VISVESVARAYA TECHNOLOGICAL UNIVERSITY

Jnana Sangama, Belagavi-590018, Karnataka, INDIA

Mini
PROJECT REPORT
on

"Adult Census Income Analysis"

Submitted in partial fulfillment of the requirements for the VIII Semester
BIG DATA ANALYSIS (15CS82)

Bachelor of Engineering
IN
INFORMATION SCIENCE AND ENGINEERING

For the Academic year
2020-2021

BY

HITASHA KAKKAD	1PE17IS400
NITESH KALAL	1PE15IS070
PRASHANT KHOT	1PE17IS404
NIKHIL SHARMA	1PE16IS069

Under the Guidance of
Prof. ARYA S
Assistant Professor, Dept. of CSE
PESIT-BSC, Bengaluru-560100

Department of Information Science and Engineering
PESIT BANGALORE SOUTH CAMPUS
Hosur Road, Bengaluru -560100

PESIT BANGALORE SOUTH CAMPUS

Hosur Road, Bangalore -560100

Department of Information Science and Engineering

CERTIFICATE

Certified that the mini project work entitled ***"Adult Income Analysis"*** *is a bonafide work carried out by* ***Hitasha Kakkad, Nitesh Kalal***, ***Prashant Khot*** *and* ***Aman Mittal*** *student of* ***PESIT Bangalore South Campus*** *in partial fulfillment for the award of* ***Bachelor of Engineering*** *in* ***Information Science and Engineering*** *of the* ***Vishveshvaraya Technological University****, Belagavi during the year 2020-2021. It is certified that all corrections/suggestions indicated for Internal Assessment have been incorporated and the mini project report has been approved as it satisfies the academic requirements in respect of Mini Project work prescribed for the said Degree.*

Signatures:

Project Guide **Prof .Evlin** Asst.Professor, Dept. of ISE PESIT-BSC, Bengaluru	Head Dept of CSE **Dr.Annapurna** Professor, Dept. of ISE, PESIT-BSC, Bengaluru

External Viva

Name of the Examiners	**Signature with date**
1. _____	_____
2. _____	_____

ACKNOWLEDGEMENT

It is our privilege to convey our sincere regards to our **Principal Dr. Subhash Kulkarni** for their encouragement and giving us all the necessary resources to complete our project at the department premises for the partial fulfillment of the requirements leading to the award of BE degree.

We deeply express our sincere thanks to our **Head of Department Dr Prof. Annapurna D** for encouraging and allowing us to present the project on the topic "Adult Census Income" at our department premises for the partial fulfillment of the requirements leading to the award of BE degree.

It is our privilege to express our sincerest regards to our project coordinator, **Prof. Arya S**, for their valuable inputs, able guidance, encouragement, whole-hearted cooperation, and constructive criticism throughout the duration of our project.

We take this opportunity to thank all our lecturers who have directly or indirectly helped our project. We pay our respects and love to our parents and all other family members and friends for their love and encouragement throughout our career. Last but not the least we express our thanks to friends for their cooperation and support.

ABSTRACT

The prominent inequality of wealth and income is a huge concern especially in the United States. The likelihood of diminishing poverty is one valid reason to reduce the world's surging level of economic inequality. The principle of universal moral equality ensures sustainable development and improve the economic stability of a nation. Governments in different countries have been trying their best to address this problem and provide an optimal solution. This study aims to show the usage of machine learning and data mining techniques in providing a solution to the income equality problem. The UCI Adult Dataset has been used for the purpose. Classification has been done to predict whether a person's yearly income in US falls in the income category of either greater than 50K Dollars or less equal to 50K Dollars category based on a certain set of attributes. The Gradient Boosting Classifier Model was deployed which clocked the highest accuracy of 88.16%, eventually breaking the benchmark accuracy of existing works.

TABLE OF CONTENTS

LIST OF FIGURES

Chapter 1

1.1. INTRODUCTION

The project tries to analyse the problem of wealth inequality in society. Analysis is done on the data collected from public data libraries. We will be using techniques of data mining and machine learning to get insights on data. There are many factors to it to be considered like the Work class, education, age and investments. Goal of the project is to shed some light on which attributes influence wealth inequality and we tried to protect the income of an individual based on these factors.

1.1.1. Purpose of the project

The purpose of the project is to Analyse the wealth inequality from the available data and what factors influence them.This model actually aims to conduct a comprehensive analysis to highlight the key factors that are necessary in improving an individual's income. Build a model which can predict the income of individuals accurately.

1.1.2. Scope

This project is helpful to gaining knowledge about how wealth is distributed in society and what factors influence for some individuals to have higher income than others. We tried to protect the income of these individuals based on these factors and compare the accuracy of our model.

1.1.3. Definitions, acronyms and abbreviations

PYTHON:

Python is an interpreted high-level general-purpose programming language. Python's design philosophy emphasizes code readability with its notable use of significant indentation. Its language constructs as well as its object-oriented approach aim to help programmers write clear, logical code for small and large-scale projects. Python has many libraries and packages to support the development of machine learning. Python's large standard library, commonly cited as one of its greatest strengths, provides tools suited to many tasks.

PANDAS :

Pandas is a software library written for the Python programming language for data manipulation and analysis. In particular, it offers data structures and operations for manipulating numerical tables and time series. It is free software released under the three-clause BSD license. Pandas is mainly used for data analysis. Pandas allows importing data from various file formats such as comma-separated values, JSON, SQL, Microsoft Excel. Pandas allows various data manipulation operations such as merging, reshaping, selecting, as well as data cleaning, and data wrangling features. In our project Pandas helps exploratory data analysis and inputting data.

