesian Networks, Decision Tree Induction, Lazy Classifier and Rule Based Learning Techniques for the Adult Dataset and presented a comparative analysis of the predictive performances.

- Lemon et. al. [6] attempted to identify the important features in the data that could help to optimize the complexity of different machine learning models used in classification tasks.

- Haojun Zhu [7] attempted Logistic Regression as the Statistical Modelling Tool and 4 different Machine Learning Techniques, Neural Network, Classification and Regression Tree, Random Forest, and Support Vector Machine for predicting Income Levels

CHAPTER 3

PROPOSED METHODOLOGY

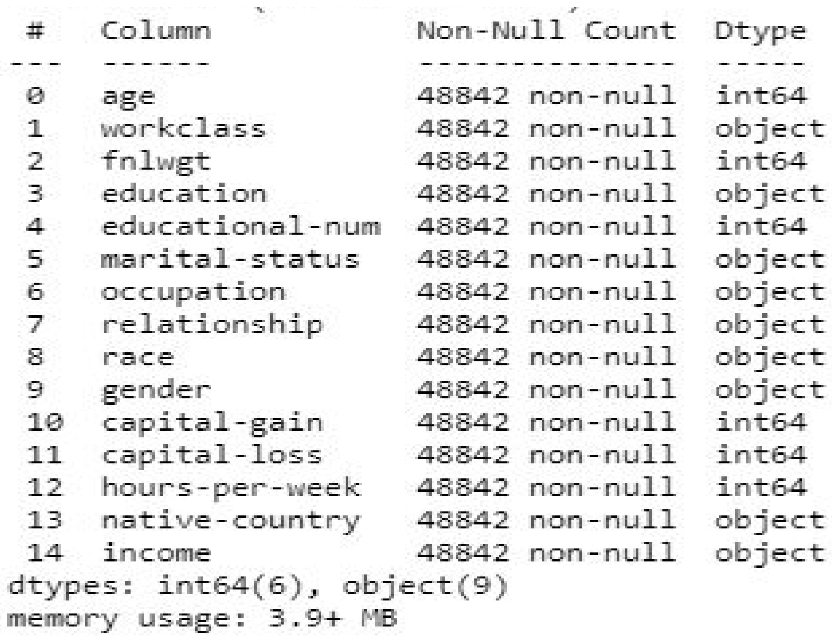
3.1 Dataset Description

- **Data Overview:**

The dataset used in this project has 48,842 records and a binomial label indicating a salary of <50K or >50K USD. 76% of the records in the dataset have a class label of <50K. The data has been divided into a training set containing 32,561 records and a test dataset containing 16,281 records.

- **Attributes Description:**

There are 14 attributes consisting of eight categorical and six continuous attributes (Fig 3.1).



```
 #   Column             Non-Null Count  Dtype
---  ------             --------------  -----
 0   age                48842 non-null  int64
 1   workclass          48842 non-null  object
 2   fnlwgt             48842 non-null  int64
 3   education          48842 non-null  object
 4   educational-num    48842 non-null  int64
 5   marital-status     48842 non-null  object
 6   occupation         48842 non-null  object
 7   relationship       48842 non-null  object
 8   race               48842 non-null  object
 9   gender             48842 non-null  object
 10  capital-gain       48842 non-null  int64
 11  capital-loss       48842 non-null  int64
 12  hours-per-week     48842 non-null  int64
 13  native-country     48842 non-null  object
 14  income             48842 non-null  object
dtypes: int64(6), object(9)
memory usage: 3.9+ MB
```

Figure 3.1 : Attributes in the dataset and their types

A. **Workclass**: Private, Self-emp-not-inc, Self-emp-inc, Federal-gov, Local-gov, State-gov, Withoutpay, Never-worked. 69.4% of values are Private. 6% of values are unknown (‘?’)

B. **Education**: Education contains the highest level of education attained such as high school or doctorate. Bachelors, Some-college, 11th, HS-grad, Prof-school, Assoc-acdm, Assoc-voc, 9th, 7th8th, 12th, Masters, 1st-4th, 10th, Doctorate, 5th-6th, Preschool. 32% are high school graduates, 22% went to some college and 16.5% have a bachelor’s degree.

C. **Education Num**: This is our self introduced attribute which provides consistency as well as an integer reference to the education attribute. The referencing is done as : High School = 10, Bachelors = 12, Diploma = 11, etc.

D. **Occupation**: The employment class describes the type of employer such as self-employed or federal and occupation describes the employment type such as farming, clerical or managerial. Tech-support, Craft-repair, Other-service, Sales, Exec-managerial, Prof-specialty,Handlers-cleaners, Adm-clerical, Farming-fishing, Transport-moving, Privhouse-serv, Protective-serv, Armed-Forces. 6% values are unknown, evenly distributed except armed forces (0.03%) and private house servant (0.5%)

E. **Capital Gain**: The total estimated gain in income from previous year.

F. **Capital Loss**: The total estimated gain in income from previous year.

G. **Hours per Week**: Continuous attribute determining the hours a person works in a week.

H. **Marital Status**: Married-civ-spouse, Divorced, Never-married, Separated, Widowed, Marriedspouse-absent, Married-AF-spouse. 46% are married to civilian, 33% are never married, 14% are divorced

I. **Relationship**: Wife, Own-child, Husband, Not-in-family, Other-relative, Unmarried. 40% are husbands, 26% are not in a family

J. **Race**: White, Asian-Pac-Islander, Amer-Indian-Eskimo, Other, Black. 86% are White and 10% are Black with negligible proportions of others .

