INSTITUTE FOR ADVANCE COMPUTING AND SOFTWARE DEVELOPMENT AKURDI, PUNE.

Documentation On,

“A GENDER PAY GAP”

e-DBDA MAY 2021

Submitted By:

Group No: 05
BHAGYASHRI GAWAS **(1311)**
PRACHI PANDE **(1341)**

Mr. Prashant Karhale
Centre Coordinator

Mr. Akshay Tilekar
Project Guide

A

PROJECT REPORT ON 'A GENDER PAY GAP'

"A GENDER PAY GAP"

Submitted by

Group No: 05

BHAGYASHI GAWAS **(1311)**

PRACHI PANDE **(1341)**

Centre Coordinator

Mr. PRASHANT KARHALE

Under the Guidance of

Mr. AKSHAY TILEKAR

In the fulfillment of e – Diploma in Big Data Analytics course from Institute for Advance Computing and Software Development Akurdi, Pune in the academic year

May 2021 – Sept.2021

e-DBDA
May 2021

Institute for Advance Computing and Software Development
Akurdi, Pune. 411044

ACKNOWLEDGEMENT

It gives us immense pleasure to present our report for project on **“A GENDER PAY GAP”** The able guidance of all teaching staff of this department made the study possible. They have been a constant source of encouragement throughout this project. We would like to express our grateful thanks to **Mr. Prashant Karhale Sir, Mr. Akshay Tilekar Sir** who guided us properly for this project. We would also like to express our sincere thanks to Institute for Advance Computing and Software Development Akurdi, Pune for giving us an opportunity to explore the subject and use our knowledge by conducting this project.

Group No: 05

Bhagyashree Gawas (1311)

Prachi Pande (1341)

e- DBDA, May 2021

Institute for Advance Computing and Software Development, Akurdi

PROJECT OVERVIEW

To analyze the difference in between the wages in gender male and female. We are trying to check which factors are responsible for causing the wages gap in male and female. In this we did our analysis on Income where income were to destruct to perform any analysis on it. So we create bin creation and make 5 classes on which we will conclude our analysis and label encoder for those column who were categorical. Implementation with the ensemble learning model random forest, ada boost, XG boost, Decision Tree machine learning classifier with data visualization. We applied fine tuning on ada boost and XG boost to increase the accuracy of this models. This project aims to develop machine Learning model for detection pf pay gap with suitable feature.

CONTENTS

List of Figures

Chapter - 1

INTRODUCTION

1.1 PROBLEM STATEMENT

The gender pay gap is observation to the difference in earnings between men and women. It has been observed that Women are get less earning than men, and the gap is much for most women in their sector. The wage gaps for each group are calculated based on earnings data from the U.S. dataset and this represent each individual woman’s information. People living in society such as transgender women and immigrant women also experience the negative effects of multiple biases on their earnings. But there are also more factors that impact on the wages on earing for women.

1.2 Abstract

These pay gap are calculated with the ratio of earnings for women and men across all features on what factor they are earning, they do not reflect a direct comparison of women and men doing identical work. In this analysis we are calculating the accuracy of the income with the factor which are affecting on earning in male and female. Men and women do different jobs. For example 85 per cent of engineers are male, while 75 per cent of primary-school/colleges teachers are female. The Statistics estimates that this is responsible for 36 per cent reason for wages gap.

We are applying the analytical strategic formulae’s to explain the gap between the both aspirants. In which we can used two type of strategical methods. That are regressor and classifier. But as refer to our dataset we create the multiple class. So as we will work on classification. In our project we are using python so we can apply machine learning model in which we will calculate result on basis of strategic formulae.

1.3 Aims & Objectives

we are using gender pay gap dataset from vincentare bundlock consisting of 61k respondents’ records. This data has contained features like occupation, education, work status, gender etc. Usually, a gender pay gap is affected by education, occupation gender and age. The primary goal of this project is to analysis gender pay gap and predict income according to features. before predicting the output, training will be done using the bunch of machine learning models which then compared by their accuracies. machine learnings that are applied to predict income on dataset are Ada boost, XG boost and random forest. the best model will be selected based on their accuracies.

GLOSSARY

Training data: The data that are used to train classification models.

Validation data: The data that are used to test the performance of the classification model during the training process. Validation data are used to fine tune parameters in the classification model and the data should not be part of the training data.

Testing data: The data that are used to test the performance of the trained classification model (after training process). Testing data are used to evaluate the final performance of the classification model. Testing data should not be part of training data or validation data. No fine tune should be made based on the result of testing data.

Overfitting: The classification model performs consistently better on training data than on validation data and testing data.

OBJECTIVE

- Develop an ensemble learning model for a gender pay gap.
- Analyse the model accuracy to better understand the important features for fall and rise of activities. Based on this understanding, validate, and further improve the performance of the model on activity detection.
- Based on the result of the ensemble learning model, develop a model design for gender pay gap.

4. Overall Description

4.1 Workflow of Project:

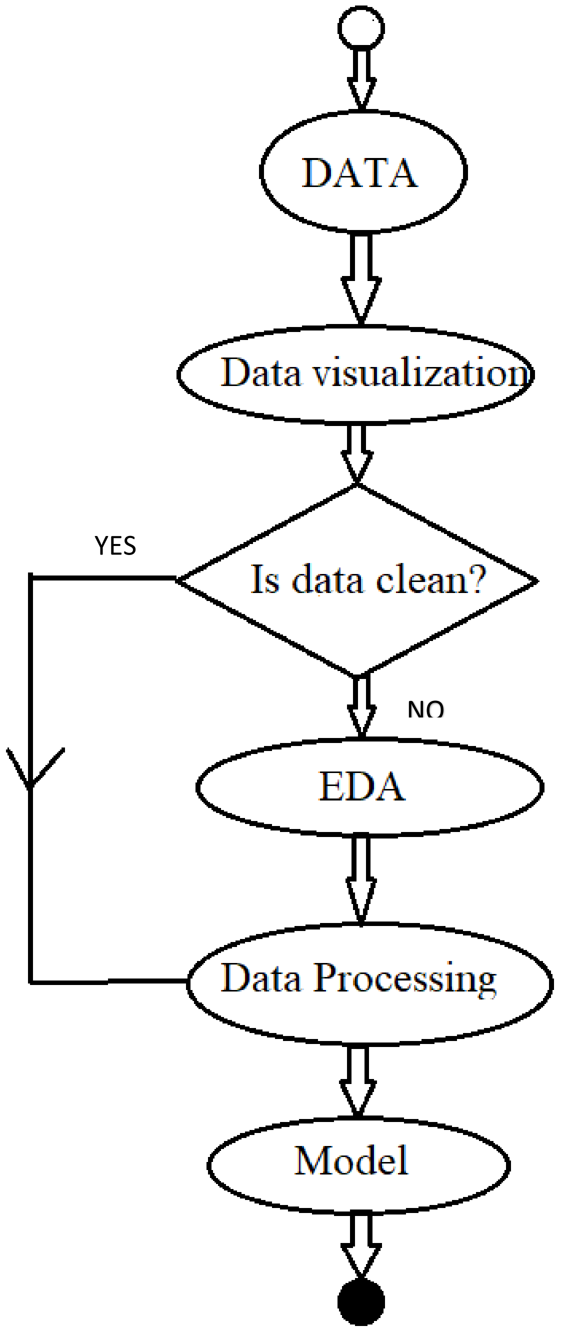


Fig. 1.1

4.2 Data Preprocessing and Cleaning:

4.2.1 Data Cleaning:

The dataset which we are using 11 features in which there are many NA values so we have to handle these null values with the data cleaning.