Scikit-learn :

Scikit-learn formerly scikits.learn and also known as sklearn is a free software machine learning library for the Python programming language. It features various classification, regression and clustering algorithms including support vector machines, random forests, gradient boosting, k-means and DBSCAN, and is designed to interoperate with the Python numerical and scientific libraries NumPy and SciPy. Scikit-learn is largely written in Python, and uses NumPy extensively for high-performance linear algebra and array operations.

1.2. Literature survey

Certain efforts using machine learning models have been made in the past by researchers for predicting income levels.

• Chockalingam 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 implemented the Random Forest Classifier algorithm to predict income levels of individuals.

• Lazar implemented Principal Component Analysis (PCA) and Support Vector Machine methods to generate and evaluate income prediction data based on the Current Population Survey provided by the U.S. Census Bureau.

1.3. Existing System

Most of the research done on this subject is conducted by private universities but the government hasn't made efforts to use machine learning to analyse census data and make decisions on this. Most of the analysis carried by the present governments is just by reading the data rather than conducting analysis based on data Analytics.

1.4. Proposed system

The data for our study was accessed from the University of California Irvine (UCI) Machine Learning Repository. It was actually extracted by Barry Becker using the 1994 census database. The data set includes figures on 48,842 different records and 14 attributes for 42 nations. The 14 attributes consist of 8 categorical and 6 continuous attributes containing information on age, education, nationality, marital status, relationship status, occupation, work classification, gender, race, working hours per week, capital loss and capital gain. The binomial label in the data set is the income level which predicts whether a person earns more than 50 Thousand Dollars per year or not based on the given set of attributes.

The solution involves steps like data preprocessing and feature selection which helps in making the data ready for model building. Models are built to predict the income of an individual. The model will be analysed based on accuracy score.

1.5. Statement of the Problem

Over the last two decades, humans have grown a lot of dependence on data and information in society and with this advent growth, technologies have evolved for their storage, analysis and processing on a huge scale. The fields of Data Mining and Machine Learning have not only exploited them for knowledge and discovery but also to explore certain hidden patterns and concepts which led to the prediction of future events, not easy to obtain.

Chapter 2

SOFTWARE REQUIREMENT SPECIFICATION

A Software Requirements Specification (SRS) is a complete description of the behavior of the system to be developed. It includes the functional and nonfunctional requirements for the software to be developed. The functional requirements include what the software should do and non-functional requirements include the constraint on the design or implementation. Requirements must be measurable, testable, related to identified needs or opportunities, and defined to a level of detail sufficient for system design.

Software requirement specification will contain what the software will do. When the software has to be directly perceived by its users – either human users or other software systems. The common understanding between the user and the developer is captured in the requirements document. The writing of software requirement specification reduces development effort, as careful review of the document can reveal omissions, misunderstandings, and inconsistencies early in the development cycle when these problems are easier to correct. The SRS discusses the product but not the project that developed it; hence the SRS serves as a basis for later enhancement of the finished product. The SRS may need to be altered, but it does provide a foundation for continued production evaluation.

2.1 Operating Environment

Our proposed system will work on any system that has the ability to run the latest version of the Google Chrome Browser or Mozilla Firefox and should have an Internet Connection.

2.1.1 Hardware Requirements

Table 2.1: Hardware Requirements

Hardware	**Description**
Processor	Intel(R)Core(TM) i3 and above
RAM	4.00 GB and above
Hard Disk	128GB and above

2.1.2 Software Requirements

Table 2.2: Software Requirements

Software	**Description**
Operating System	Ubuntu 14.04 or newer Windows 10
Programming Language	Python
Database	CSV file

2.3. Functional Requirements

These are the statements of services which, system should provide, how the system should react for particular inputs and how the system should behave in particular situations.

They are:-

- Project should be implemented with Open Architecture.
- Project should provide easy user interfaces.
- Project should aim to provide clarity of its working.
- Project should have the goal of making data analysis to derive more insights.
- Project should present results in a suitable format for clear understanding.

2.4. Non functional requirements

Accessibility

Accessibility can be viewed as the "ability to access" and benefit from a system. Help text for the modules is provided wherever necessary, which guides the user into being able to access the functionalities of the modules.

Availability

These are the statements of services which, system should provide, how the system should react for particular inputs and how the system should behave in particular situations. The modules are available to the users at all times.

Compatibility

As the system requires just the latest Google Chrome Browser or Mozilla Firefox, so it can be run on any operating system.

Performance

Since the web application does not require any critical procedure, the system performance only depends on the server's capacity to serve the client(s).

Reliability

Reliability is the ability of the system to perform and maintain its functions in routine, hostile and unexpected circumstances.

Portability

Since the app just requires latest Chrome browser or Mozilla Firefox so portability is not an issue

Usability

A system like the one proposed in this project is very helpful for governments and companies who want to gain knowledge about income distribution.

Security

Since the project is done on open sourced data we don't have no issue of security. Our project is stored on online cloud.

2.5. User characteristics

User

- Need to have a system connected to the Internet.
- User can run program with anytime with ease
- Users can see all the EDA conducted to gain more insights.
- Users can see the working of the model and the results presented.

2.6. Applications of Adult Income Analysis.

- Can help governments in key decision making regarding policies related to income inequality, taxes etc. which can help in bridging the gap in incomes.
- Gives proof to the world by research and data that wealth inequality is a real problem.
- It is helpful in creating awareness of the problem.

2.7. Advantages of Adult Income Analysis.

- Shows how different factors influence the income of people and how certain metrics like education directly correlates to the income of individuals.
- Preparing a machine learning model to predict the income on the factors.
- Conducting analysis on accuracy of these models and to increase accuracy.