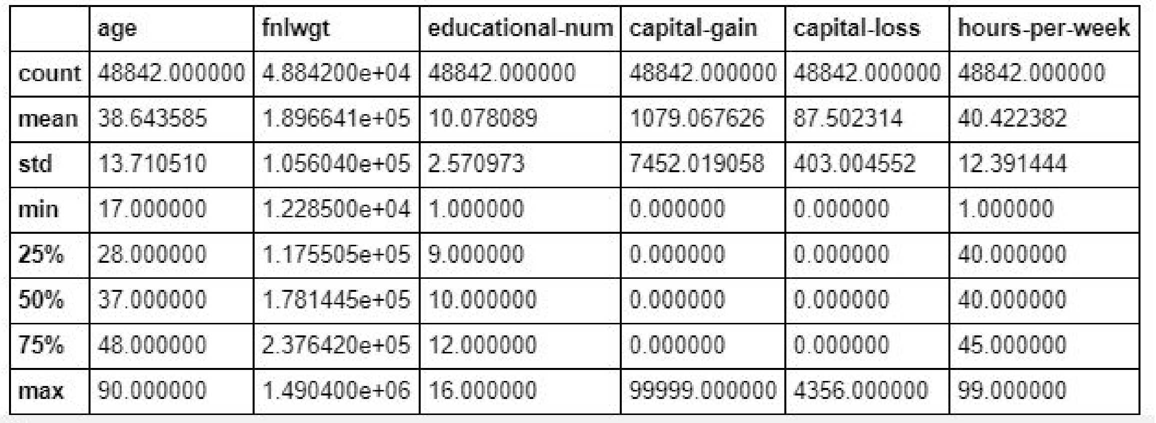
K. **Gender**: Female, Male. 67% are male and 33% female

L. **Native Country**: United-States, Cambodia, England, Puerto-Rico, Canada, Germany, OutlyingUS(Guam-USVI-etc), India, Japan, Greece, South, China, Cuba, Iran, Honduras, Philippines, Italy, Poland, Jamaica, Vietnam, Mexico, Portugal, Ireland, France, Dominican-Republic, Laos, Ecuador, Taiwan, Haiti, Colombia, Hungary, Guatemala, Nicaragua, Scotland, Thailand, Yugoslavia, ElSalvador, Trinidad and Tobago, Peru, Hong, Holland-Netherlands. 42 categories, 90% are from United States, 2% values are unknown

M. **Fnlwgt**: The weight of the individual.

N. **Income**: The predictive class for less or more than 50K.

Table 3.1 - Describing the attributes with continuous data



	age	fnlwgt	educational-num	capital-gain	capital-loss	hours-per-week
count	48842.000000	4.884200e+04	48842.000000	48842.000000	48842.000000	48842.000000
mean	38.643585	1.896641e+05	10.078089	1079.067626	87.502314	40.422382
std	13.710510	1.056040e+05	2.570973	7452.019058	403.004552	12.391444
min	17.000000	1.228500e+04	1.000000	0.000000	0.000000	1.000000
25%	28.000000	1.175505e+05	9.000000	0.000000	0.000000	40.000000
50%	37.000000	1.781445e+05	10.000000	0.000000	0.000000	40.000000
75%	48.000000	2.376420e+05	12.000000	0.000000	0.000000	45.000000
max	90.000000	1.490400e+06	16.000000	99999.000000	4356.000000	99.000000

3.2 Methods

The following Machine Learning methods were applied on our dataset to choose the perfect model with highest accuracy and work on the same.

- Logistic Regression
- Random Forest
- K-nearest neighbors or KNN
- Naive Bayes

Each of the above algorithms have their own set of advantages and disadvantages and their implementation with every necessary detail is described with full explanation ahead in the report.

Initially we have researched that logistic regression as well as linear regression are more prone to overfitting in these types of binary classification, so we shall take care of feeding the data to these algorithms and will create a separate training set for it.

Now, the algorithms with their working and accuracy is described :

Logistic Regression:

Logistic regression is a classification algorithm used to assign observations to a discrete set of classes. Some of the examples of classification problems are Email spam or not spam, Online transactions Fraud or not Fraud, Tumor Malignant or Benign. Logistic regression transforms its output using the logistic sigmoid function to return a probability value.

Why logistic Regression?

Suppose you have given data on "time spent on studying and exam scores by students". Linear Regression and logistic regression can predict different things. Linear Regression could help us predict the student's test score on a scale of 0 - 100. Logistic Regression could help us to predict whether the student passed or failed.

What are the types of logistic regression

1. Binary Logistic Regression (0/1) --> has only two 2 possible outcomes.

2. Multinomial Logistic Regression (Veg, Non-Veg, Vegan) --> Three or more categories without ordering.

3. Ordinal Logistic Regression (Low, Medium, High or movie rating 1 to 5) --> Three or more categories with ordering.

The hypothesis of logistic regression tends it to limit the cost function between 0 and 1. Therefore linear functions fail to represent it as it can have a value greater than 1 or less than 0 which is not possible as per the hypothesis of logistic regression.

$$0 \leq h_{\theta}(x) \leq 1$$

Figure 3.2.1 : Logistic regression hypothesis expectation

What is the Sigmoid Function?

In order to map predicted values to probabilities, we use the Sigmoid function. The function maps any real value into another value between 0 and 1. In machine learning, we use sigmoid to map predictions to probabilities.

$$f(x) = \frac{1}{1+\mathrm{e}^{-(x)}}$$

Figure 3.2.2 : Formula of sigmoid function

Hypothesis Representation

$$h\theta(X) = \frac{1}{1+e^{-(\beta_0+\beta_1 X)}}$$

Figure 3.2.3 : The hypothesis of logistic regression

Advantages of Logistic Regression

1. Logistic Regression performs well when the dataset is linearly separable.

2. Logistic regression is less prone to overfitting but it can overfit in high dimensional datasets. You should consider Regularization (L1 and L2) techniques to avoid over-fitting in these scenarios.

3. Logistic Regression not only gives a measure of how relevant a predictor (coefficient size) is, but also its direction of association (positive or negative).

