High level Document

Adult Census Income Prediction

Revision number : 1.0
Last Date of revision : 07/05/2022

Contents

Abstract

From an adult income census data I want to derive some specific observations which will help us in predicting income of a given individual belonging to that same population upon which the sample census data was made. In this project , I have filtered out all those features which are important in determining the income status of an individual. Based on those features a machine learning classifier model is being trained which is further hypertuned to produce absolute results. I have fitted the data in many different classifiers and collected the model score. Adaboost classifiers are found to perform the best among all other classifiers.

Introduction

a. Why this high level document?

The purpose of this high level document is to add necessary details to the current project description to represent a suitable model for coding. This document is also intended to help detect contradictions prior to coding, and can be used as a reference manual for how the model interacts at a high level.

THE HLD WILL:

- Present all of the design aspects and define them in details
- Describe the user interface being implemented
- Describe the hardware and software interfaces
- Describe the performance requirements.
- Include design features and the architecture of project

b.Scope

The HLD documentation presents the structure of the system, such as database, application architecture, application flow and technology architecture.

c. Definition

Term	Description
>50K	Salary above $50K/yr
<=50K	Salary below $50K/yr

General Description

a . Problem Statement

The Goal is to predict whether a person has an income of more than 50K a year or not. This is basically a binary classification problem where a person is classified into the '>50K' group or '<=50K' group.

b. Proposed Solution

The solution here is a meta-estimator that begins by fitting a classifier on the original dataset and then fits additional copies of the classifier on the same dataset but where the weights of incorrectly classified instances are adjusted such that subsequent classifiers focus more on difficult cases. The classifier we are talking about is the *Adaboost classifier.* Using adaboost classifiers we can efficiently classify the data points into two respective classes.

c. Furthermore Improvements

A web based interface can be developed which will predict the income status of an individual once his/her details are entered. Proper pipelines can be implemented based upon the model and hypertuned parameter.

d. Technical Requirements

Implementation of this model will not need any high power computational processor or storage.

e. Data requirements

The data used in this project is taken from https://www.kaggle.com/datasets/overload10/adult-census-dataset .

f. Tool used

Tool used in this project are :