Chapter 3

Design

3.1. System Architecture

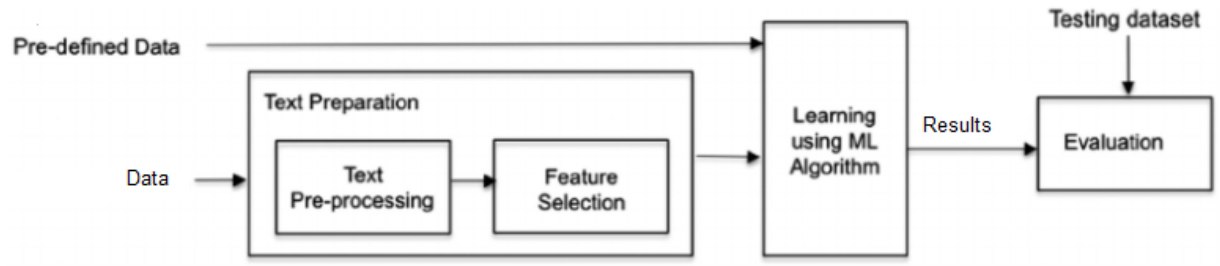

Fig 3.1 System Architecture

3.2. Data Flow Diagram

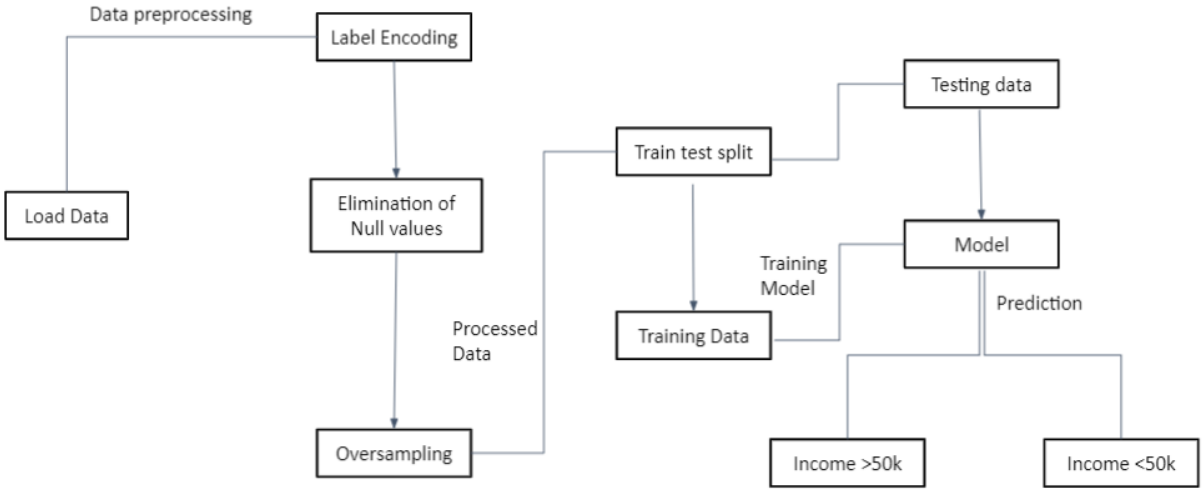

Fig 3.1 Data flow Diagram

Chapter 4

4.1 Implementation

The term implementation has different meanings, ranging from the conversion of a basic application to a compatible replacement of a computer system. Implementation is used here to mean the process of converting a new or revised system into an operational one.

During the implementation stage we convert the detailed code in a programming language. If the implementation stage is not carefully planned and controlled, it can cause great chaos. Thus it can be considered to be the most crucial stage in achieving the user's confidence that the new system will work effectively.

4.2. Programming Language Selection

We have selected python for the implementation along with its wide range of libraries available for conducting big data analysis and machine learning . The advantages of using pyhton along with its libraries are as follows –

1. Due to its ability to run on multiple platforms without the need to change, developers prefer Python, unlike in other programming languages. Python runs across different platforms, such as Windows, Linux, and macOS, thus requiring little or no changes.

2. The ease of executability makes it easy to distribute software, allowing standalone software to be built and run using Python.

3. The Python code is concise and readable, which simplifies the presentation process.A developer can write code easily and concisely compare it to other programming languages

4. Python's independence across platforms saves time and resources for developers, who would otherwise incur a lot of resources to complete a single project.

5. Libraries and frameworks are vital in the preparation of a suitable programming environment. Python frameworks and libraries offer a reliable environment that reduces software development time significantly.

Python has many important libraries which have helped us scikit-learn, Pandas, numpy and Matplotlib. The advantages of using these libraries in the for machine learning and big data programming are as follows –

1. Scikit-learn features various classification, regression and clustering algorithms including support vector machines, random forests, gradient boosting, k-means and DBSCAN, and is designed to interoperate with the Python numerical and scientific libraries NumPy and SciPy.

2. NumPy is a library in Python for adding support for large, multi-dimensional arrays and matrices, along with a large collection of high-level mathematical functions to operate on these arrays.

3. Pandas is a software library written for the Python programming language for data manipulation and analysis. In particular, it offers data structures and operations for manipulating numerical tables and time series.

4.3. Dataset.

Dataset imported from public library and analyzed.

