A Statistical Approach to Adult Census Income

-Mohit Singh

Project in computational Science: Report

MAY 2023

PROJECT REPORT

Lovely Professional university

A Statistical Approach to Adult Census Income

MASTER OF TECHNOLOGY

in

COMPUTER SCIENCE AND ENGINEERING

By

MOHIT SINGH

12013468

Supervisor

Ved Prakash Chaubey

School of Computer Science and Engineering

Lovely Professional University

Phagwara, Punjab (India)

Month - MAY Year 2023

DECLARATION STATEMENT

I hereby declare that the research work reported in the dissertation/dissertation proposal entitled " **A Statistical Approach to Adult Census Income** ” in partial fulfilment of the requirement for the award of Degree for Bachelor of Technology in Computer Science and Engineering at Lovely Professional University, Phagwara, Punjab is an authentic work carried out under supervision of my research supervisor Mr./Mrs. **Ved Prakash Chaubey**. I have not submitted this work elsewhere for any degree or diploma.

I understand that the work presented herewith is in direct compliance with Lovely Professional University’s Policy on plagiarism, intellectual property rights, and highest standards of moral and ethical conduct. Therefore, to the best of my knowledge, the content of this dissertation represents authentic and honest research effort conducted, in its entirety, by me. I am fully responsible for the contents of my dissertation work.

Signature of Candidate

Mohit Singh

12013468

SUPERVISOR'S CERTIFICATE

This is to certify that the work reported in the B.Tech Dissertation/dissertation proposal entitled "**A Statistical Approach to Adult Census Income"**, submitted by **Mohit Singh** at **Lovely Professional University, Phagwara, India** is a Bonafede record of his original work carried out under my supervision. This work has not been submitted elsewhere for any other degree.

Signature of Supervisor

(Name of Supervisor)

Date:

Counter Signed by:

1) **Concerned HOD:**
   HoD's Signature: ________________

   HoD Name: ____________________

   Date: ___________________

2) **Neutral Examiners:**

   External Examiner

   Signature: _______________

   Name: __________________

   Affiliation: ______________

   Date: ___________________

   Internal Examiner

   Signature: _______________

   Name: __________________

   Date: ___________________

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 optimalsolution. 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.

Table of Contents

1 Introduction

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. The problem of income inequality has been of great concern in the recent years. Making the poor better off does not seem to be the sole criteria to be in quest for eradicating this issue. People of the United States believe that the advent of economic inequality is unacceptable and demands a fair share of wealth in the society. This model actually aims to conduct a comprehensive analysis to highlight the key factors that are necessary in improving an individual's income. Such an analysis helps to set focus on the important areas which can significantly improve the income levels of individuals. This paper has been structured as an introduction, literature review, proposed methodology, training the model, implementation details, results and conclusion

2 Theoretical Background

Supervised and Unsupervised Learning

Supervised learning: "Supervised learning is a type of machine learning algorithm that uses a known dataset (called the training dataset) to make predictions. The training dataset includes input data and response values.", [4].

Unsupervised learning: "Unsupervised learning is a type of machine learning algorithm used to draw inferences from datasets consisting of input data without labeled responses.",[5].

In data mining and machine learning an abundance of models and algorithms can be found, but most fundamentally these are divided into supervised and unsupervised learning. One fundamental example has been mentioned in the foregoing section, the clustering of *iris*-species. Former is a supervised process where data points are labeled ("species A", "species B" or "species C") and labels are calculated for new data points. Comparing calculated labels according to the trained model with the original label gives the model's accuracy, hence supervised.

Unsupervised learning on the other hand does not require any labeling, since the algorithm is searching for a pattern in the data. This might be useful when categorizing customers into different groups without a priori knowledge of which groups they belong to.