4. Logistic regression is easier to implement, interpret and very efficient to train.

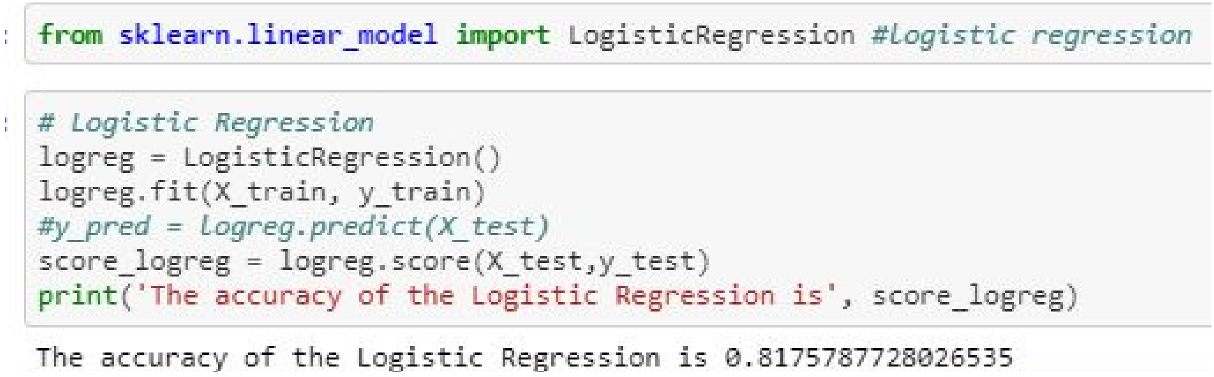
Disadvantages of Logistic Regression

1. Main limitation of Logistic Regression is the assumption of linearity between the dependent variable and the independent variables. In the real world, the data is rarely linearly separable. Most of the time data would be a jumbled mess.

2. If the number of observations are lesser than the number of features, Logistic Regression should not be used, otherwise it may lead to overfit.

3. Logistic Regression can only be used to predict discrete functions. Therefore, the dependent variable of Logistic Regression is restricted to the discrete number set. This restriction itself is problematic, as it is prohibitive to the prediction of continuous data.

How to apply?



```
from sklearn.linear_model import LogisticRegression #logistic regression
```

```
# Logistic Regression
logreg = LogisticRegression()
logreg.fit(X_train, y_train)
#y_pred = logreg.predict(X_test)
score_logreg = logreg.score(X_test,y_test)
print('The accuracy of the Logistic Regression is', score_logreg)
```

```
The accuracy of the Logistic Regression is 0.8175787728026535
```

Figure 3.2.4 : The implementation of logistic regression

ACCURACY : 81.75%

- **Random Forest :**

Random forests or random decision forests are an ensemble learning method for classification, regression and other tasks that operate by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classes (classification) or mean prediction (regression) of the individual trees. Random decision forests correct for decision trees' habit of overfitting to their training set.

The above is the definition of the Random Forest algorithm in wikipedia however we shall be describing it in our own way of understanding and interpretation.
First of all we shall define forests in data structures. The forests typically are trees data structure but having a single node which makes it extend in a direction and grow like a forest, hence its name. These are the extensions of decision trees, but in their own peculiar way which is the use of randomisation.

The first algorithm for random decision forests was created by Tin Kam Ho using the random subspace method, which, in Ho's formulation, is a way to implement the "stochastic discrimination" approach to classification proposed by Eugene Kleinberg.

Another amazing part of Random Forest Algorithm is that it can be used for both classification and regression problems. There is a direct relationship between the number of trees in the forest and the results it can get: the larger the number of trees, the more accurate the result. But one thing to note is that creating the forest is not the same as constructing the decision with information gain or gain index approach.

The difference between the Random Forest algorithm and the decision tree algorithm is that in Random Forest, the processes of finding the root node and splitting the feature nodes will run randomly.

- **Why** was Random Forest considered in this model ?
  - The first and the foremost reason being that it can be used for both classification and regression tasks.
  - Overfitting is one critical problem that may make the results worse, but for the Random Forest algorithm, if there are enough trees in the forest, the classifier won’t overfit the model.
  - The third advantage is the classifier of Random Forest can handle missing values.
  - It can also be modelled for categorical values.

- **How** the Random Forest Algorithm works ?

There are two stages in Random Forest algorithm, one is random forest creation, the other is to make a prediction from the random forest classifier created in the first stage.

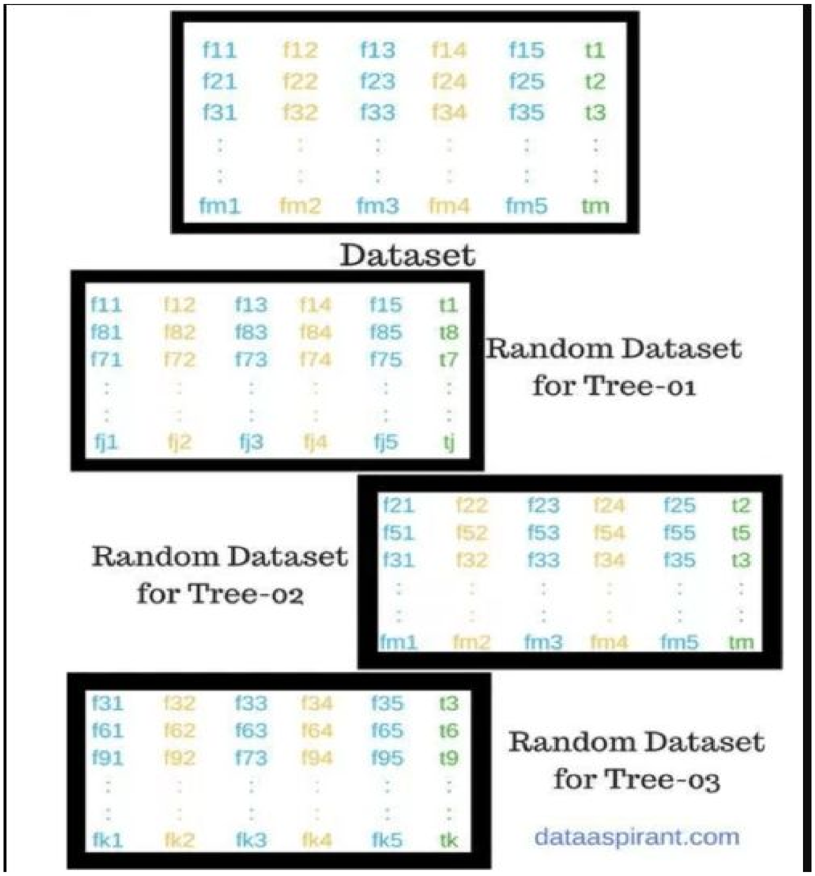
The pseudo code for random forest creation is as follows :