1. Python
2. Matplotlib
3. Seaborn
4. Scipy
5. Numpy

6. Pandas
7. Sckit learn
8. Flask
9. Heroku
10. Git

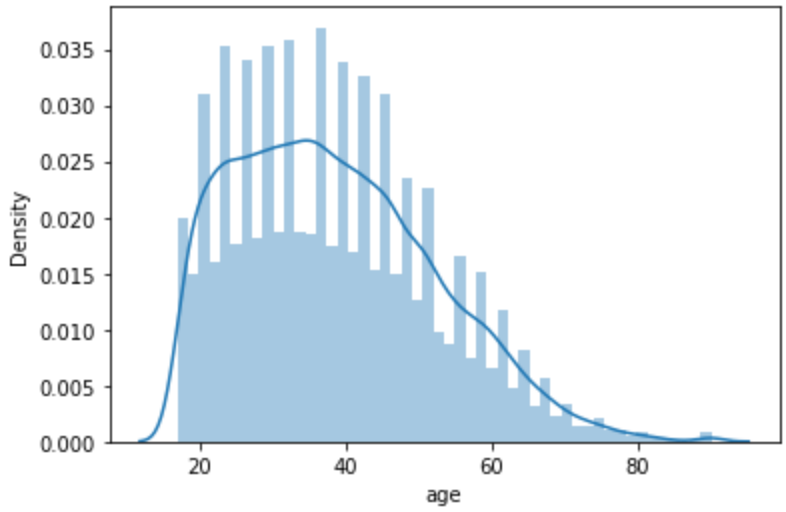
Design Details

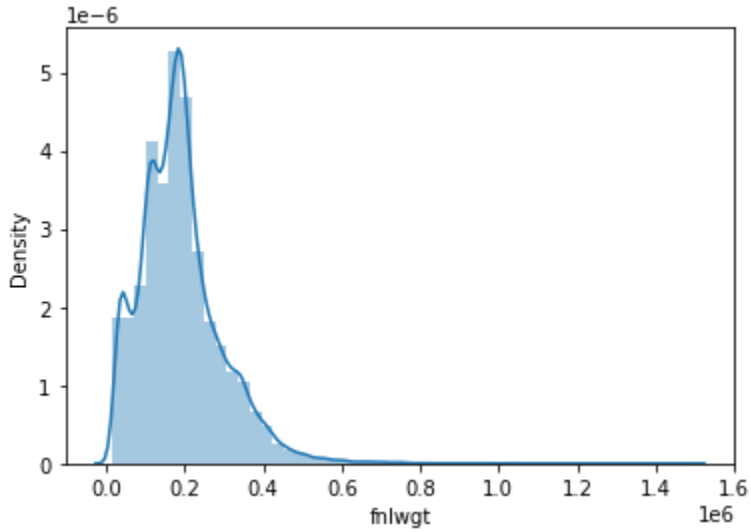
a.Process flow

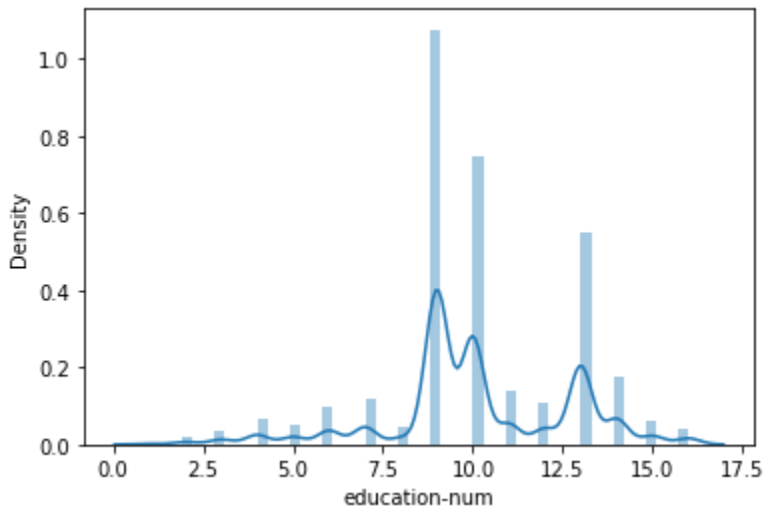
i. **Exploratory data Analysis** : There are a total 14 features in the dataset .

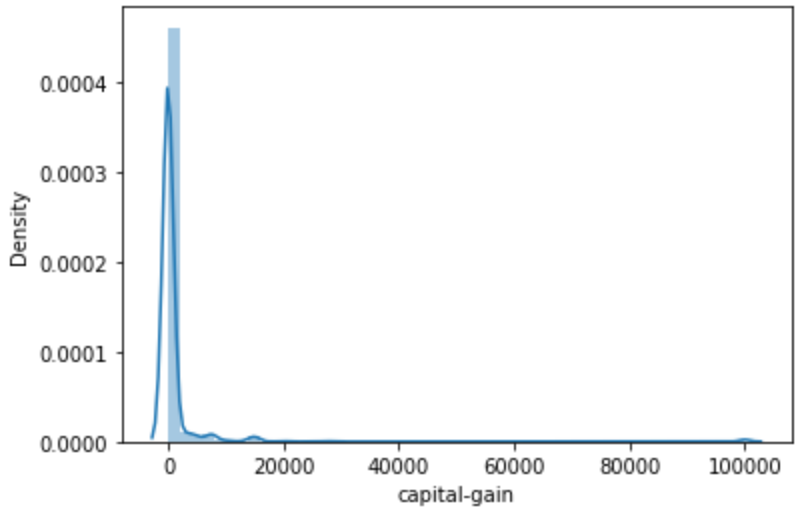
age	continuous
workclass	Private, Self-emp-not-inc, Self-emp-inc, Federal-gov, Local-gov, State-gov, Without-pay, Never-worked.
fnlwgt	continuous
education	Bachelors, Some-college, 11th, HS-grad, Prof-school, Assoc-acdm, Assoc-voc, 9th, 7th-8th, 12th, Masters, 1st-4th, 10th, Doctorate, 5th-6th, Preschool
education-num	continuous
marital-status	Married-civ-spouse, Divorced, Never-married, Separated, Widowed, Married-spouse-absent, Married-AF-spouse.
occupation	Tech-support, Craft-repair, Other-service, Sales, Exec-managerial, Prof-specialty, Handlers-cleaners, Machine-op-inspect, Adm-clerical, Farming-fishing, Transport-moving, Priv-house-serv, Protective-serv, Armed-Forces.
relationship	Wife, Own-child, Husband, Not-in-family, Other-relative, Unmarried.
race	White, Asian-Pac-Islander, Amer-Indian-Eskimo, Other, Black.