Machine Learning Algorithm Types

For machine learning many different algorithms can be found. A wide variety of these are available in Azure Machine Learning Studio. For simplicity these can be subdivided into four categories, where each category is good for different kind of problems. **Anomaly detection** algorithms, are good for finding unusual data points. Trained **classification** algorithms can be used to categorize unseen data. As an example, it could be used to take in data from a phone on movement to categorize what activity is being performed. **Clustering** algorithms group data into clusters and look for the greatest similarities. This can be used to find unknown connections on huge sets of data. **Regression** algorithms are used to find patterns and build models to predict numerical values from datasets. These will take multiple inputs and determine how much each input affects the output

Within the regression category there are eight different basic algorithms, each suitedfor different kind of problems, available in Azure Machine Learning Studio. **Boosted decision tree regression**, which is based on decision trees, where each tree depends on prior trees, uses decision splitting to create stepped functions. It learns by fitting the residuals of preceding trees to improve accuracy. **Decision forest regression** consistsof decision trees in regression and is resilient against noise, due to the fact that manytrees form a "forest". This makes it easily to parallelize. **Fast forest quantile regression** is effective in predicting weak relationships between variables. Unlike linear regression quantile algorithms try to find patterns in the distribution of the predicted values rather than just predict values. **Linear Regression** is the most classic type which solves linear relationships between inputs and outputs. **Neural network regression** is most common indeep learning and adaptable to regression problems, but might be too complex for simple regression problems and requires thorough training. This method is very stable and is often used when other algorithms can not find a solution. **Ordinal regression** is found useful for predicting discrete ranking. **Poisson regression** is useful to predict values if the response variable follow a Poisson distribution.

In general it should be noted, that the algorithms from Microsoft are not open source and it is therefore impossible to completely comprehend the underlying mechanisms.

Bias-Variance-Trade-off

A problem in machine learning is to find a balanced compromise between training accuracy and validation accuracy. This means to find a function that solves a training problem accurately, e.g. a high-degree polynomial that fits the training data well, butis not overfitting the validation data. On the other hand, a function too simple can oversimplify (underfit) a problem and neglect present patterns. This is a problem evident only in supervised learning and therefore relevant for regression analysis. The dilemma often leads to trial-and-error strategies of finding suitable models [3].

3 Problem Statement

We have to predict the income of a person whether he/she is earning more than 50k or less then 50k on the basis of certain factors like age, work class, education, marital status, occupation, relationship, race, sex, hours per week, native country etc using different machine learning algorithm.

4 Methodology

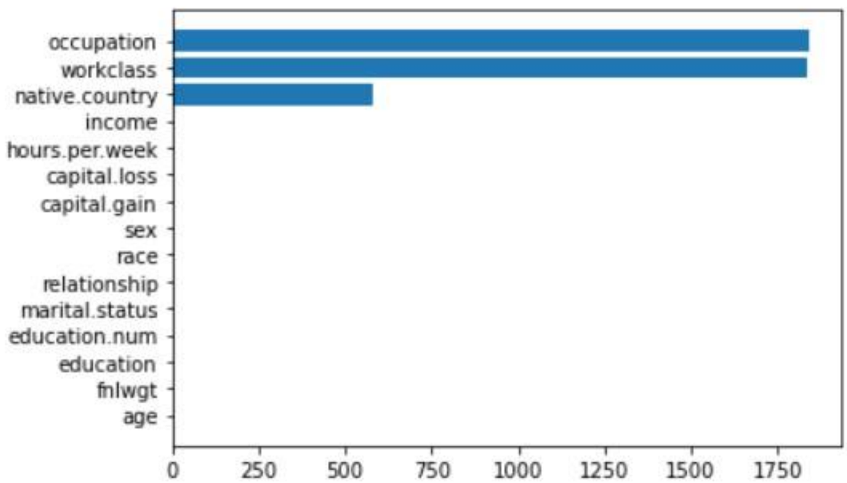
4.1 Data Review

The data for our study was accessed from the University of California Irvine (UCI) Machine Learning Repository [8]. 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 as shown in Table 1. 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