Importing dataset

```
# Importing dataset
dataset = pd.read_csv('adult.csv')
```

Dataset Description

```
# Preview dataset
dataset.head()
```

	age	workclass	fnlwgt	education	education.num	marital.status	occupation	relationship	race	sex	capital.gain	capital.loss	hours.per.week	native
0	90	?	77053	HS-grad	9	Widowed	?	Not-in-family	White	Female	0	4356	40	United
1	82	Private	132870	HS-grad	9	Widowed	Exec-managerial	Not-in-family	White	Female	0	4356	18	United
2	66	?	186061	Some-college	10	Widowed	?	Unmarried	Black	Female	0	4356	40	United
3	54	Private	140359	7th-8th	4	Divorced	Machine-op-inspct	Unmarried	White	Female	0	3900	40	United
4	41	Private	264663	Some-college	10	Separated	Prof-specialty	Own-child	White	Female	0	3900	40	United

Shape of dataset

```
# Shape of dataset
print('Rows: {} Columns: {}'.format(dataset.shape[0],
dataset.shape[1]))
```

```
Rows: 32561 Columns: 15
```

4.4. Data preprocessing

Checking null values

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

```
Out[30]: 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
```

Fig 5.1 Null values present in each feature

Label Encoding

```
from sklearn.preprocessing import LabelEncoder


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

Feature Scaling

```
from sklearn.preprocessing import StandardScaler

for col in X.columns:
    scaler = StandardScaler()
    X[col] = scaler.fit_transform(X[col].values.reshape(-1, 1))
```

Fixing imbalanced dataset using Oversampling

```
In [44]: round(Y.value_counts(normalize=True) * 100, 2).astype('str') + ' %'
Out[44]: 0    75.92 %
         1    24.08 %
         Name: income, dtype: object

In [45]: from imblearn.over_sampling import RandomOverSampler
         ros = RandomOverSampler(random_state=42)

In [46]: ros.fit(X, Y)
Out[46]: RandomOverSampler(random_state=42)

In [47]: X_resampled, Y_resampled = ros.fit_resample(X, Y)

In [48]: round(Y_resampled.value_counts(normalize=True) * 100, 2).astype('str') + ' %'
Out[48]: 1    50.0 %
         0    50.0 %
         Name: income, dtype: object
```

Fig 5.4 steps to fixing imbalance data.

Creating a train test split

```
In [49]: from sklearn.model_selection import train_test_split
         X_train, X_test, Y_train, Y_test = train_test_split(
             X_resampled, Y_resampled, test_size=0.2, random_state=42)

In [50]: print("X_train shape:", X_train.shape)
         print("X_test shape:", X_test.shape)
         print("Y_train shape:", Y_train.shape)
         print("Y_test shape:", Y_test.shape)

         X_train shape: (39552, 8)
         X_test shape: (9888, 8)
         Y_train shape: (39552,)
         Y_test shape: (9888,)
```

Fig 5.5 Description of Loan table

4.5. Data Modelling

Logistic Regression

```
In [51]: from sklearn.linear_model import LogisticRegression
         log_reg = LogisticRegression(random_state=42)

In [52]: log_reg.fit(X_train, Y_train)

Out[52]: LogisticRegression(random_state=42)

In [53]: Y_pred_log_reg = log_reg.predict(X_test)
```

Fig 4.5.1 Logistic Regression

KNN Classifier

```
In [54]: from sklearn.neighbors import KNeighborsClassifier
         knn = KNeighborsClassifier()

In [55]: knn.fit(X_train, Y_train)

Out[55]: KNeighborsClassifier()

In [56]: Y_pred_knn = knn.predict(X_test)
```

Fig 4.5.2 KNN Classifier

Support Vector Classifier

```
In [57]: from sklearn.svm import SVC
         svc = SVC(random_state=42)

In [58]: svc.fit(X_train, Y_train)

Out[58]: SVC(random_state=42)

In [59]: Y_pred_svc = svc.predict(X_test)
```

Fig 4.5.3 Support Vector Classifier

Naive Bayes Classifier

```
In [60]: from sklearn.naive_bayes import GaussianNB
         nb = GaussianNB()

In [61]: nb.fit(X_train, Y_train)

Out[61]: GaussianNB()

In [62]: Y_pred_nb = nb.predict(X_test)
```

Fig 4.5.4 Naive Bayes Classifier

Decision Tree Classifier

```
In [63]: from sklearn.tree import DecisionTreeClassifier
         dec_tree = DecisionTreeClassifier(random_state=42)

In [64]: dec_tree.fit(X_train, Y_train)

Out[64]: DecisionTreeClassifier(random_state=42)

In [65]: Y_pred_dec_tree = dec_tree.predict(X_test)
```

Fig 4.5.5 Decision Tree Classifier

Random Forest Classifier

```
In [66]: from sklearn.ensemble import RandomForestClassifier
         ran_for = RandomForestClassifier(random_state=42)

In [67]: ran_for.fit(X_train, Y_train)

Out[67]: RandomForestClassifier(random_state=42)