1. Removed null values

In the dataset target column was Income which is dependant attribute. That's why we dropped all null values from that attribute.

2. Replace null values

The attribute prestige code represents score for work status and the higher number of null values were from housekeeper. So, on the basis of work status of housekeeper we replaced the null values with the medium of rest housekeeper’s prestige score. The attribute child is related to gender and age so on the basis of same gender and same age respondents we take the median and replace with null values of child. The attribute age is related to gender and child so on the basis of same gender and child respondents we take the median and replace with null values of age.

3. Bin creation

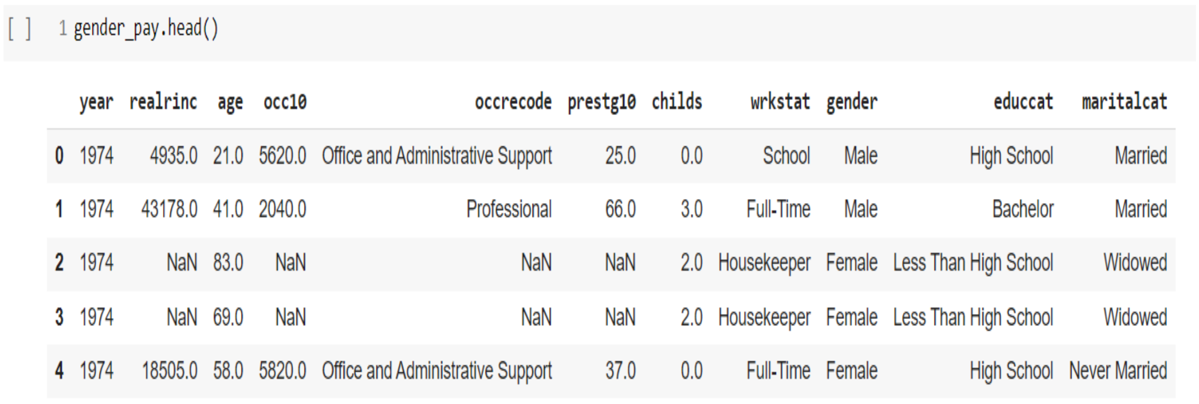
The Analysis is done in two ways regressor and classifier. The dataset we were using was having the target Income it was continuous but for particulate the result we need to divide the income into multiclass bins. They are classified in the ways like the income (very low, low, moderate, high , superhigh).

4.2.2 Label encoding:

To make the data understandable or in human readable form, the training data is often labeled in words. Label Encoding refers to converting the labels into numeric form so as to convert it into the machine-readable form. Machine learning algorithms can then decide in a better way on how those labels must be operated.

4.3 Exploratory Data Analysis:

Exploratory Data Analysis refers to the critical process of performing initial investigations on data so as to discover patterns, to spot anomalies, to test hypothesis and to check assumptions with the help of summary statistics and graphical representations.

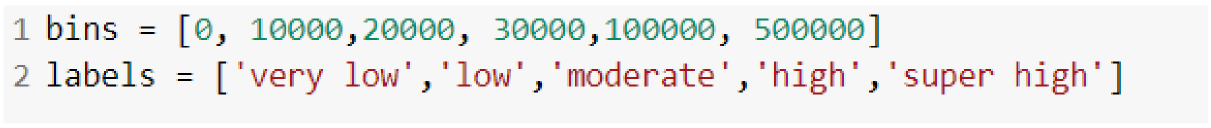


```
[ ]  gender_pay.head()
```

	year	realrinc	age	occ10	occrecode	prestg10	childs	wrkstat	gender	educcat	maritalcat
0	1974	4935.0	21.0	5620.0	Office and Administrative Support	25.0	0.0	School	Male	High School	Married
1	1974	43178.0	41.0	2040.0	Professional	66.0	3.0	Full-Time	Male	Bachelor	Married
2	1974	NaN	83.0	NaN	NaN	NaN	2.0	Housekeeper	Female	Less Than High School	Widowed
3	1974	NaN	69.0	NaN	NaN	NaN	2.0	Housekeeper	Female	Less Than High School	Widowed
4	1974	18505.0	58.0	5820.0	Office and Administrative Support	37.0	0.0	Full-Time	Female	High School	Never Married

Fig 1.2

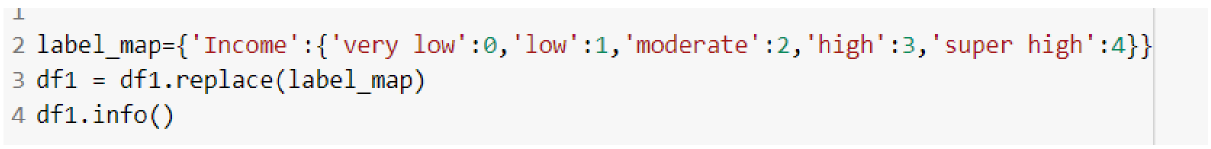
In fig 1.2, There are null values which are replaced by the multiple data cleaning process.



```
bins = [0, 10000,20000, 30000,100000, 500000]
labels = ['very low','low','moderate','high','super high']
```

Fig 1.3

In fig 1.3, Bins are created according to income of respondents



```
label_map={'Income':{'very low':0,'low':1,'moderate':2,'high':3,'super high':4}}
df1 = df1.replace(label_map)
df1.info()
```

Fig 1.4

In fig 1.4, Label encoding is done on the multiple classes created during fig 1.2

```
from sklearn.preprocessing import LabelEncoder
for c in col:
    lbl = LabelEncoder()
    lbl.fit(list(df1[c].values))
    df1[c] = lbl.transform(list(df1[c].values))
```

Fig 1.5

In fig 1.5, Label Encoder is imported from sklearn preprocessing to those features which are categorical. For perform machine learning model the encoding is used on categorical values for better performance.