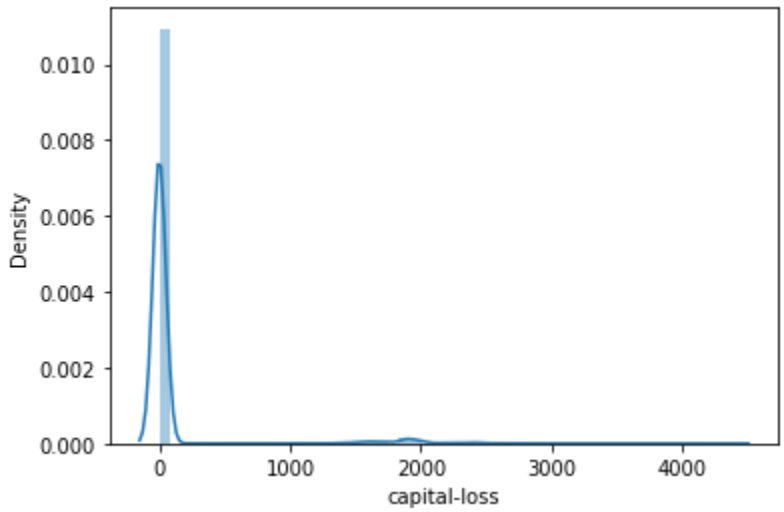
sex	Female, Male.
capital-gain	A capital gain is the increase in a capital asset's value and is realized when the asset is sold. Capital gains apply to any type of asset, including investments and those purchased for personal use. The gain may be short-term (one year or less) or long-term (more than one year) and must be claimed on income taxes.
capital-loss	A capital loss is the loss incurred when a capital asset, such as an investment or real estate, decreases in value. This loss is not realized until the asset is sold for a price that is lower than the original purchase price.
hours-per-week	continuous
country	United-States, Cambodia, England, Puerto-Rico, Canada, Germany, Outlying-US(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, Columbia, Hungary, Guatemala, Nicaragua, Scotland, Thailand, Yugoslavia, El-Salvador, Trinadad&Tobago, Peru, Hong, Holand-Netherlands.