In [68]: Y_pred_ran_for = ran_for.predict(X_test)
```

Fig 4.5.6 Random Forest Classifier

Chapter 5

Testing and Results

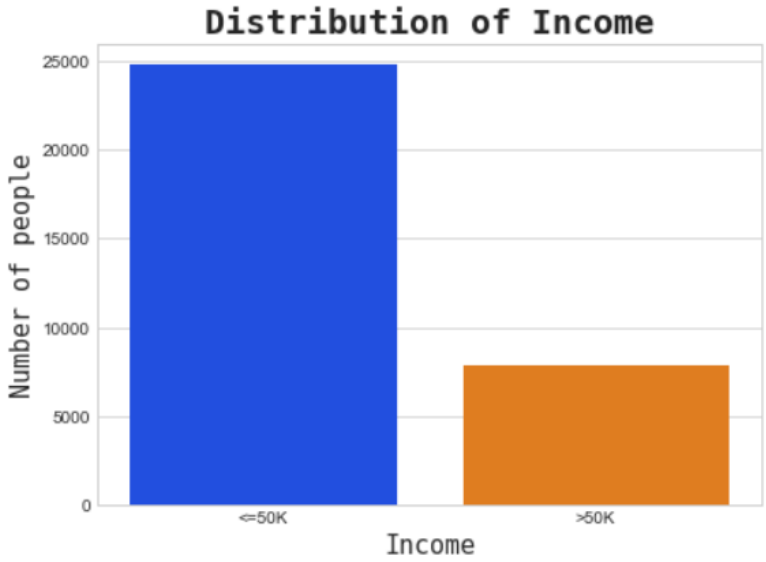

Fig 5.1 Distribution of income

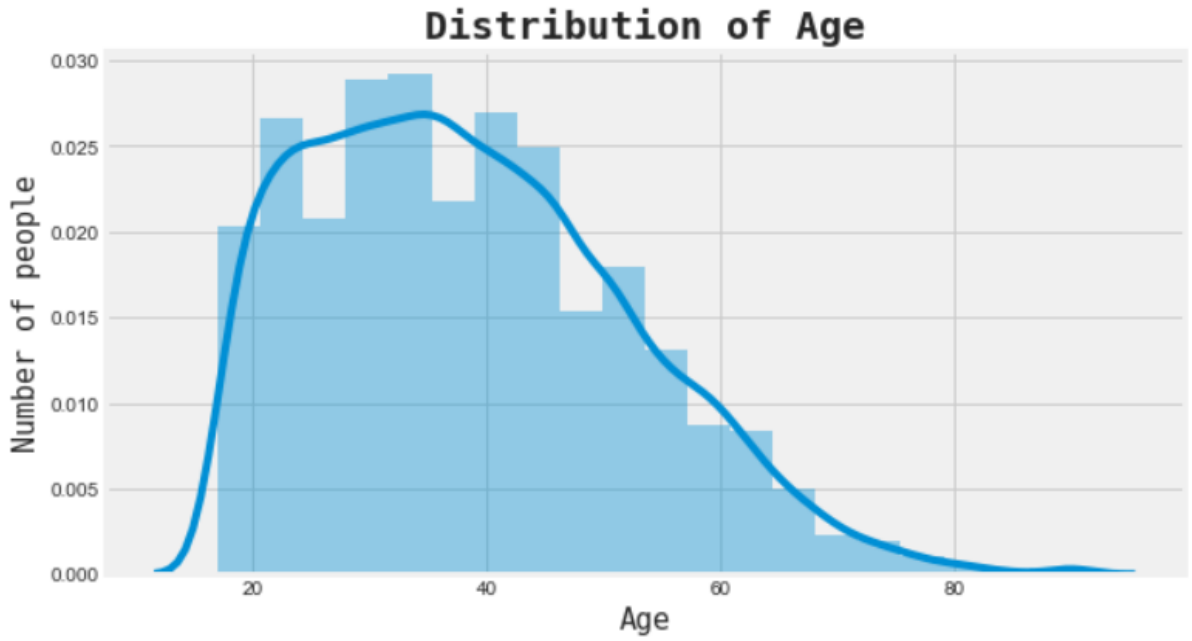

Fig 5.2 Distribution of Age

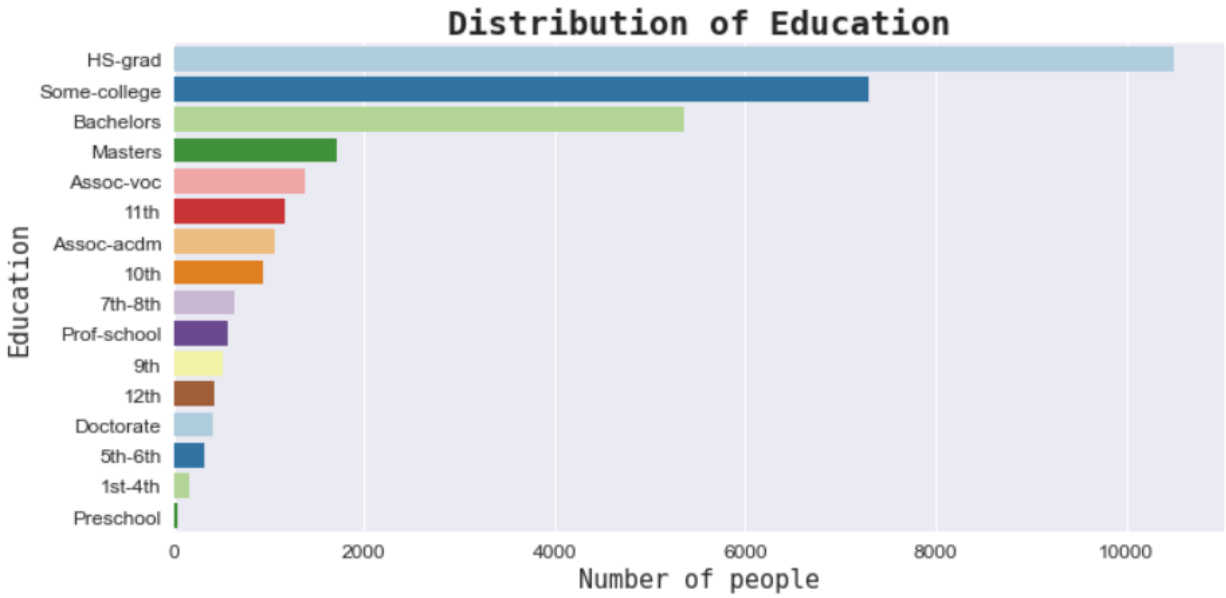