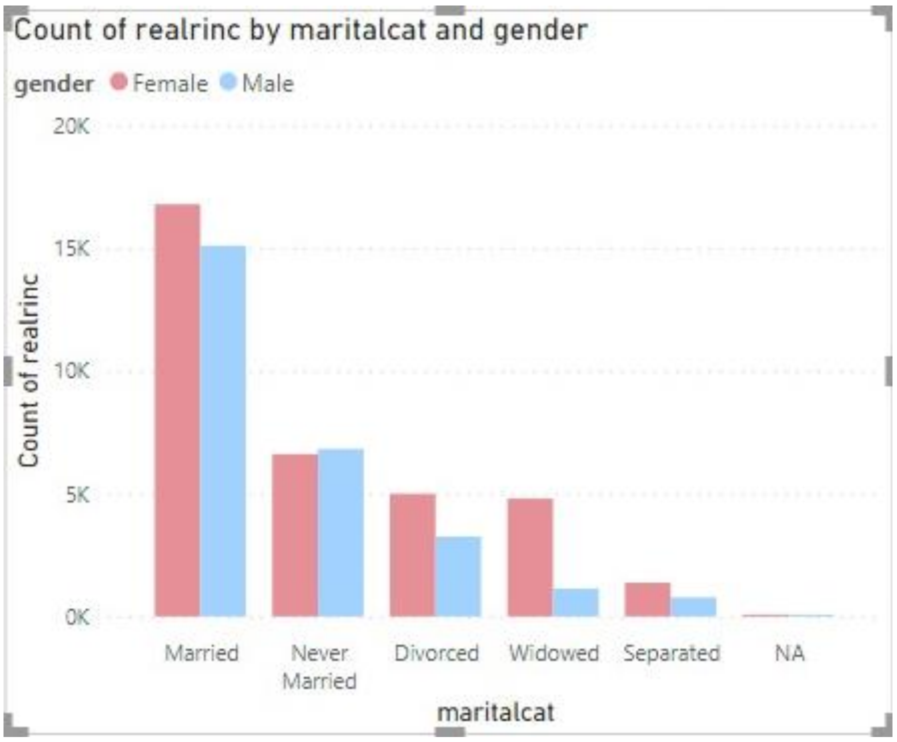


Fig 1.6

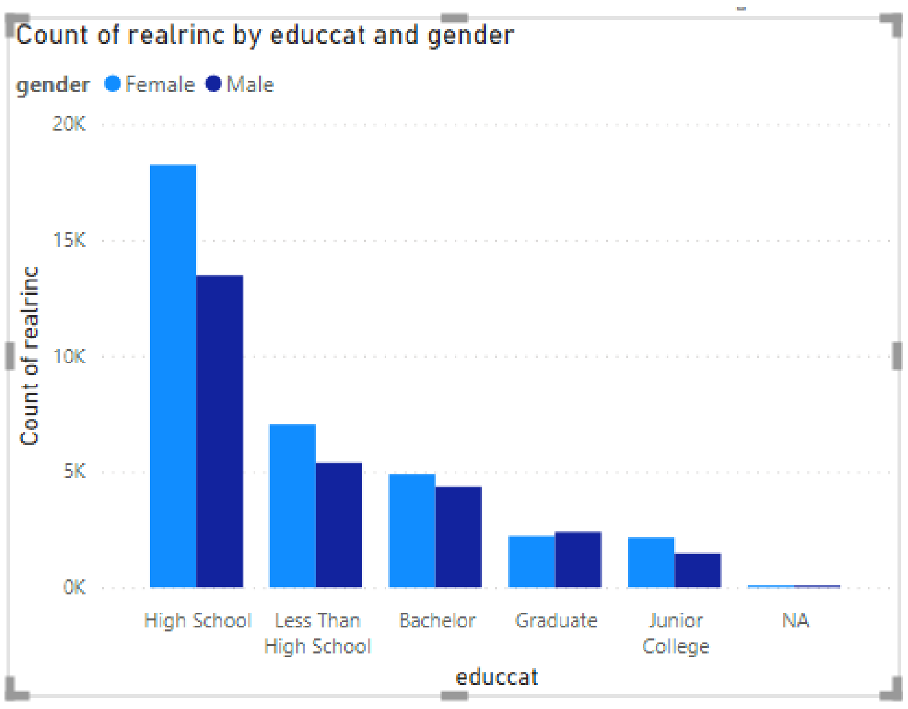

Count of realrinc by educcat and gender
gender ● Female ● Male
20K
15K
10K
5K
0K
Count of realrinc
High School
Less Than High School
Bachelor
Graduate
Junior College
NA
educcat