4.2 Data Visualization

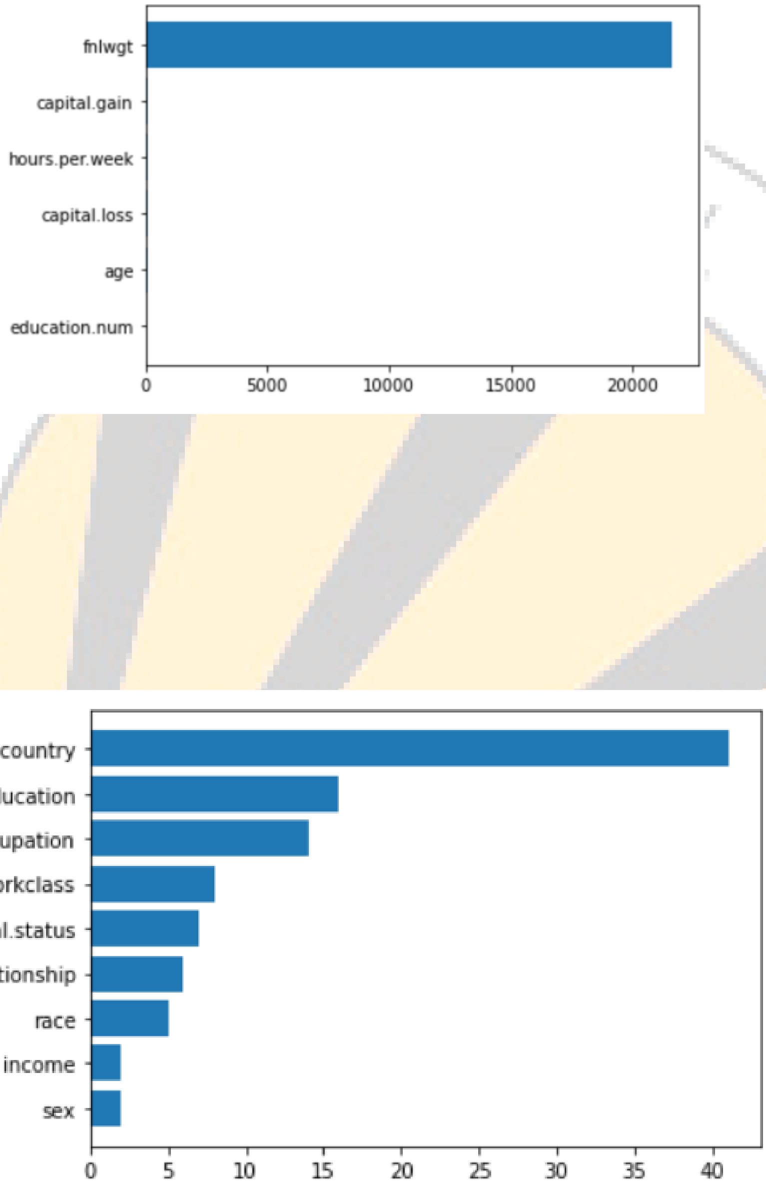
The exploratory data analysis is done to get the insight about the data and get to know about the feature dependency.

- Bar plot which shows the distribution of null values of data set. It can be inferred from the plot that only occupation workplace and native country have null values



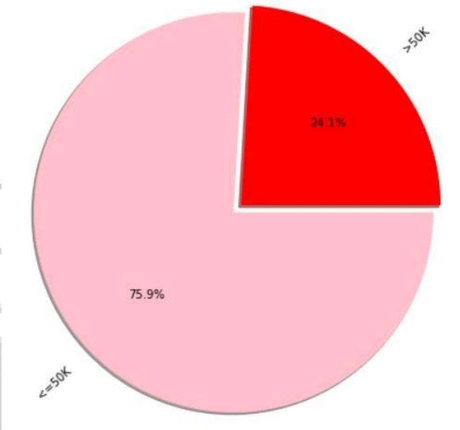
FIG(1)

- Bar plot which shows the distribution of unique value of data set.



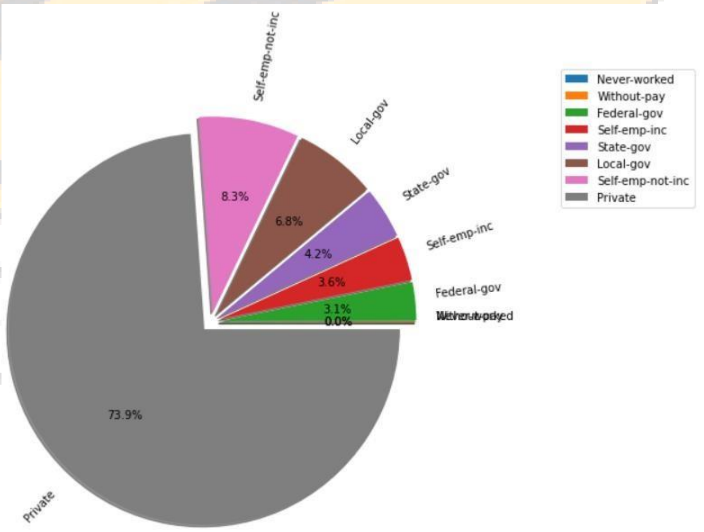
FIG(2)