Fig 5.3 Distribution of Education

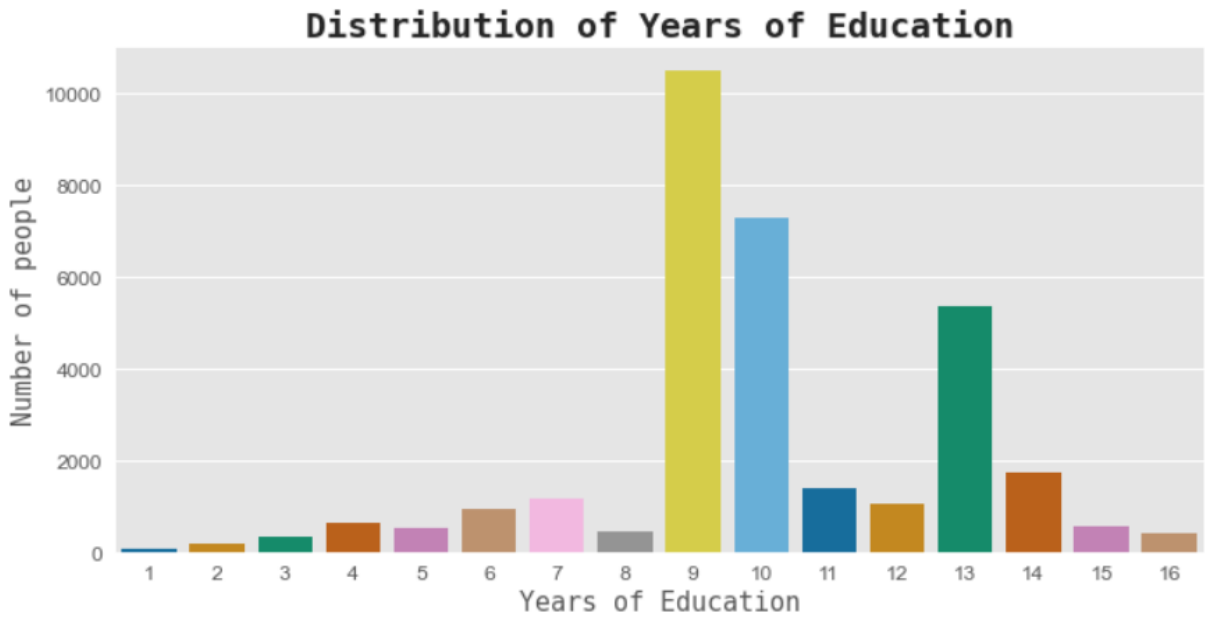

Fig 5.4 Distribution of years of education

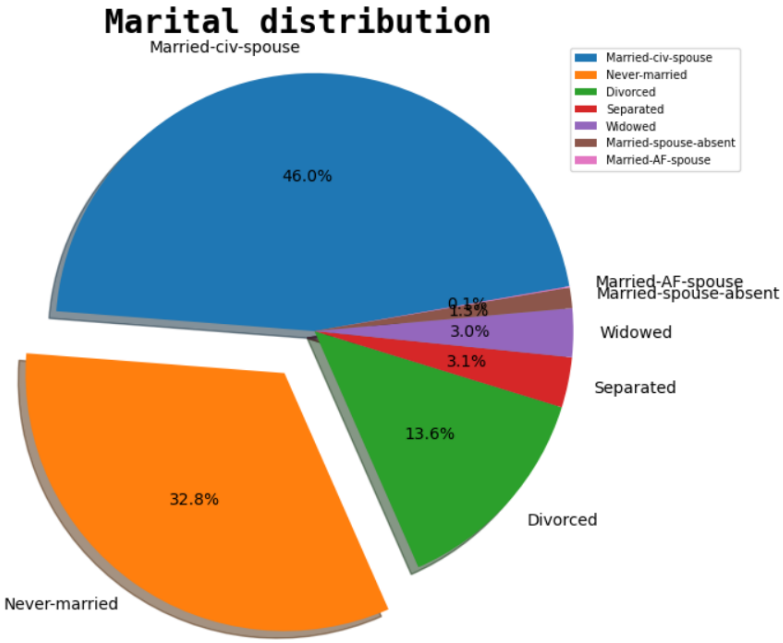

Fig 5.5 Marital distribution.

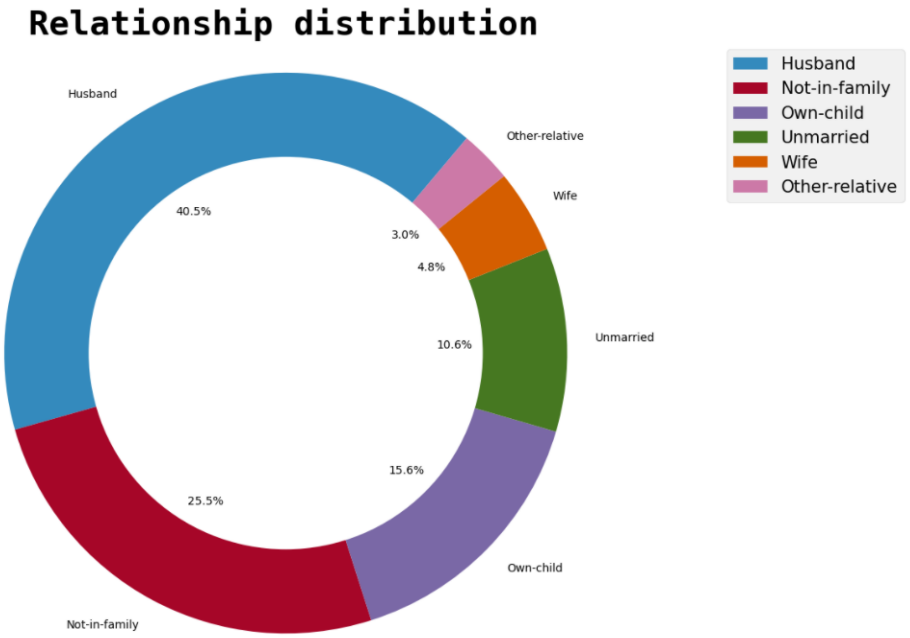

Fig 5.6 Relationship Distribution