Fig 1.7

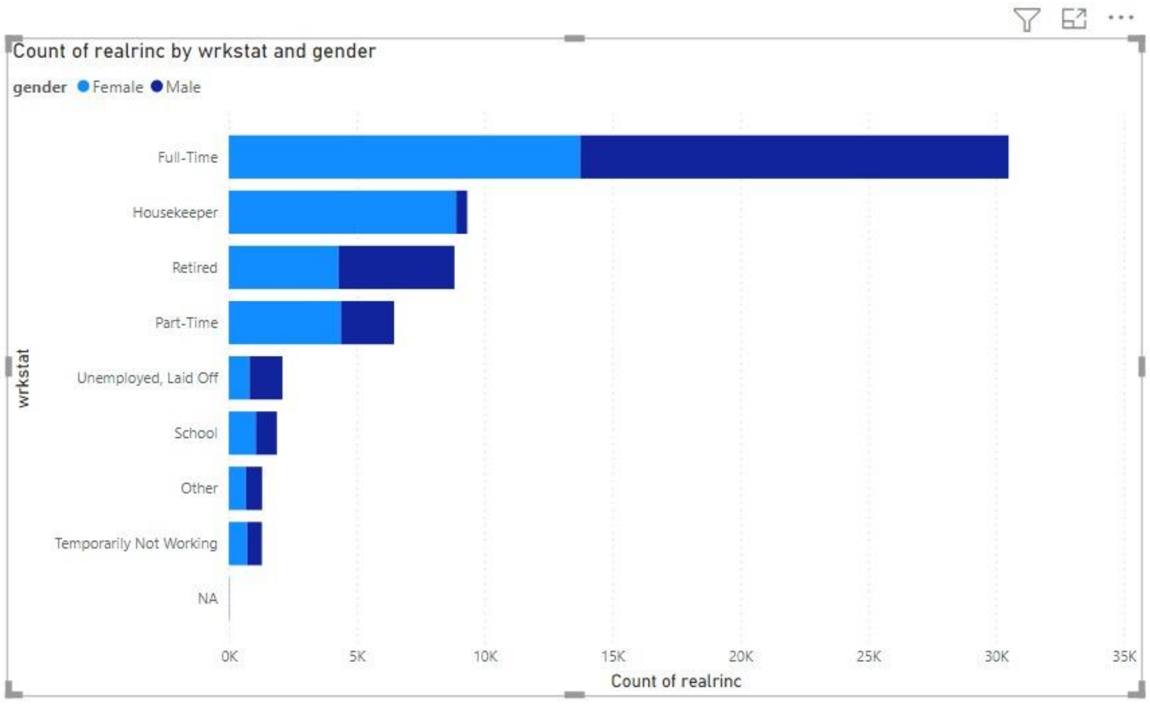

Count of realrinc by wrkstat and gender
gender ● Female ● Male
Full-Time
Housekeeper
Retired
Part-Time
Unemployed, Laid Off
School
Other
Temporarily Not Working
NA
wrkstat
0K
5K
10K
15K
20K
25K
30K
35K
Count of realrinc

Fig 1.8

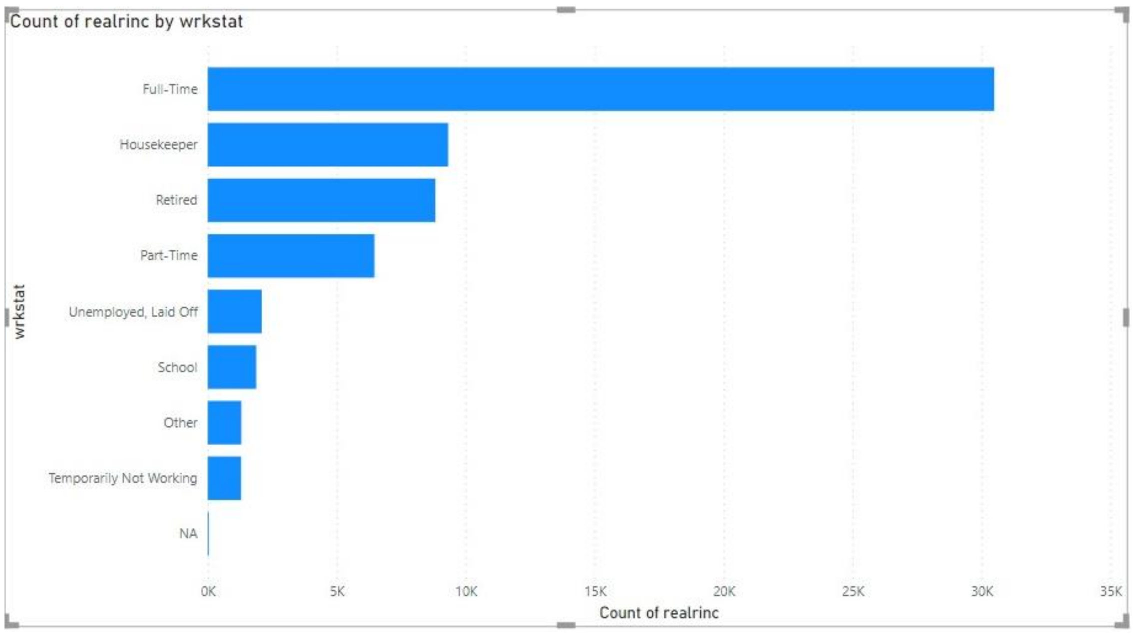


Fig 1.9

```
sns.set_theme(style="ticks", color_codes=True)
fig_dims = (16, 4)
fig, ax = plt.subplots(figsize=fig_dims)
sns.barplot(x = "occupation", y = "Income", ax=ax, data=df)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f7385489750>
```

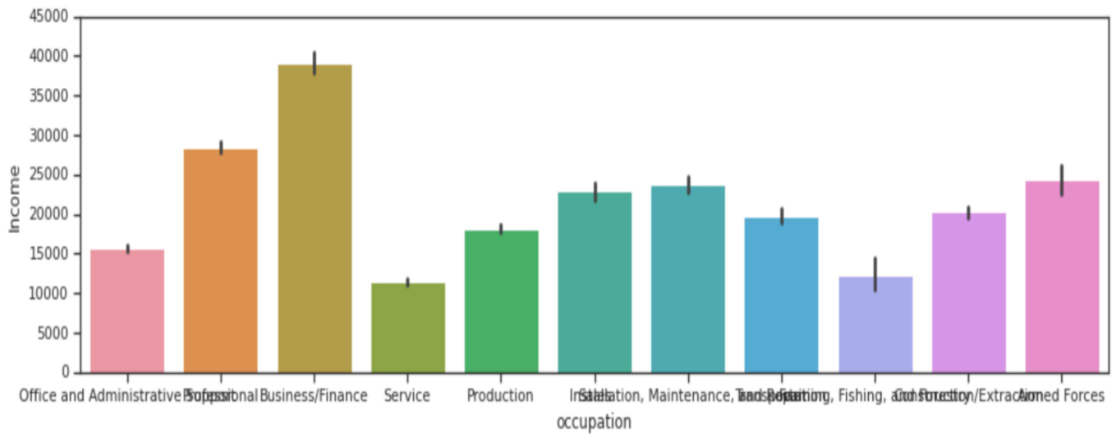


Fig 1.10

```
sns.catplot(x="education", y="Income", hue="gender", kind="bar", data=df)
```

```
<seaborn.axisgrid.FacetGrid at 0x7f738de62890>
```