- Pie chart which shows the distribution of income. It can be inferred from the chart that more than 75% income is less than 50k



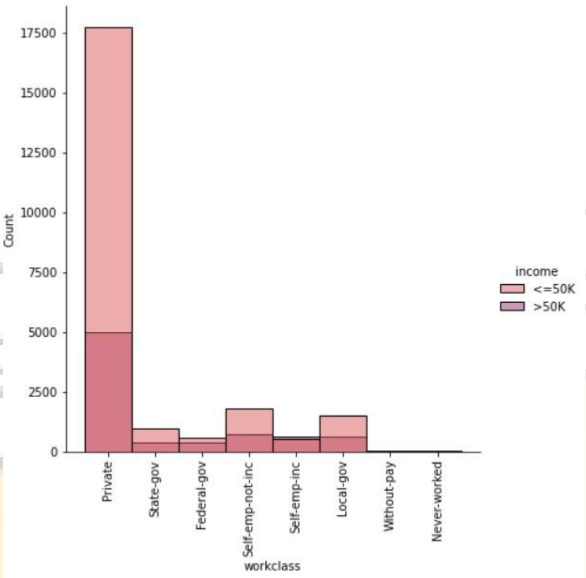
FIG(3)

- Pie chart which shows the distribution of work class . It can be inferred from the chart that more than 73% income is private class



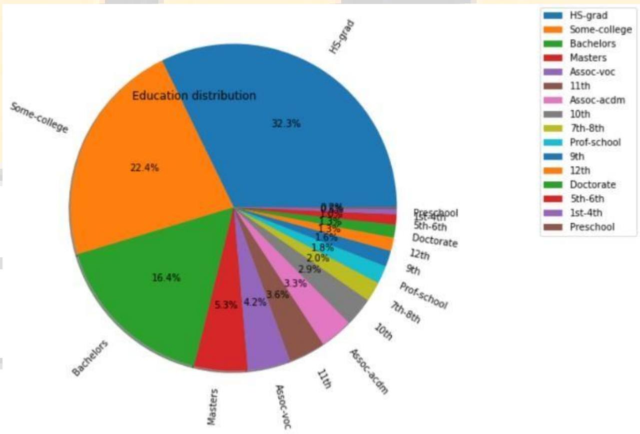
FIG(4)

- Histogram chart which shows the distribution of work class with respect to income. It can be inferred from the chart that private class has more number of income source and more number of person who has earning more than 50k and less than 50k



FIG(5)

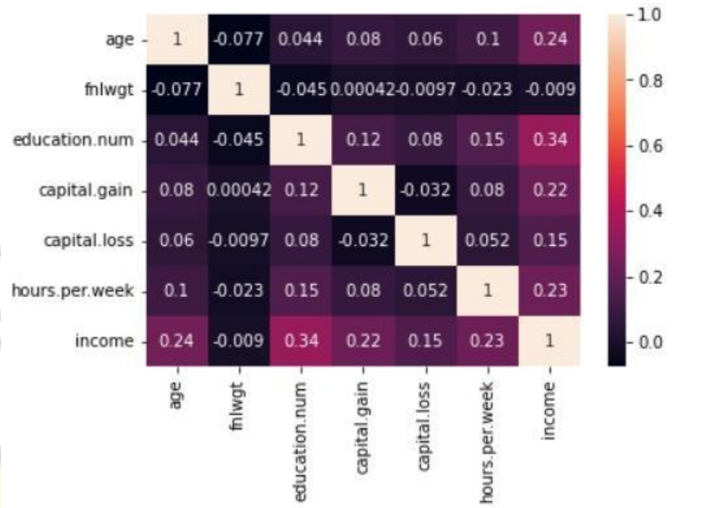
- Pie chart which shows the distribution of Education . It can be inferred from the chart that more than 32% are educated in High Secondary Graduation



FIG(6)

4.3 Feature Study and Selection

A Correlation Matrix is shown in the form of a HeatMap showing Feature-to-Feature and Feature-to-Label Pearson Correlations where all the features are Continuous Variables.



FIG(7)

Heat-Map showing Feature-to-Feature and Feature-to-Label's Pearson Correlation Coefficients

.