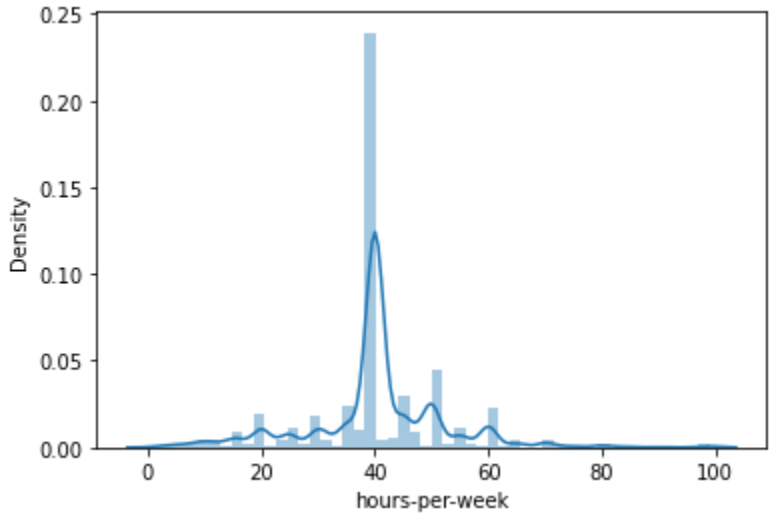

0.035
0.030
0.025
0.020
0.015
0.010
0.005
0.000
Density
20
40
60
80
age


1e−6
5
4
3
2
1
0
Density
0.0
0.2
0.4
0.6
0.8
1.0
1.2
1.4
1.6
fnlwgt
1e6


1.0
0.8
0.6
0.4
0.2
0.0
Density
0.0
2.5
5.0
7.5
10.0
12.5
15.0
17.5
education-num







ii. Feature Selection

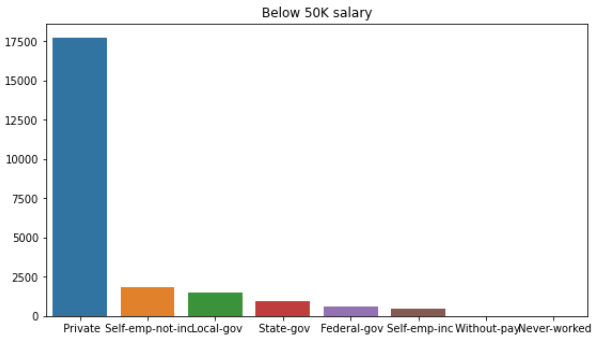
From the above result I can observe :

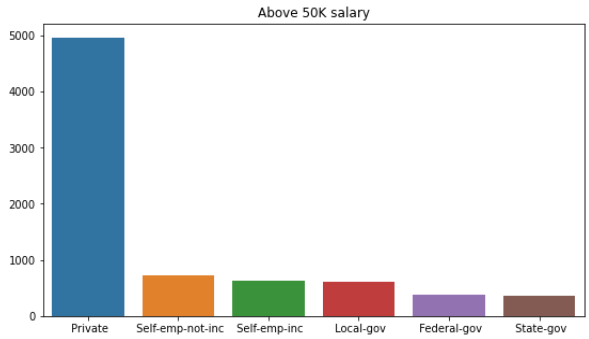
1. fnlwgt is just a census number and it is not useful for the model prediction
2. education.num is ordered-encoding of education data
3. presence of outliers in capital gain and capital loss columns

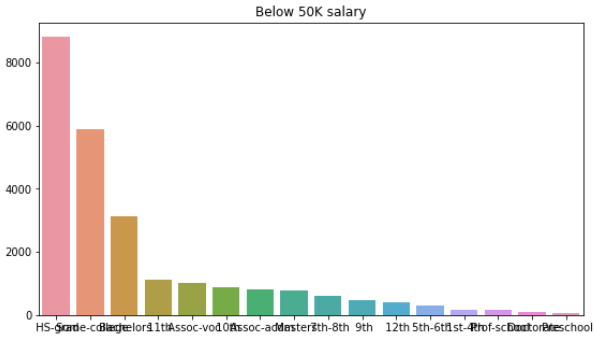
If we try to treat the outliers in the capital-gain column of the dataset then we are losing a huge amount of data. Hence I will first try to create a model keeping the outliers watching the failure of the model. I will develop another outlier treatment process.