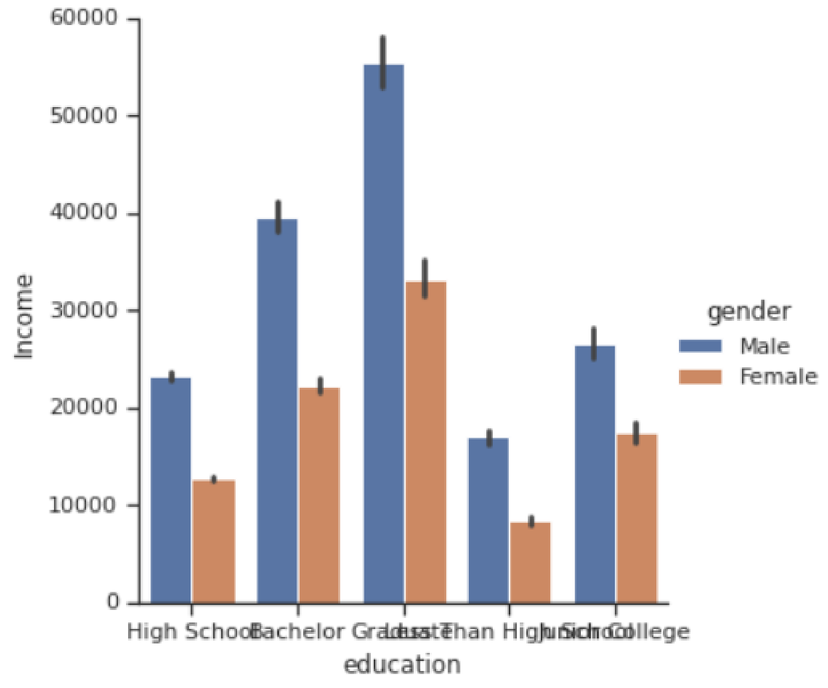


Fig 1.11

4.4 Model Building:

1. Train/Test split:

One important aspect of all machine learning models is to determine their accuracy. Now, in order to determine their accuracy, one can train the model using the given dataset and then predict the response values for the same dataset using that model and hence, find the accuracy of the model. A better option is to split our data into two parts: first one for training our machine learning model, and second one for testing our model.

- Split the dataset into two pieces: a training set and a testing set.
- Train the model on the training set.
- Test the model on the testing set, and evaluate how well our model did.

Advantages of train/test split:

- Model can be trained and tested on different data than the one used for training.
- Response values are known for the test dataset, hence predictions can be evaluated
- Testing accuracy is a better estimate than training accuracy of out-of-sample performance.

Machine learning consists of algorithms that can automate analytical model building. Using algorithms that iteratively learn from data, machine learning modelsfacilitate computers to find hidden insights from Big Data without being explicitly programmed where to look.

We have used the following three algorithms to build predictive model.

Oversampling:

Imbalanced datasets are referring as where there is a severe skew in the class distribution. This bias in the training dataset can influence many machine learning algorithms which lead it to ignore the minority class entirely. Typically, the minority class on which predictions are most important. Imbalanced classification involves developing predictive models on classification datasets that have a severe class imbalance. The working with imbalanced datasets is that most machine learning techniques will ignore and it could have poor performance on model.

The minority class, although typically it is performance on the minority class that is most important. One of the approaches to overcome imbalanced datasets is to oversample the minority class. The simplest approach involves duplicating samples in the minority class. New samples can be synthesized from the existing samples. This is a type of data augmentation for the minority class and is referred to as the Synthetic Minority Oversampling Technique, or SMOTE. SMOTE helps to correctly fit and evaluate machine learning models on SMOTE-transformed training datasets.

Ensemble learning Models

Ensemble learning is explained as to combine the predictions from two or more models. There are three classes of ensemble learning techniques that are most commonly discussed and used in machine learning.

1. Bagging is used to fit many decision trees on different samples of the same dataset and averaging the predictions. Each model gets assign with the weight on the summation of weights and predicted value the model accuracy is considered as a result.
2.Boosting is used by adding ensemble members sequentially that correct the predictions made by previous models and outputs a weighted average of the predictions.
3.Stacking is used to fit many different model’s types on the same data and using another model to learn how to best combine the predictions.

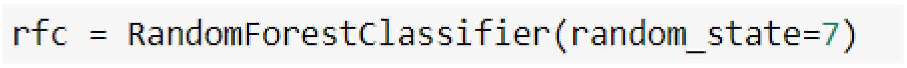
4.4.1. Random Forest Classifier

Random forests are one the most popular machine learning algorithms. Random forests consist of hundred decision trees which is default value, each of them built over a random extraction of the observations from the dataset and a random extraction of the features. A random subset of the features is taken into as one dataset by the algorithm for splitting a node. It make trees more randomly by additional using randomly selecting each feature rather than searching for the best possible features.

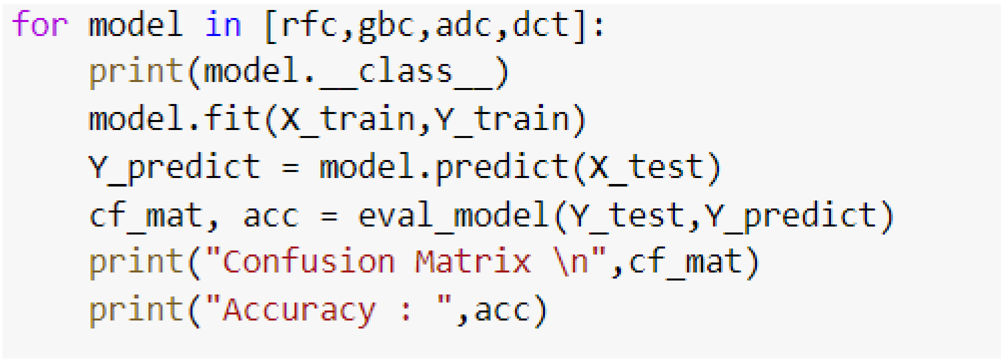
Random forest is a flexible algorithm which gives best result without fine tuning the reason random forest is popular because of its simplicity and diversity. It can be used for both classification and regression. The forest word in algorithm stands for multiple trees which are usually trained with bagging method. The bagging stands for a combination of learning models which increases the overall result. This algorithm establishes the outcome based on predictions of the decision trees. Increasing the number of trees increases the precision of the outcome. It has nearly same hyperparameters as a decision tree or bagging classifier. In random forest, while growing the trees it searches for the best feature among a random subset of features. This technique results in a better performance of model.

Features of a Random Forest Algorithm

- It’s more accurate than the decision tree algorithm.
- It provides an effective way of handling missing data.
- It can produce a reasonable prediction without hyper-parameter tuning.
- It solves the issue of overfitting in decision trees.
- In every random forest tree, a subset of features is selected randomly at the node’s splitting point.

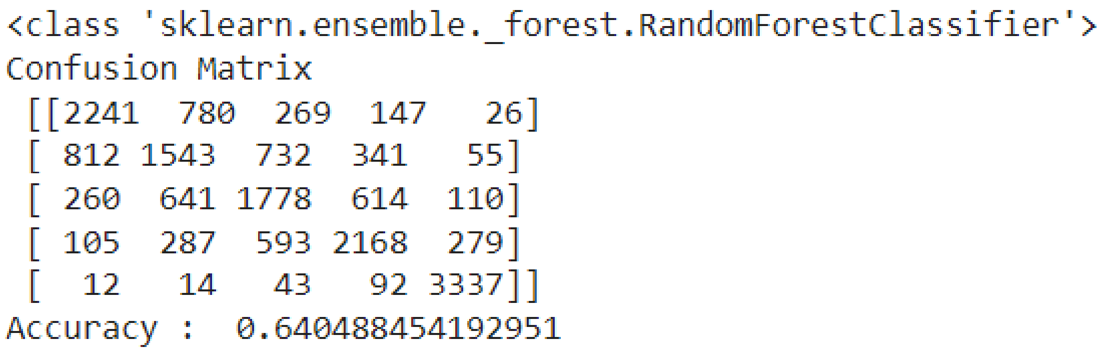


```
rfc = RandomForestClassifier(random_state=7)
```



```
for model in [rfc,gbc,adc,dct]:
    print(model.__class__)
    model.fit(X_train,Y_train)
    Y_predict = model.predict(X_test)
    cf_mat, acc = eval_model(Y_test,Y_predict)
    print("Confusion Matrix \n",cf_mat)
    print("Accuracy : ",acc)
```