4.4 SPLITTING DATASET AND MODELLING:-

The shape of the dataset after the deletion of the duplicates is (32537, 15). The dataset is split where 70% is the used for training the model and 30% for testing the model. Hence out of 32,537 data entries, 21,485 are used for training and 9,209 are used for testing the model. 1 Classification Algorithms are used.

4.4.1 Logistic Regression

Logistic regression is a statistical model that in its basic form uses a logistic functionto model a binary dependent variable. In regression analysis, logistic regression is estimating the parameters of a logistic model (a form of binary regression). Logistic regression is the appropriate regression analysis to conduct when the dependent variable is dichotomous (binary). Like all regression analyses, the logistic regression is a predictive analysis. Logistic regression is used to describe data

4.4.2 RandomForest Classifier

Random Forest is a popular machine learning algorithm that belongs to the supervised learning technique. It can be used for both Classification and Regression problems in ML. It is based on the concept of ensemble learning, which is a process of combining multiple classifiers to solve a complex problem and to improve the performance of the model

As the name suggests, "Random Forest is a classifier that contains a number of decision trees on various subsets of the given dataset and takes the average to improve the predictive accuracy of that dataset." Instead of relying on one decision tree, the random forest takes the prediction from each tree and based on the majority votes of predictions, and it predicts the final output.

4.4.3 BernoulliNB

Bernoulli Naïve Bayes is another useful naïve Bayes model. The assumption in this model is that the features binary (0s and 1s) in nature. An application of Bernoulli Naïve Bayes classification is Text classification with ‘bag of words’ model. The Scikit-learn provides sklearn.naive_bayes.BernoulliNB to implement the Gaussian Naïve Bayes algorithm for classification.

4.4.4 Support Vector Classifier

Support Vector Machine or SVM is one of the most popular Supervised Learning algorithms, which is used for Classification as well as Regression problems. However, primarily, it is used for Classification problems in Machine Learning.

The goal of the SVM algorithm is to create the best line or decision boundary that can segregate n-dimensional space into classes so that we can easily put the new data point in the correct category in the future. This best decision boundary is called a hyperplane.

5 CONCLUSION AND FUTURE SCOPE

This paper proposed the application of Ensemble Learning Algorithm, on Adult Census Data.
Finally, the Validation Accuracy, so obtained, **84.4%** which is, by the best of our knowledge, has been the highest ever numeric accuracy achieved by any Income Prediction Model so far. The future scope of this work involves achieving an over-all better set of results by using hybrid models with inclusion of Machine Learning and Deep Learning together, or by applying many other advanced preprocessing techniques without further depletion in the accuracy.

6 CODE

In [36]:
```python
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import warnings
warnings.filterwarnings('ignore')
```

In [37]:
```python
df = pd.read_csv("adult.csv" )
```

In [38]:
```python
df.head()
```

Out[38]:

	age	workclass	fnlwgt	education	education.num	marital.status	occupation	relationship	race	sex	capital.gain	capital.loss	hours.per.week	native.cc
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-

In [4]:
```python
df.shape
```

Out[4]:
```
(32561, 15)
```

In [5]:
```python
df.isnull().sum()
```

Out[5]:
```
age               0
workclass         0
fnlwgt            0
education         0
education.num     0
marital.status    0
occupation        0
relationship      0
race              0
sex               0
capital.gain      0
capital.loss      0
hours.per.week    0
native.country    0
income            0
dtype: int64
```

In [7]:
```python
df.dtypes
```

Out[7]:
```
age                int64
workclass         object
fnlwgt             int64
education         object
education.num      int64
marital.status    object
occupation        object
relationship      object
race              object
sex               object
capital.gain       int64
capital.loss       int64
hours.per.week     int64
native.country    object
income            object
dtype: object
```

replacing "?" value to null values

In [8]:
```python
df[df=='?']=np.nan
```

In [9]:
```
df.isnull().sum()
```

Out[9]:
```
age                  0
workclass         1836
fnlwgt               0
education            0
education.num        0
marital.status       0
occupation        1843
relationship         0
race                 0
sex                  0
capital.gain         0
capital.loss         0
hours.per.week       0
native.country     583
income               0
dtype: int64
```

In [11]:
```
df.nunique()
```

Out[11]:
```
age                  73
workclass             8
fnlwgt            21648
education            16
education.num        16
marital.status        7
occupation           14
relationship          6
race                  5
sex                   2
capital.gain        119
capital.loss         92
hours.per.week       94
native.country       41
income                2
dtype: int64
```