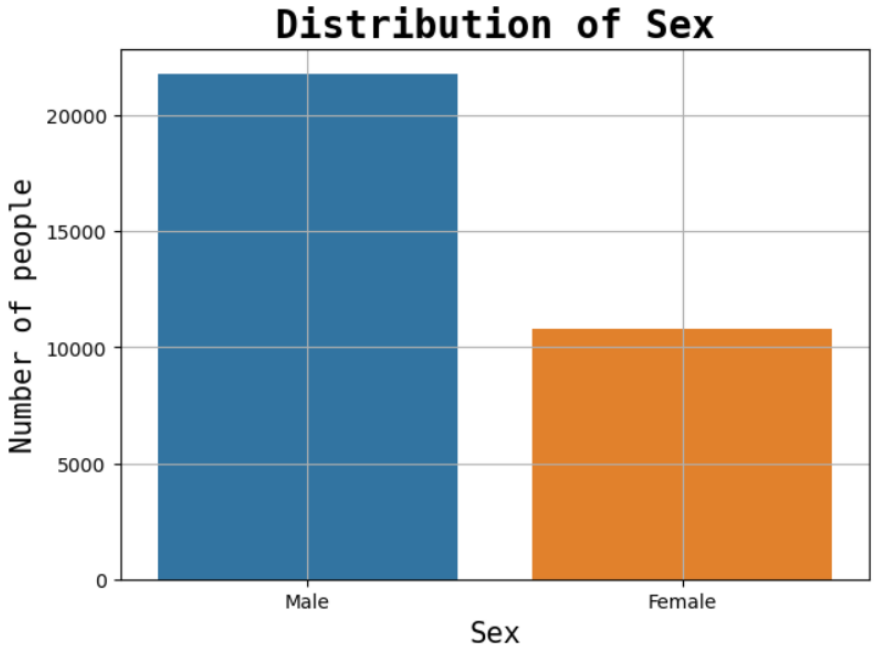

Fig 5.7 Distribution of sex

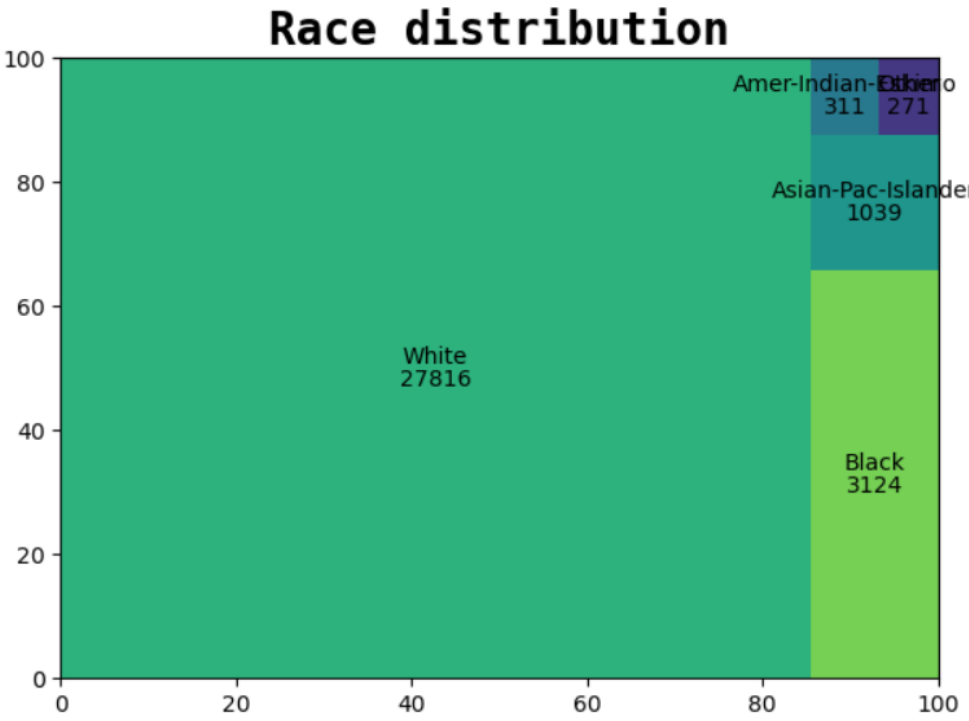

Fig 5.8 Race Distribution

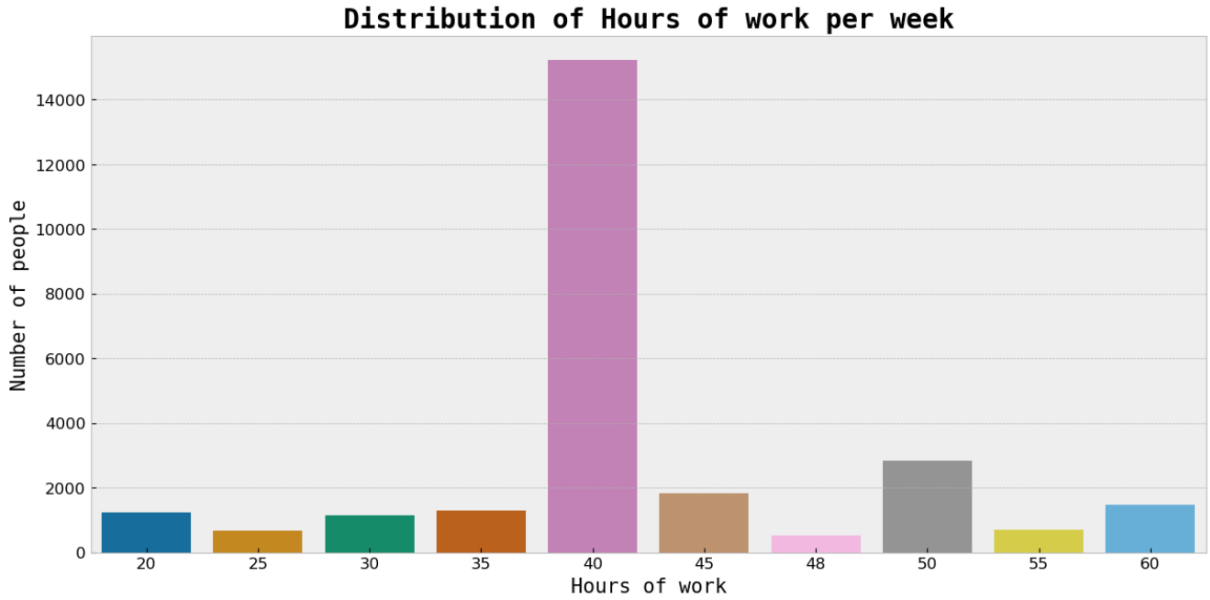

Fig 5.9 Distribution of working hours per week

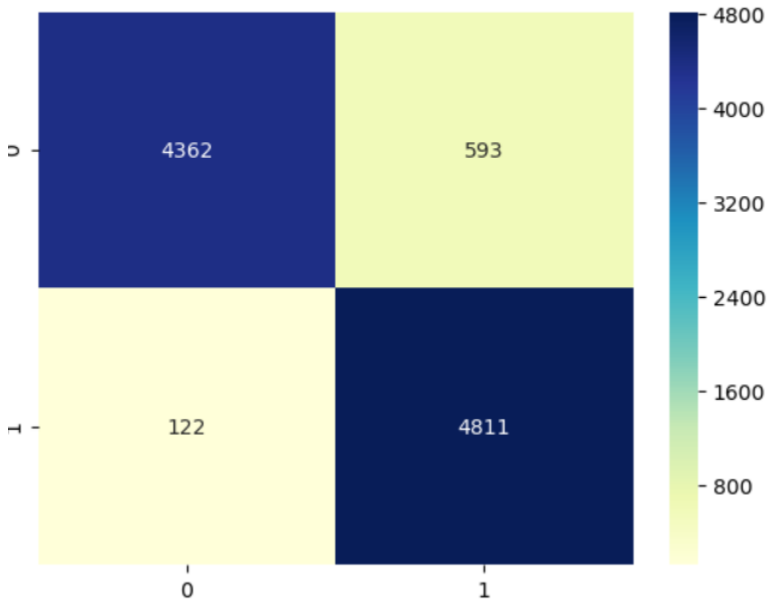

Fig 5.10 confusion matrix