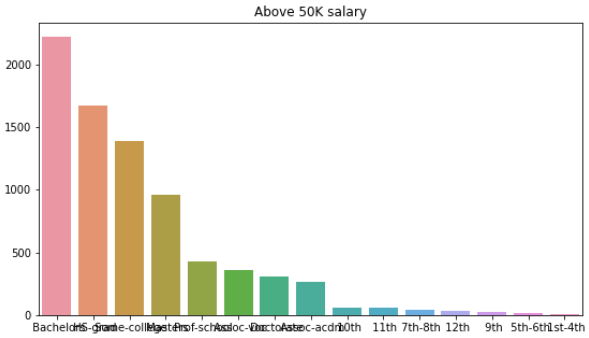
The continuous variable which I am including in the final data are

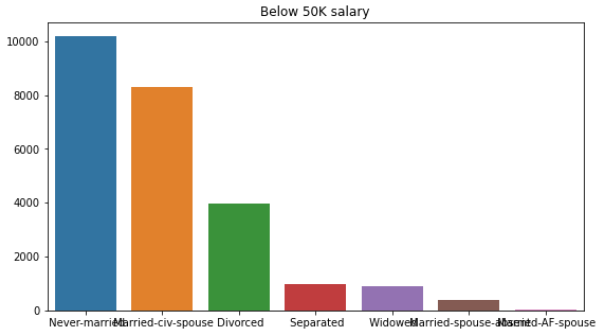
1. Age
2. education-num
3. capital-gain
4. capital-loss
5. hours-per-week