1. Randomly select “K” features from total “m” features where k << m.
2. Among the “K” features, calculate the node “d” using the best split point.
3. Split the node into daughter nodes using the best split.

4. Repeat the a to c steps until “l” number of nodes has been reached.

5. Build forest by repeating steps a to d for “n” number times to create “n” number of trees.

The process can be illustrated by the following figure [3.2.1]


Dataset
Random Dataset for Tree-01
Random Dataset for Tree-02
Random Dataset for Tree-03
dataaspirant.com

f11	f12	f13	f14	f15	t1
f21	f22	f23	f24	f25	t2
f31	f32	f33	f34	f35	t3
:	:	:	:	:	:
:	:	:	:	:	:
fm1	fm2	fm3	fm4	fm5	tm

f11	f12	f13	f14	f15	t1
f81	f82	f83	f84	f85	t8
f71	f72	f73	f74	f75	t7
:	:	:	:	:	:
:	:	:	:	:	:
fj1	fj2	fj3	fj4	fj5	tj

f21	f22	f23	f24	f25	t2
f51	f52	f53	f54	f55	t5
f31	f32	f33	f34	f35	t3
:	:	:	:	:	:
:	:	:	:	:	:
fm1	fm2	fm3	fm4	fm5	tm

f31	f32	f33	f34	f35	t3
f61	f62	f63	f64	f65	t6
f91	f92	f73	f94	f95	t9
:	:	:	:	:	:
:	:	:	:	:	:
fk1	fk2	fk3	fk4	fk5	tk

Figure 3.2.5 Example of Random Forest data division

In the following stage, with the random forest classifier created, we will make the prediction. The random forest prediction pseudocode is shown below:

1. Take the test features and use the rules of each randomly created decision tree to predict the outcome and store the predicted outcome (target).

2. Calculate the votes for each predicted target.

3. Consider the high voted predicted target as the final prediction from the random forest algorithm

- **Advantages** of Random Forest Algorithm :

  These are the following advantages of Random Forest Algorithm over other algorithms:

  ★ For applications in classification problems, Random Forest algorithm will avoid the overfitting problem.

  ★ For both classification and regression tasks, the same random forest algorithm can be used.

  ★ The Random Forest algorithm can be used for identifying the most important features from the training dataset, in other words, feature engineering.

- **Applications** of Random forest Algorithm :

  The random forest algorithm has its application in different sectors, some of them are mentioned below and includes Banking, Medicine, Stock Market and E-commerce.

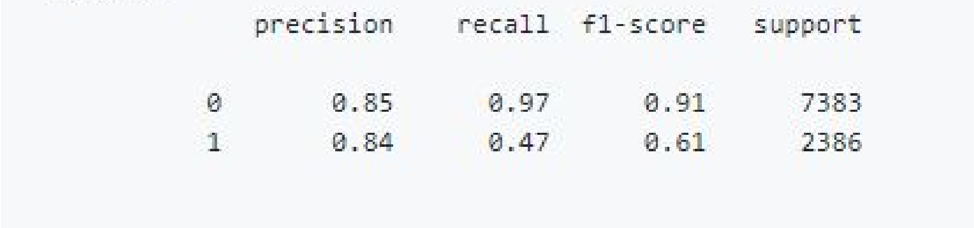
  ★ For the application in banking, Random Forest algorithm is used to find loyal customers, which means customers who can take out plenty of loans and pay interest to the bank properly, and fraud customers, which means customers who have bad records like failure to pay back a loan on time or have dangerous actions.

★ For the application in medicine, Random Forest algorithm can be used to both identify the correct combination of components in medicine, and to identify diseases by analyzing the patient's medical records.

★ For the application in the stock market, Random Forest algorithm can be used to identify a stock's behavior and the expected loss or profit.

★ For the application in e-commerce, the Random Forest algorithm can be used for predicting whether the customer will like the recommended products, based on the experience of similar customers.

Table 3.2.1 Random Forest Accuracy and Classification Report



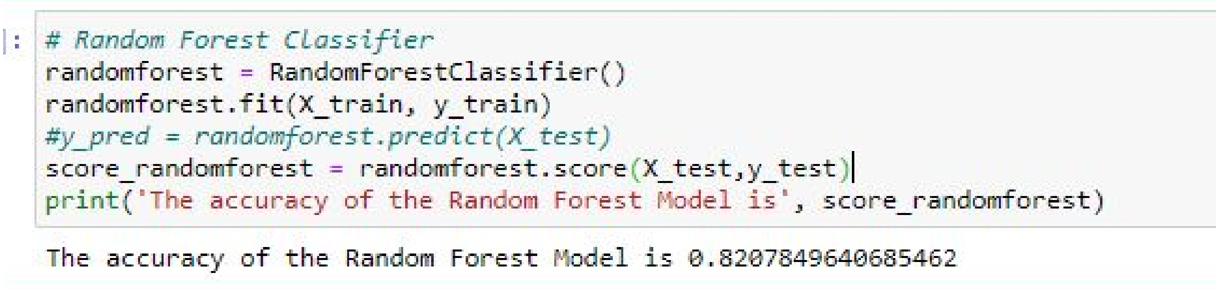
```
              precision    recall  f1-score   support

           0       0.85      0.97      0.91      7383
           1       0.84      0.47      0.61      2386
```