```
In [94]: from sklearn.metrics import classification_report
         print(classification_report(Y_test, Y_pred_rf_best))
```

	precision	recall	f1-score	support
0	0.97	0.88	0.92	4955
1	0.89	0.98	0.93	4933
accuracy			0.93	9888
macro avg	0.93	0.93	0.93	9888
weighted avg	0.93	0.93	0.93	9888

Fig 5.11 accuracy score

Chapter 6

6.1. Conclusion

We have successfully implemented the proposed model. The project is designed keeping in view the problems existing due to income inequality. Adult Income Analysis has to do with analysing income of individuals to gain insights. In this project, a system that can be used to aid all types of researchers that want to learn about EDA, data preparation, feature selection and data modelling. We tried to predict the income of users who earned above 50k and where able to predict with 93percent accuracy.

6.2. Limitations of the Project

- Project explores the basics of big data analysis and concepts here are for beginners for those who are getting into the field.

- Project if implemented in a local system can be process intensive and might require high processing capacity. We have used online free servers to implement the project.

- Project cannot be used for other datasets as data preparation and modelling is specifically done for this dataset.

- Project does not go into advanced data models like neural networks and RNN for better accuracy because it would complicate the project and goal was to do big data analysis.

6.3. Future Enhancement.

- There is a chance for accuracy improvements with better modelling and advanced deep learning models like artificial neural networks and recurrent neural networks.

- Project doesn't go deep in parameter tuning as that is also a good way to increase a little accuracy.

- New research is being conducted on feature selection algorithms and those algorithms can help in better feature selection.

- If the project is to be presented to non programming background individuals having some GUI could help them understand the project .

- Project can be open sourced in platforms like Github and kaggle that accept contributions from other fellow big data enthusiasts to enhance the project.

- We can explore census data from other countries to analyse the income distribution in other countries.

References

1] Kaggle Adult Census Income Data Set -
https://www.kaggle.com/uciml/adult-census-income

[2] Logistic Regression -
https://www.statisticssolutions.com/what-islogistic-regression/

[3] Random Forest Classifier -
https://towardsdatascience.com/understanding-random-forest58381e0602d2

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

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

[6] 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.