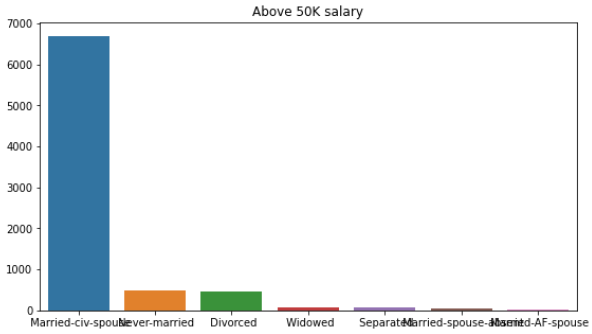


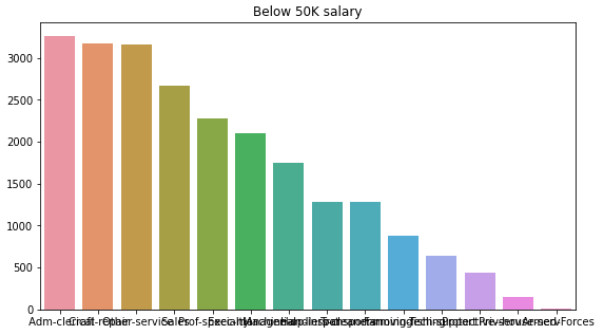


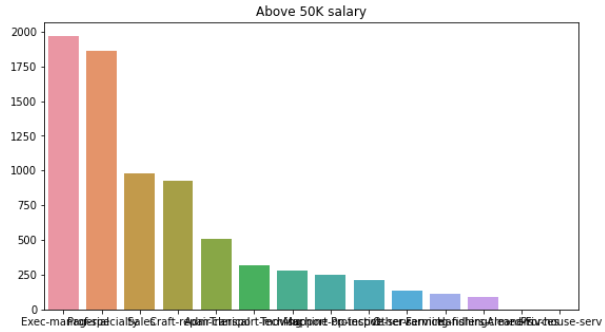


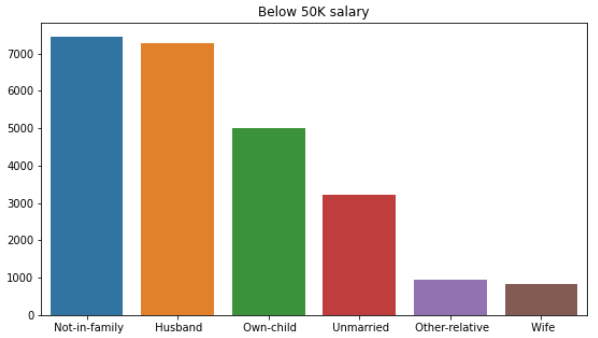


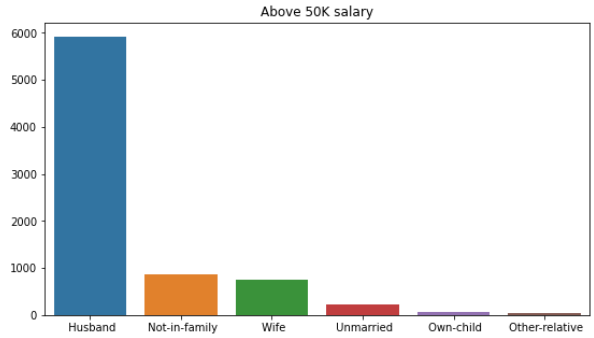

Below 50K salary
Never-married
Married-civ-spouse
Divorced
Separated
Widowed
Married-spouse-absent
Married-AF-spouse

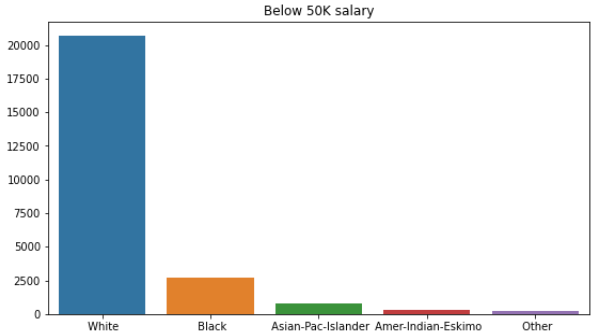

Above 50K salary
Married-civ-spouse
Never-married
Divorced
Widowed
Separated
Married-spouse-absent
Married-AF-spouse

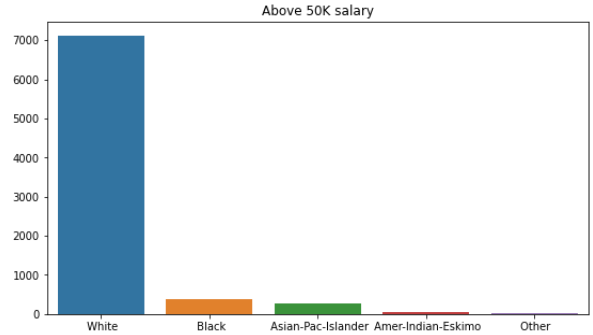

Below 50K salary

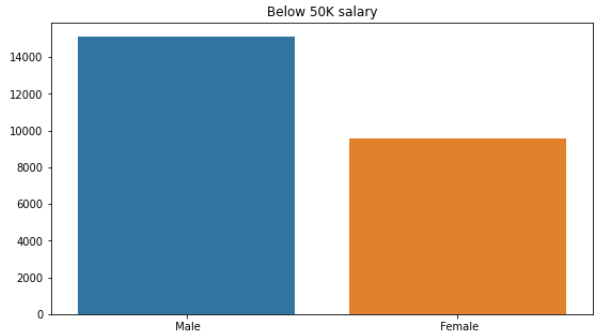

Above 50K salary

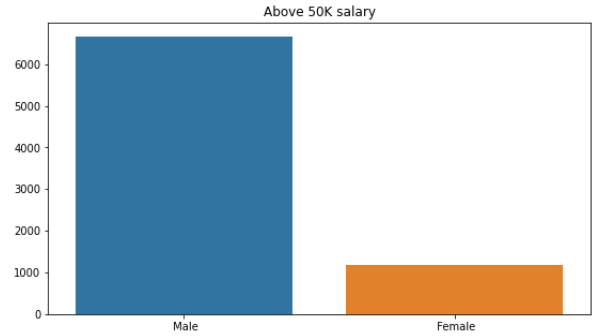

Below 50K salary
Not-in-family
Husband
Own-child
Unmarried
Other-relative
Wife


Above 50K salary
Husband
Not-in-family
Wife
Unmarried
Own-child
Other-relative


Below 50K salary
White
Black
Asian-Pac-Islander
Amer-Indian-Eskimo
Other


Above 50K salary
White
Black
Asian-Pac-Islander
Amer-Indian-Eskimo
Other





From the visualization of the categorical variables I conclude ▸

1. Drop - workclass , marital-status, relationship , country
2. Drop education as education-num (continuous variable) is already present

iii. Model Selection

The models we have analyzed in this project are :