- **APPLYING** the Random Forest Algorithm in our project :

Below is the implementation of Random Forest Algorithm our our dataset. [figure 3.2.2]

The training set consisted of 50 % of our dataset
The testing set had 20 % of our dataset.
The predictive set had 30% of our dataset.



```
]: # Random Forest Classifier
   randomforest = RandomForestClassifier()
   randomforest.fit(X_train, y_train)
   #y_pred = randomforest.predict(X_test)
   score_randomforest = randomforest.score(X_test,y_test)
   print('The accuracy of the Random Forest Model is', score_randomforest)

   The accuracy of the Random Forest Model is 0.8207849640685462
```

Figure 3.2.6 The implementation of Random Forest

ACCURACY: 82 %

Naive Bayes

Naive Bayes methods are a set of supervised learning algorithms based on applying Bayes' theorem with the "naive" assumption of conditional independence between every pair of features given the value of the class variable. Bayes' theorem states the following relationship,given class variable y and dependent feature vector x1,through x n,:

$$P(y \mid x1,\ x2,\ ...xn)\ =\ (P(y) * \ P(x1,\ x2,\ ...xn))\ /\ P(x1,\ x2,\ ...\ xn)$$

which can be plainly represented as:

$$P(y|x1,\ldots..,xn)=\frac{P(y)P(x1,.....,xn)}{P(x1,x2,...,xn)}$$

as P(x1,x2,...,xn) is same(const.) for every sample,

$$\hat{y} = \underset{y}{argmax}\, P(y) \prod_{i=1}^{n} P(x_i | y)$$

where argmax is const. of proportionality;

Why Naive Bayes?

Naive Bayes classifiers have worked quite well in many real-world situations, famously document classification and spam filtering. They require a small amount of training data to estimate the necessary parameters.

Types of Naive Bayes:

1. Gaussian Naive Bayes : GaussianNB implements the Gaussian Naive Bayes algorithm for classification.

2. Multinomial Naive Bayes :Multinomial NB implements the naive Bayes algorithm for multi normally distributed data, and is one of the two classic naive Bayes variants used in text classification (where the data are typically represented as word vector counts).

3. Complement Naive Bayes :ComplementNB implements the complement algorithm. CNB is an adaptation of the standard multinomial naive Bayes (MNB) algorithm that is particularly suited for imbalanced data sets.

4. Bernoulli Naive Bayes : BernoulliNB implements the naive Bayes training and classification algorithms for data that is distributed according to multivariate Bernoulli distributions; i.e., there may be multiple features but each one is

assumed to be a binary-valued (Bernoulli, boolean) variable.

5. Categorical Naive Bayes : CategoricalNB implements the categorical naive Bayes algorithm for categorically distributed data. It assumes that each feature, which is described by the index i , has its own categorical distribution.

6. Out-of-core Naive Bayes : Naive Bayes models can be used to tackle large scale classification problems for which the full training set might not fit in memory. To handle this case, MultinomialNB, BernoulliNB, and GaussianNB expose a partial_fit method that can be used incrementally as done with other classifiers as demonstrated in out-of-core classification of text documents.

Advantages of Naive Bayes:-

1. Naive Bayes requires a small amount of training data to estimate the test data. So, the training period is less.

2. When assumption of independent predictors holds true, a Naive Bayes classifier performs better as compared to other models.

3. Naive Bayes is also easy to implement.

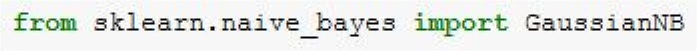
Disadvantages of Naive Bayes:-

1. Main imitation of Naive Bayes is the **assumption of independent predictors**. Naive Bayes implicitly assumes that all the attributes are mutually independent. In real life, it is almost impossible that we get a set of predictors which are completely independent.

2. If categorical variable has a category in test data set, which was not observed in training data set, then model will assign a 0 (zero) probability and will be unable to make a prediction. This is often known as **Zero Frequency**.

How to Apply?

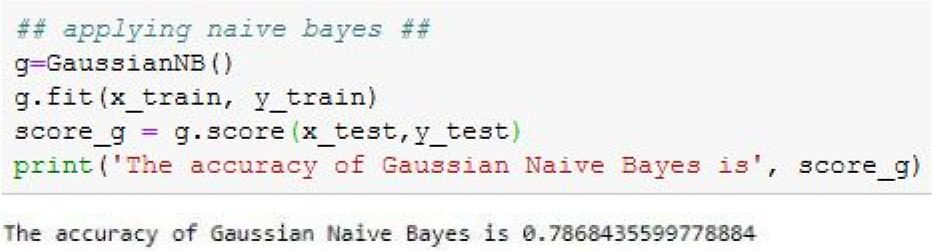
1. Importing GaussianNB module from naive bayes library in sklearn.



```
from sklearn.naive_bayes import GaussianNB
```

Figure:3.2.7 Importing GaussianNB module

2. fitting naive bayes algo in our training data and finally checking accuracy score on our test data



```
## applying naive bayes ##
g=GaussianNB()
g.fit(x_train, y_train)
score_g = g.score(x_test,y_test)
print('The accuracy of Gaussian Naive Bayes is', score_g)
```

```
The accuracy of Gaussian Naive Bayes is 0.7868435599778884
```

Figure:3.2.8 Implementing GaussianNB

Accuracy : 78%

3.3 Hardware / Software Requirements

- *Laptop/ Desktop* (Necessary): x64 bit Windows 10 machine with 1TB Hard disk Space and 4GB RAM.
- *Command Prompt/ Terminal*: Command Prompt or Terminal for giving commands directly to the system.
- *Jupyter Notebook* (Necessary): For executing python and ML.
- *Anaconda IDE*: IDE for python and development
- *Visual Studio Code*: Text Editor :
- *Git*: maintaining version control history and uploading necessary commits/changes as well as working in a team.
- *Python3* (Necessary): Python programming language version 3.8.1
- *Python IDLE*: For executing inline commands.
- *Numpy library* (Necessary): For faster execution of numeric python as in list and numbers.
- *Pandas library* (Necessary): Pandas v1.0.5
- *Matplotlib* (Necessary): For data Visualisation
- *Seaborn* (Necessary): For data Visualisation

- *Pyplot* (Necessary): For data Visualisation
- *Scikit-learn* (Necessary): For machine learning kit.