In [12]:
```
df.describe()
```

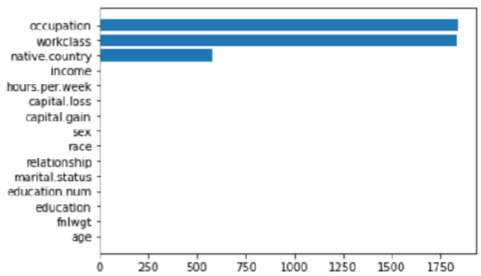
Out[12]:

	age	fnlwgt	education.num	capital.gain	capital.loss	hours.per.week
count	32561.000000	3.256100e+04	32561.000000	32561.000000	32561.000000	32561.000000
mean	38.581647	1.897784e+05	10.080679	1077.648844	87.303830	40.437456
std	13.640433	1.055500e+05	2.572720	7385.292085	402.960219	12.347429
min	17.000000	1.228500e+04	1.000000	0.000000	0.000000	1.000000
25%	28.000000	1.178270e+05	9.000000	0.000000	0.000000	40.000000
50%	37.000000	1.783560e+05	10.000000	0.000000	0.000000	40.000000
75%	48.000000	2.370510e+05	12.000000	0.000000	0.000000	45.000000
max	90.000000	1.484705e+06	16.000000	99999.000000	4356.000000	99.000000

Visulazation

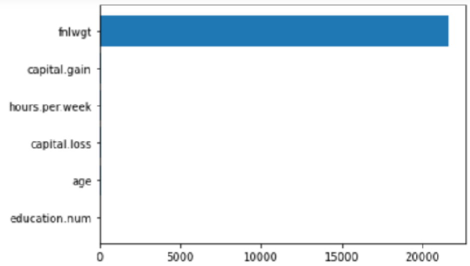
In [13]:
```
temp = df.isnull().sum().sort_values()
plt.barh(temp.index,temp  )
plt.figure(figsize=(20,20))
```

Out[13]: <Figure size 1440x1440 with 0 Axes>

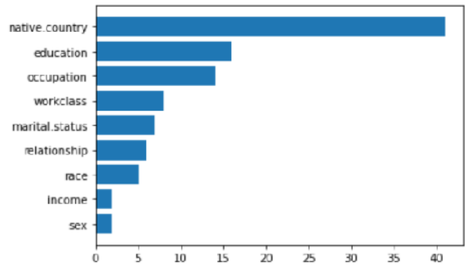


<Figure size 1440x1440 with 0 Axes>

In [14]:
```
temp1 = df.select_dtypes(include='number').nunique().sort_values()
plt.barh(temp1.index ,temp1  )
plt.figure(figsize=(10,10))
plt.show()
temp2 = df.select_dtypes(exclude='number').nunique().sort_values()
plt.barh(temp2.index ,temp2   )
plt.figure(figsize=(20,20))
plt.show()
```