Fig 2.0



```
<class 'sklearn.ensemble._forest.RandomForestClassifier'>
Confusion Matrix
 [[2241  780  269  147   26]
 [ 812 1543  732  341   55]
 [ 260  641 1778  614  110]
 [ 105  287  593 2168  279]
 [  12   14   43   92 3337]]
Accuracy :  0.640488454192951
```

Fig 2.1

4.4.2. AdaBoost Classifier

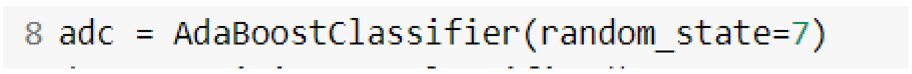
AdaBoost algorithm is also named as adaptive boosting. Boosting technique used as an ensemble Method in Machine Learning. In which weights are re-assigned to each model, with higher weights assigned gets more priories as classified model. Boosting is used to reduce bias and also variance for supervised learning. It works with the principle of model improving sequentially. Except for the first, rest all subsequent model is improved from previously improved model. Predictions are made by calculating the weighted average of the weak classifiers.

Ada boost can be used to boost the performance of any machine learning algorithm. It stands for adaptive boosting. It focuses on classification problems and aims to convert a set of weak classifiers into strong one. In simple words weak learner are converted into strong ones. It is done by using weak model series. First model is build based on train data. Then the second model is built which tries to correct errors based on fisrt models. In short it updates weights based on errors. AdaBoost algorithms can be used for both classification and regression problem. This procedure is continued and models are added until the complete training data set is predicted correctly or the maximum number of models are added. The most suited algorithm used with ada boost is decision tree with one level. The reason for this is these trees are short and only contain one decision for classification.

Each instance in the training dataset is weighted. The initial weight is set to:

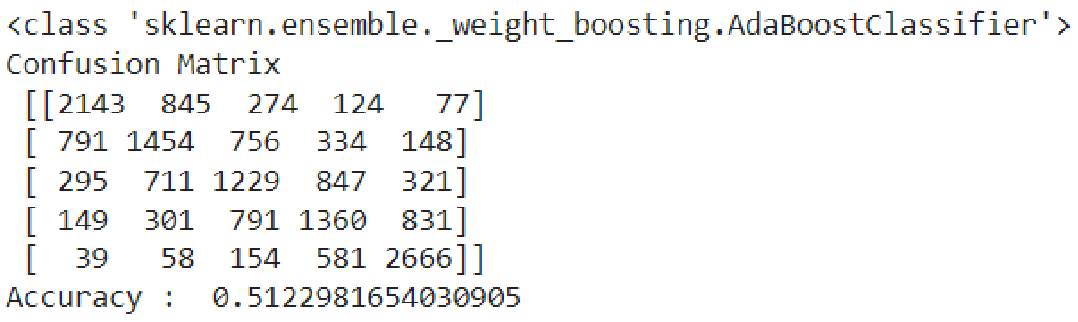
weight(xi) = 1/n

Where xi is the i’th training instance and n is the number of training instances.



```
adc = AdaBoostClassifier(random_state=7)
```

Fig 2.2



```
<class 'sklearn.ensemble._weight_boosting.AdaBoostClassifier'>
Confusion Matrix
 [[2143  845  274  124   77]
 [ 791 1454  756  334  148]
 [ 295  711 1229  847  321]
 [ 149  301  791 1360  831]
 [  39   58  154  581 2666]]
Accuracy :  0.5122981654030905
```

Fig 2.3

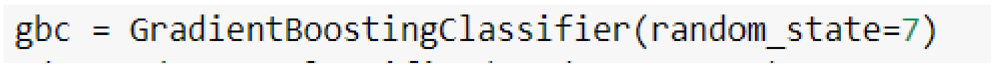
4.4.3. XG Boost Classifier

XG boost is the most widely used algorithm in machine learning, whether the problem is a classification or a regression problem. Boosting is an ensemble techniques where previous model errors are resolved in the new models. These models are added until no other improvement is required. Gradient boosting is a method where the new models are created that computes the error in the previous model and then rest is added to make the final prediction. It uses a gradient descent algorithm that is the reason it is called a Gradient Boosting Algorithm. The prediction could be any classification or regression methods that fully supported for both types of predictive modelling problems.

The models that form the ensemble, also known as base learners, could be either from the same learning algorithm or different learning algorithms. Bagging and boosting are two widely used ensemble learners. Though these two techniques can be used with several statistical models, the most predominant usage has been with decision trees.

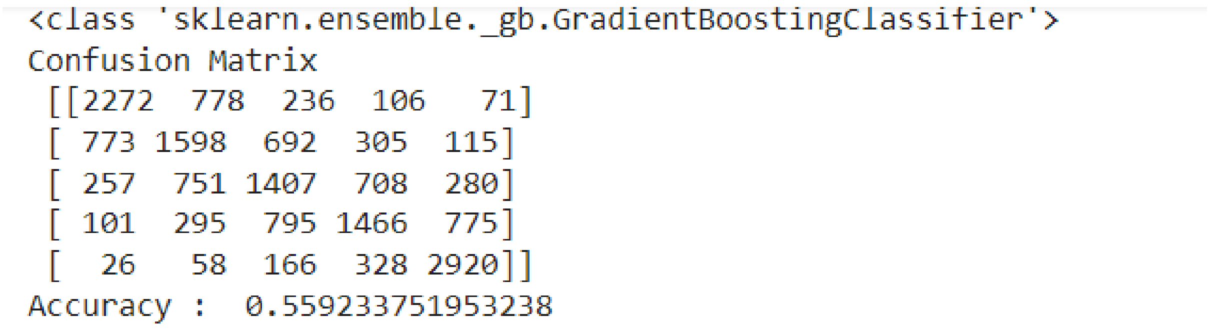
Regularization: XG Boost has an option to penalize complex models through both L1 and L2 regularization. Regularization helps in preventing overfitting