3.4 Our Methodology

The major task that we had to do after getting the dataset from the UCI repository was to clean it and make it ready for use in Machine Learning. It was necessary as well as an important task in itself .

The following methods were applied in achieving the above:

- Data Cleaning
- Dropping insignificant columns
- Treating missing values
- Redefining Attributes
- Manipulating the prediction class

- **Data Cleaning :** The most important part was to clean the data. Our dataset had several insignificant rows and missing values and ambiguous data which were to be removed as described below.

- **Dropping insignificant columns:** Out of the 14 attributes which as described in the dataset, we found three of them as unnecessary which were namely fnlwgt, relationship, education.

- fnlwgt: we consider the weight of the person as indeterminate in the analysis of the income as it has no effect on it.

- relationship: A person’s position in the family as well as their relationship with their siblings, parents or spouse is unnecessary in determining their income.

- education: Education is a very necessary part in determining the income of a person, but we removed it from our dataset because of the two ultimate reasons which are stated below:

    - Firstly, the column with education had ambiguous values, some being random strings or strings like *‘some-college’ , ‘no college’,* etc. So it had obviously no meaning to the context and so it had to be removed.

    - Secondly, our dataset had a similar attribute named education-num which had the education number of the corresponding education they achieved. It was more relevant to the context and easy to handle and plot. So it was given preference in choosing over two similar attributes.

- **Removing missing values :** Our dataset had no *nan* values which we checked using numpy but it had ambiguous data and strings in the form of *‘?’* which were in around 3048 rows. The following procedures were taken to eradicate those values:

  - Finding their total occurrence: Firstly we found out the total occurence of those strings using the *np.count* method. There were 4021 occurrences of the string. The distribution was uneven with some being in the complete rows filled with *'?'* and others being a single data or two.

  - Replacing with np.nan: The *'?'* was initially replaced by *np.nan* of numpy which stands for *not a number* value. We chose it because pandas/ numpy offers easy dropping of those values and we were happy with it.

  - Dropping the rows containing them: The rows with *np.nan* were finally dropped using *pd.dropna* method which offers easy dropping of data with nan.

- **Redefining the attributes:** We had to redefine several attributes and the data contained in it to make it useful for machine learning. It was a tedious process which was taken out precautionarily so as to not completely change the originality of the data. The following attributes were redesigned and with the reason of doing so is mentioned.

  - Workclass : Workclass contained different values like Government, Armed forces, Private Sector , 'Private', 'Local-gov', 'Self-emp-not-inc', 'Federal-gov', 'State-gov', 'Self-emp-inc', 'Without-pay' which have been encoded into numeric values based on their importance and priority basis like :

    - Without Pay = 0,
    - Self Employed = 2,
    - State Gov = 3,
    - Federal Gov = 3,
    - Private Sector = 2.

- Education : The attribute which is estimated to have a great impact on our estimation has been changed based on the degree and skillset of education which is stated to be achieved. It has been modified in the following ways :

  - Uneducated = 5
  - Preschool = 10
  - Primary Classes 1-10 : their corresponding numbers which are to be less than 10
  - HS-grad : 10
  - 11th = 11
  - 12th = 12
  - Bachelors = 15
  - Some college = 13
  - Doctorate = 15
  - Diploma = 14
  - Some degree = 13

  The hereby modified education attribute will be referred to as *education_encoded* in the upcoming references.

- Education Num : This had the total number of years a person spent on education it is very similar to the above encoded education of ours but was not consistent. So this has been modified in the following way:

  - Computing the sum of *(2 * Education Num)* and *education_encoded,* let’s call it *education_sum.* Dividing education sum by 3 and taking the rounded value to the nearest integer. It is therefore a cumulative partition of education_encoded and education_number shifted towards the education_number in the ratio of 2:1.

- Marital Status :This attribute contains categorical values like Never married, Marries-civ-spouse,Divorced,widowed,separated etc. which has been encoded as:

    - Married : 3
    - Not-married : 1
    - Seperated : 2

- Gender : This attribute basically had just two values with no abnormality such as transgender etc so it was coded in binary :

    - Female : 0
    - Male : 1

- Occupation : This categorical attribute had the maximum number of unique entries describing the specifications of different jobs so it was very difficult to first make a layout for the classification on numeric basis. However, we generalised the different attributes in major classifications and then encoded tem as following :