1. Logistic Regression -It is based on log regularized logistic regression. It minimizes log probability.
2. SGD classifier- This is a linear classifier that minimizes the cost function using stochastic gradient descent.
3. Logistic Regression CV -logistic regression with builtin cross validation
4. Naive Bayes
    1. Gaussian Nave Bayes --implemet when likelihood of features are gaussian
    2. Multinomial Naive Bayes --multinomially distributed data
    3. Complement Naive Bayes --works on imbalenced multinomially distributed data set
    4. Bernoulli Naive Bayes --works for multivariate bernoulli distributed data
    5. Categorical Naive Bayes -- for categorically distributed data
    6. Out of core naive bayes model fitting -- for large classification problem

note :-> Bernoulli:Binomial::Categorical:Multinomial

1. Nearest Neighbor -- do scaling before fitting the data , good for small dataset.
2. Support vector classifier
3. Gaussian classifier
4. Decision Tree
5. Random forest
6. Ada boost
7. Neural Net

8. QDA

Score of respective models

Model	Accuracy(%)
Logistic Regression	80
SGD Classifier	79
Logistic Regression CV	80
Multinomial NB	76
KNeighbors Classifier	79
SVC	81
Decision Tree Classifier	78
Random Forest Classifier	79
AdaBoost Classifier	81
MLP Classifier	81
Quadratic Discriminant Analysis	80

Hence we choose the Adaboost classifier to be best fit for this dataset.

iv. Hyper parameter tuning

The hyper parameter that will be tuned are :-

```
base_estimator
n_estimator
```

```
learning rate
algorithm
```

On performing Randomized Search CV followed by Grid Search CV that best parameters that we get for the model are :

```
{'algorithm': 'SAMME.R', 'learning_rate': 1, 'n_estimators': 99}
```

The final classification report of the project is :

	precision	recall	f1-score	support
0	0.84	0.93	0.88	8128
1	0.68	0.44	0.54	2618
accuracy			0.81	10746
macro avg	0.76	0.69	0.71	10746
weighted avg	0.80	0.81	0.80	10746

Deployment

For deploying the project I have used the flask framework in which a simple webpage is designed which predicts the income status of an individual based upon six parameters. These are :

1.Age 2.Education 3.Hours per week 4.Occupation 5.Race 6.Gender

The form is allowed to be filled with integers and dropdown menu options. On choosing an option from the dropdown menu the website assigns a numerical value to it and thus this value is converted to an array for future prediction.

After successful performance on the localhost computer , I have deployed it to the heroku web server. These are the following commands which need to be implemented for successful deployment of a flask app.

>> Login to heroku
>> create an app “ income-predictor-flask” choose the usa server
>> install heroku CLI
>> Open your project
>> Create a Procfile : this file helps to run the flask app in heroku server

```
web: gunicorn app:app
```

>> activate the virtual environment.

```
git init
heroku login
heroku git:remote -a income-predictor-flask
pip install gunicorn
pip freeze > requirements.txt
git add .
git commit -m “Initial commit”
git push heroku master
heroku open
```

The project is deployed at : https://income-predictor-flask.herokuapp.com/

Conclusion

The hypertuned Adaboost classifier model is very accurate in classifying the population data into two groups based upon the salary (<50K and >=50K respectively) . For the classification the features being used are - 'age', 'education-num', 'hours-per-week', 'occupation' , 'race', 'sex'.