Handling sparse data: Missing values or data processing steps like one-hot encoding make data sparse. XG Boost incorporates a sparsity-aware split finding algorithm to handle different types of sparsity patterns in the data



```
gbc = GradientBoostingClassifier(random_state=7)
```

Fig 2.4

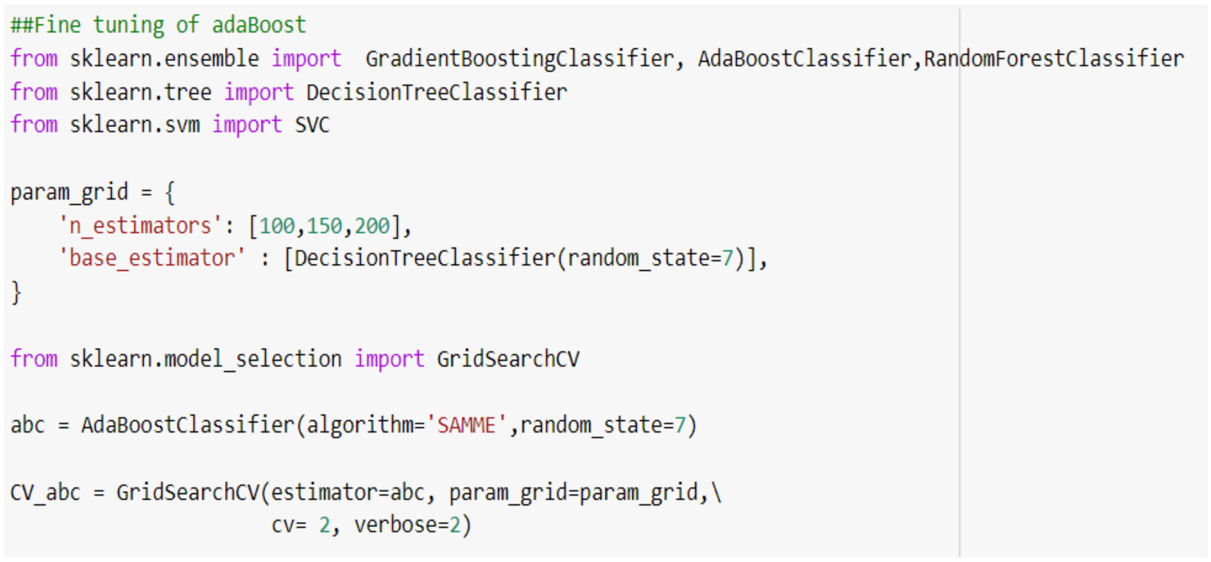


```
<class 'sklearn.ensemble._gb.GradientBoostingClassifier'>
Confusion Matrix
 [[2272  778  236  106   71]
 [ 773 1598  692  305  115]
 [ 257  751 1407  708  280]
 [ 101  295  795 1466  775]
 [  26   58  166  328 2920]]
Accuracy :  0.559233751953238
```

Fig 2.5

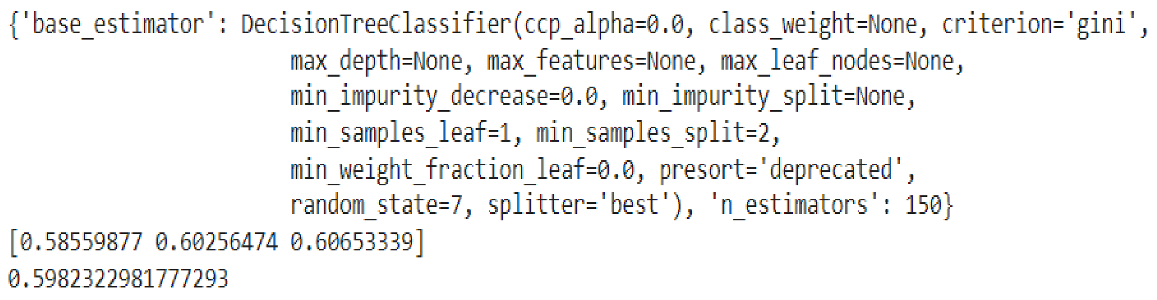
Fine Tuning:

Fine-tuning is the process which works on parameters of a model that they should be very precisely in order to fit with observations which are best in. Fine tuning is a crucial step in machine learning predictive model to improve accuracy of the predictive results. it is used to decompose the feature and set it to improve accuracy of your machine learning models. Retrieving best feature performance of a model’s using scoring of its data values.



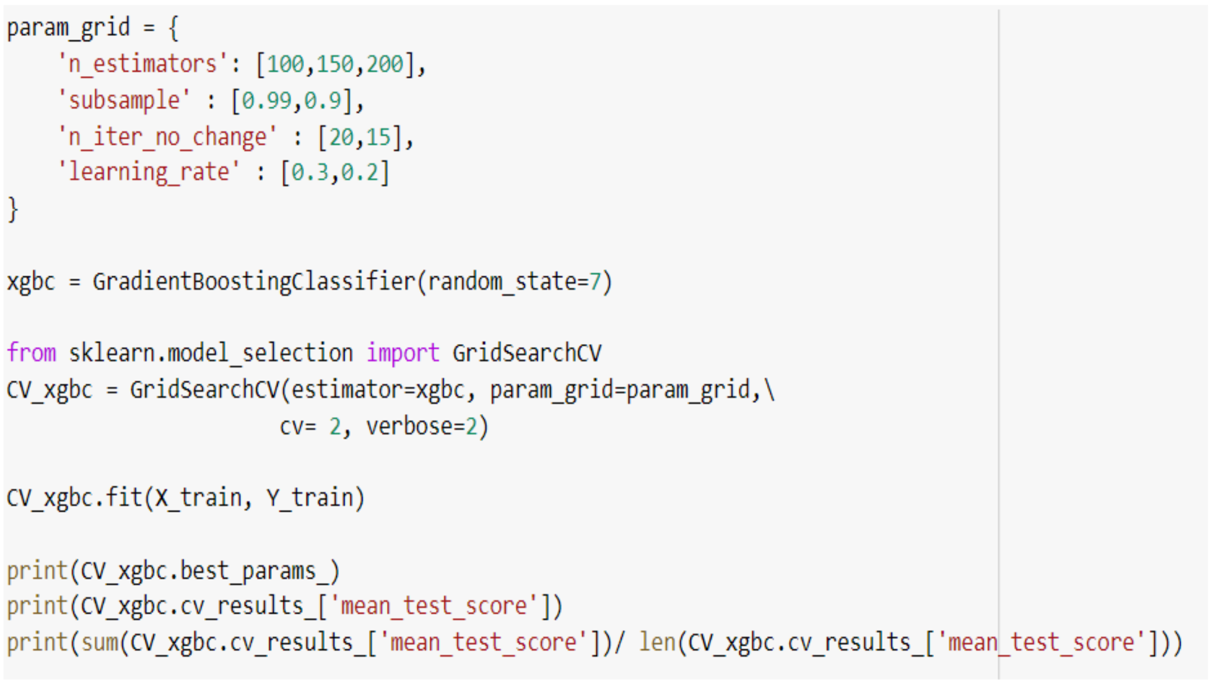
```
##Fine tuning of adaBoost
from sklearn.ensemble import  GradientBoostingClassifier, AdaBoostClassifier,RandomForestClassifier
from sklearn.tree import DecisionTreeClassifier
from sklearn.svm import SVC

param_grid = {
    'n_estimators': [100,150,200],
    'base_estimator' : [DecisionTreeClassifier(random_state=7)],
}

from sklearn.model_selection import GridSearchCV

abc = AdaBoostClassifier(algorithm='SAMME',random_state=7)

CV_abc = GridSearchCV(estimator=abc, param_grid=param_grid,\
                      cv= 2, verbose=2)
```