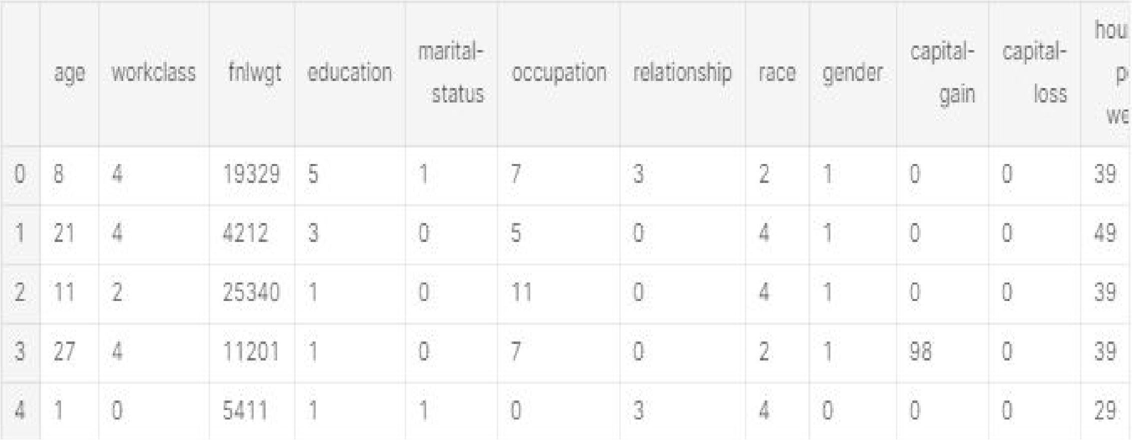
    - Tech Related: 4
    - Hardware Related : 3
    - Labour : 2
    - Service : 5
    - Business : 5
    - Govt. Jobs: 5
    - Others : 3

We used regular expressions and np.where as well as Python's own string methods to mark the attributes.

- Race : Race is encoded into numeric as :
    - White : 1
    - Black : 2
    - Others : 3

- Native Country : We initially thought of classifying them based on the continents from np.arange(0, 7) but this idea was later dropped after analysing that the continents like Oceania had very less population and the majority of Asia's population was contributed through India. The only two major continents that came out as a factor were America and Europe. So we thought it would make better if we just classify the data as:
    - Locals : 1
    - Non - Locals : 0

The final dataset after numerically encoded looks like this.



	age	workclass	fnlwgt	education	marital-status	occupation	relationship	race	gender	capital-gain	capital-loss	hou p we
0	8	4	19329	5	1	7	3	2	1	0	0	39
1	21	4	4212	3	0	5	0	4	1	0	0	49
2	11	2	25340	1	0	11	0	4	1	0	0	39
3	27	4	11201	1	0	7	0	2	1	98	0	39
4	1	0	5411	1	1	0	3	4	0	0	0	29

Figure 3.4.1 : Final dataset

NOTE : The other attributes were already integer based or continuous so there was no need for the dedicated encoding there. Attributes like fnlwgt, relationship, Education were ultimately dropped.

CHAPTER 4

EXPERIMENT AND RESULT ANALYSIS

After cleaning, analyzing, and plotting the data, here are some insights:

- Only about 1/3 of the population at the time would be considered high income while 2/3 of the population was making less than 50,000 USD per year.

- The greatest percentage of Asians were making over 50K with White class following close behind.

- Capital Gain was a good indicator of wealth with a pretty clear separation of people making higher than 50k with higher capital gain which is an indicator of the wealth gap in the US starting to grow.

- Capital Loss was a mixture of both high income and low income individuals and not a clear indicator of wealth.

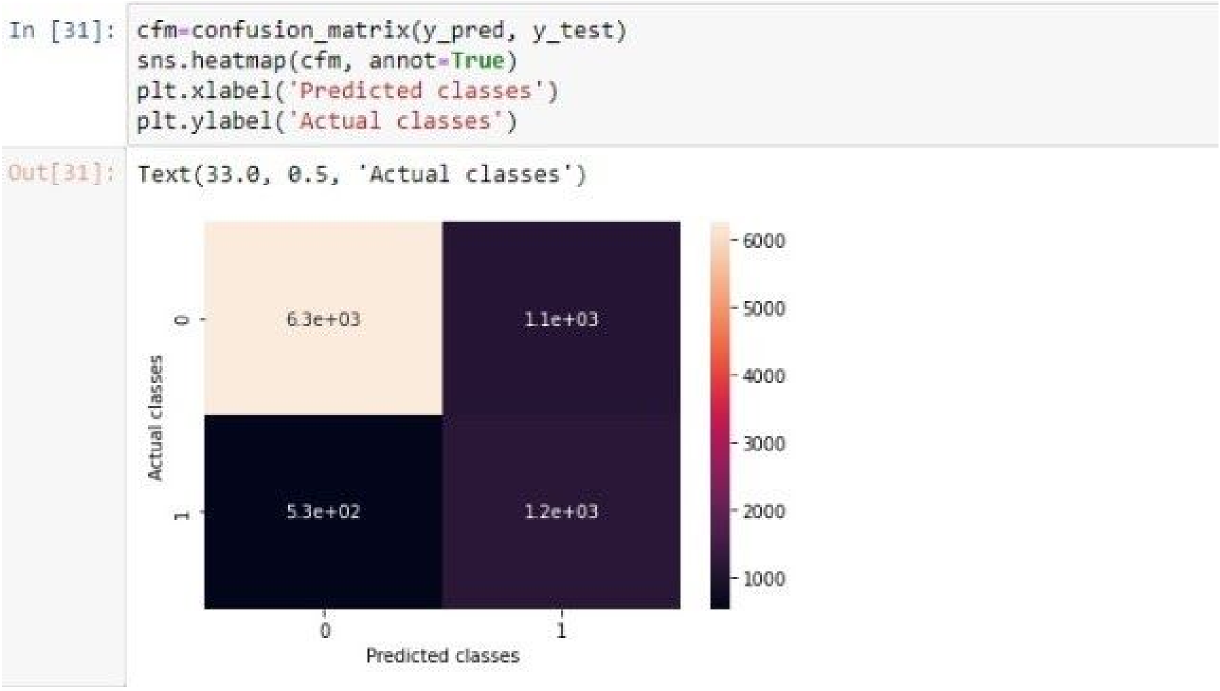
- Education was a pretty good indicator of income with the highest percentage of high income individuals finishing a pHD, Masters, or Bachelor's degree. The majority of the population had either a high school degree and / or some college finished..

- Married people had the highest percentage of high income people with husbands making up the majority of the workforce.

- The male working market more than doubled the female working market in 1994.

- The male dominant job was Craft-repair whereas the female dominant job was Adm-clerical.

Confusion Matrix:



```
In [31]: cfm=confusion_matrix(y_pred, y_test)
         sns.heatmap(cfm, annot=True)
         plt.xlabel('Predicted classes')
         plt.ylabel('Actual classes')

Out[31]: Text(33.0, 0.5, 'Actual classes')
```

Fig. 4.1 Confusion Matrix

A confusion matrix is a summary of prediction results on a classification problem. The number of correct and incorrect predictions are summarized with count values and broken down by each class. This is the key to the confusion matrix. It gives us insight not only into the errors being made by a classifier but more importantly the types of errors that are being made.

Definition of the Terms:

- Positive (P) : Observation is positive (for example: is an apple).
- Negative (N) : Observation is not positive (for example: is not an apple).
- True Positive (TP) : Observation is positive, and is predicted to be positive.
- False Negative (FN) : Observation is positive, but is predicted negative.
- True Negative (TN) : Observation is negative, and is predicted to be negative.
- False Positive (FP) : Observation is negative, but is predicted positive.

Classification rate/accuracy:

$$\textbf{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

Recall: Recall can be defined as the ratio of the total number of correctly classified positive examples divided to the total number of positive examples. High Recall indicates the class is correctly recognized (a small number of FN).

$$\textbf{\textit{Recall}} = \frac{TP}{TP + FN}$$

Precision: To get the value of precision we divide the total number of correctly classified positive examples by the total number of predicted positive examples. High

Precision indicates an example labelled as positive is indeed positive (a small number of FP).

$$\textit{Precision} = \frac{TP}{TP + FP}$$

F-measure: Since we have two measures (Precision and Recall) it helps to have a measurement that represents both of them. We calculate an F-measure which uses Harmonic Mean in place of Arithmetic Mean as it punishes the extreme values more.

$$\textit{F - measure} = \frac{2*Recall*Precision}{Recall + Precision}$$

in above fig.4.2, x axis indicates predicted class and y axis indicates actual class. in x axis 0 denotes “Less than or equal 50K” and 1 denotes “Greater than 50k”. This confusion matrix is made by using a random forest model. Class 0 is predicted correctly 6.3e+03 times and class 1 is predicted correctly 1.2e+03 times.

we can make another conclusion that class 0 means people whose income is less than or equal to 50k are more than class 1 i.e. people whose income is greater than 50k.

CHAPTER 5
CONCLUSION

5.1 Discussion:

We used StratifiedKFold to divide our dataset into k folds. In each iteration, k - 1 folds are used as the training the remaining fold is used as the validation. We use Stratified KFold because it preserves the percentage of samples from each class if we use KFold, we might run the risk of introducing sampling bias ie, the training set might contain a large number of samples where income is greater than 50K and test set contains More samples where income is less than 50K. Whereas Stratified KFold will ensure that there are enough samples of each class in both the train and test dataset. fine tuned using the Fine Tuning The model was GridSearchCV feature of sklearn.

5.2 Future Work :

- Adding a GUI : We shall be adding a Graphical User Interface using Python Tkinter and Flask Framework so as to make user interaction with the model easier and better.

- We shall be working on the accuracy of our model, the current accuracy lies in the range of around 80 to 85 percent. We will work on improving it.

- Making a better prediction model : We shall be making a better prediction model which shall not only work as a binary classifier but can also predict close to exact income of an individual through their given demographic factors.

- Dataset: to achieve the above goal our preliminary goal will be to get a dataset for that so we will also work on collecting the dataset from different repositories or otherwise work on arranging a survey to get necessary details for the same.

- As we can see that the dataset on which we worked was from the United States and it had a versatility but still it was limited when we start talking about a country like India which has ample of population as well as versatile culture and widespread demographical region. Therefore we shall next look forward to working for a model based on a country like India or more straight forward on Indian subcontinent.

- We will be looking forward to getting our model trained with a more recent dataset. The Adult Income dataset was of 1995.

- As mentioned in the objectives, one of the main objectives of working on a project like this was to help students in decision making for opting higher studies or not. Currently our model can use education for the same purpose but we will later work on extending the education number to a range of 100 (currently we have 16) based on the college, degree and quality of education.

- We will add decision making ability to our model.

REFERENCES

[1] Vidya Chockalingam, Sejal Shah and Ronit Shaw: “Income Classification using Adult Census Data”, https://cseweb.ucsd.edu/classes/wi17/cse258-a/reports/a120.pdf.

[2] Sisay Menji Bekena:“Using decision tree classifier to predict income levels”, Munich Personal RePEc Archive 30th July, 2017

[3] Mohammed Topiwalla: “Machine Learning on UCI Adult data Set Using Various Classifier Algorithms And Scaling Up The Accuracy Using Extreme Gradient Boosting”, University of SP Jain School of Global Management.

[4] Alina Lazar: “Income Prediction via Support Vector Machine”, International Conference on Machine Learning and Applications - ICMLA 2004, 16-18 December 2004, Louisville, KY, USA.

[5] S. Deepajothi and Dr. S.Selvarajan: “A Comparative Study of Classification Techniques On Adult Data Set”, International Journal of Engineering Research Technology (IJERT), ISSN: 2278-0181 Vol. 1 Issue 8, October-2012.

[6] Chet Lemon, Chris Zelazo and Kesav Mulakaluri: “Predicting if income exceeds $50,000 per year based on 1994 US Census Data with Simple Classification Techniques”, https://cseweb.ucsd.edu/ jm-cauley/cse190/reports/sp15/048.pdf.

[7] Haojun Zhu: “Predicting Earning Potential using the Adult Dataset”, https://rstudio-pubs-static.s3.amazonaws.com/23561751e06fa6c43b47d1b6daca2523b2f9e4.html

[8] https://archive.ics.uci.edu/ml/datasets/Adult

[9] https://medium.com/mlreview/gradient-boosting-from-scratch-1e317ae4587d