Fig 2.6



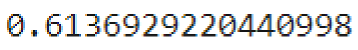
```
{'base_estimator': DecisionTreeClassifier(ccp_alpha=0.0, class_weight=None, criterion='gini',
                       max_depth=None, max_features=None, max_leaf_nodes=None,
                       min_impurity_decrease=0.0, min_impurity_split=None,
                       min_samples_leaf=1, min_samples_split=2,
                       min_weight_fraction_leaf=0.0, presort='deprecated',
                       random_state=7, splitter='best'), 'n_estimators': 150}
[0.58559877 0.60256474 0.60653339]
0.5982322981777293
```

Fig 2.7



```
param_grid = {
    'n_estimators': [100,150,200],
    'subsample' : [0.99,0.9],
    'n_iter_no_change' : [20,15],
    'learning_rate' : [0.3,0.2]
}

xgbc = GradientBoostingClassifier(random_state=7)

from sklearn.model_selection import GridSearchCV
CV_xgbc = GridSearchCV(estimator=xgbc, param_grid=param_grid,\
                       cv= 2, verbose=2)

CV_xgbc.fit(X_train, Y_train)

print(CV_xgbc.best_params_)
print(CV_xgbc.cv_results_['mean_test_score'])
print(sum(CV_xgbc.cv_results_['mean_test_score'])/ len(CV_xgbc.cv_results_['mean_test_score']))
```

Fig 2.8



```
0.6136929220440998
```

Fig 2.9

Decision Trees:

The decision tree Algorithm belongs to the family of supervised machine learning algorithms. It can be used for both a classification problem as well as for regression problem.

The goal of this algorithm is to create a model that predicts the value of a target variable, for which the decision tree uses the tree representation to solve the problem in which the leaf node corresponds to a class label and attributes are represented on the internal node of the tree

-In the beginning, we consider the whole training set as the root.

-Feature values are preferred to be categorical, if the values continue then they are converted to discrete before building the model.

-Based on attribute values records are distributed recursively. We use a statistical method for ordering attributes as a root node or the internal node. Entropy: Entropy is the measures of impurity, disorder or uncertainty in a bunch.

“Entropy values range from 0 to 1”, Less the value of entropy more it is trusting able.

5. Requirements Specification

5.1 Hardware Requirement:

- 500 GB hard drive (Minimum requirement)
- 8 GB RAM (Minimum requirement)
- PC x64-bit CPU

5.2 Software Requirement:

3.1.1 Windows/Mac/Linux

3.1.2 Python-3.9.1

3.1.3 Python Extension for VS Code

3.1.4 Libraries:

3.1.4.1 NumPy 1.18.2

3.1.4.2 Pandas 1.2.1

3.1.4.3 Matplotlib 3.3.3

3.1.4.4 Scikit-learn 0.24.1

FUTURE SCOPE

A gender pay gap is more global issue where there are many sections which are affecting by it. Men and women working in different fields. The differences in work hours, with more women working part-time and men putting in more overtime. And part of it is because women take more time off for child care and other family responsibilities. Lower pay for women makes it harder especially for single women to get financially stable. Their lower earnings make it harder for them to save money for emergencies or retirement. But the impact of the gender pay gap isn’t limited up to women. It also affects their families. Strategies to decrease the gender pay gap include pay transparency, expanding paid family and medical leave, and improving work-life balance.

This project can be used to determine gender pay gap and how it varies by year-to-year. As gender pay gap is the global issue so it will help to determine what features are highly making impact and what can be done to overcome those features. It will help to provide equality for genders income with respect to all working sectors.

APPLICATIONS

- To compare the wages gap between the occupation with respect to education of male and female.
- To compare the wages gap between the age with respect to work status of male and female.
- Year wise changes of income between male and female

CONCLUSION

In this project, A gender pay gap is been observed for many years. Taking this considerate we have seen that there is gender pay gap in data of United States from 1974 to 2018. The data consist of 11 features on which will carried out our analysis with particular model of machine learning. The features on which we are analyzing the gap between men and women. Around 61k respondent’s data was recorded in the dataset. The features were causing the difference that was occupation, work status, education, marital status, age, profession. This is due to the gender difference in education, as well as the substantial reduction in the Income. There is a gender pay gap with respect to education and occupation in Income.

In our project we have analyzed the pattern of the data. We have found that there are some features that are affecting on income. that features was occupation, gender and education. After training the dataset on different models we have found that random forest was giving best performance than other models which was 64%. the random forest is also known for there simplicity and randomly selection of features. That was beneficial for our gender pay data.The testing result was also successful. So, here we conclude that there is gender pay gap in the wages of respondent.

BIBLIOGRAPHY

[1] Fortin, Nicole M. 2015. “Gender Role Attitudes and Women's Labor Market Participation: Opting Out, AIDS, and the Persistent Appeal of Housewifery,” *Annals of Economics and Statistics*, Special Issue on Economics of Gender (117/118): 379-401.

[2] Hirsch, Boris, Thorsten Schank, and Claus Schnabel. 2010. “Differences in Labor Supply to Monopsonistic Firms and the Gender Pay Gap: An Empirical Analysis Using Linked Employer-Employee Data from Germany.” *Journal of Labor Economics* 28 (2): 291-330.

[3] Kahn, Lawrence M. 2007. “The Impact of Employment Protection Mandates on Demographic Temporary Employment Patterns: International Microeconomic Evidence.” *Economic Journal,* 117 (521): F333-F356.

[4] Kassenboehmer, Sonja C. and Mathias G. Sinning. 2014. “Distributional Changes in the Gender Wage Gap.” *ILRReview* 67 (2): 335-361.