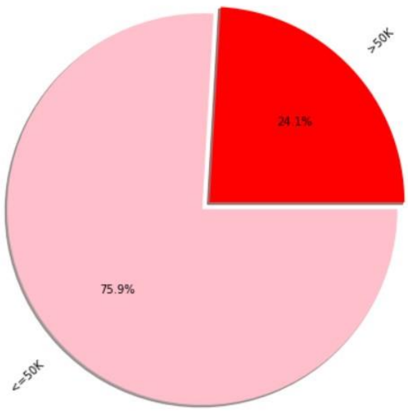


<Figure size 720x720 with 0 Axes>



<Figure size 1440x1440 with 0 Axes>

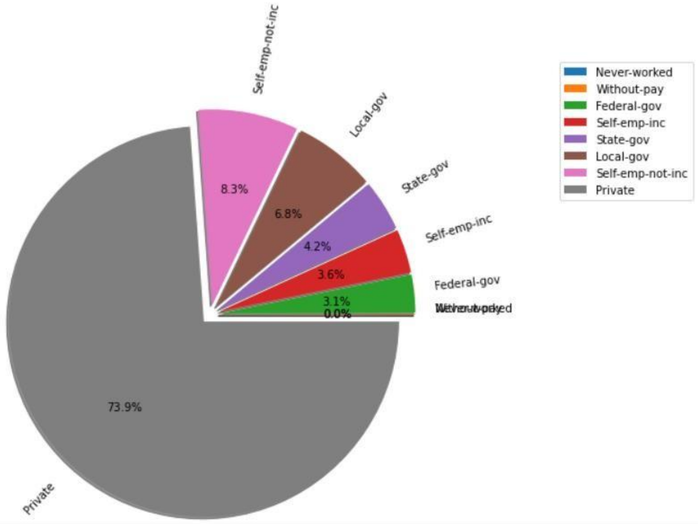
In [15]:
```
temp = df['income'].value_counts().sort_values()
explode =[0,0.1]
plt.pie(temp,labels=temp.index, radius=2, colors=['red','pink'], shadow=True,rotatelabels=True , autopct='%1.1f%%',explode=exploc
plt.show()
```



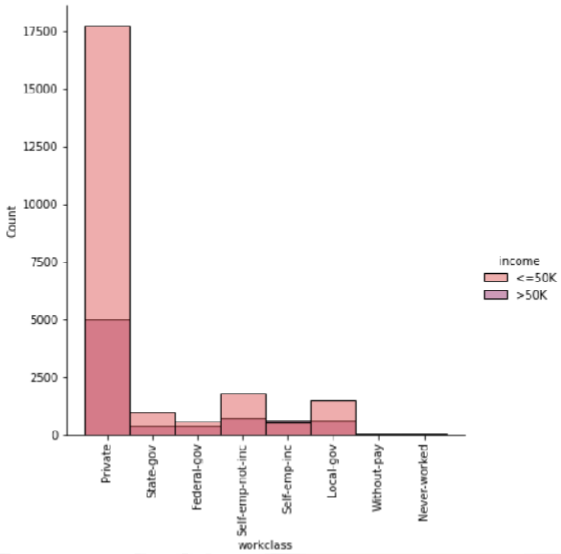
```
In [16]: temp = df['workclass'].value_counts().sort_values()


         explode =[0.1,0.1,0.1,0.1,0.1,0.1,0.1,0.1]
         plt.pie(temp, labels = temp.index , radius=2, shadow=True,rotatelabels=True , autopct='%1.1f%%', explode=explode  )
         plt.legend(loc ="right" , bbox_to_anchor=(2, 1, 0.5, 0.5))

         plt.show()
```



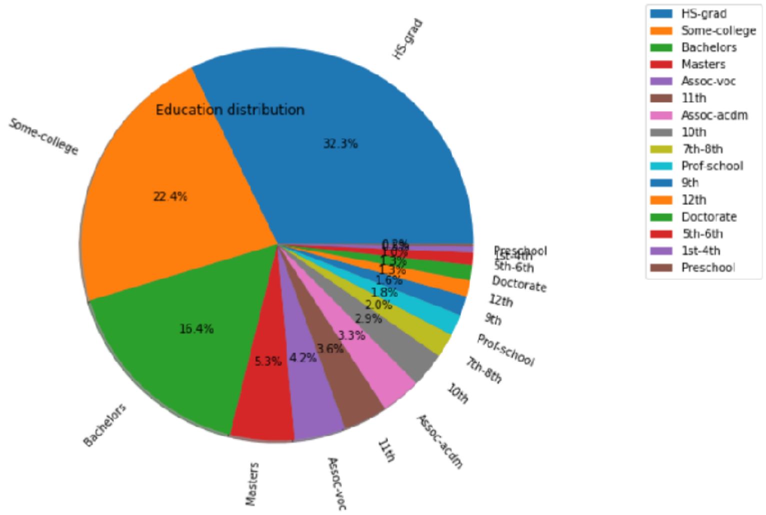

```
In [17]: sns.displot(df,x='workclass' , hue='income' , palette = "flare"  , height= 6 )
         plt.xticks(rotation = 90)
         plt.show()
```



```
In [18]: temp = df['education'].value_counts()


         plt.pie(temp, radius=2, labels=temp.index , shadow=True,rotatelabels=True , autopct='%1.1f%%' )
         plt.legend(loc='upper right' , bbox_to_anchor=(2, 1, 0.5, 0.5))

         plt.title("Education distribution" ,loc='left' )
         plt.show()
```



Dealing with null values

In [19]:
```
df.isnull().sum()
```

Out[19]:
```
age                  0
workclass         1836
fnlwgt               0
education            0
education.num        0
marital.status       0
occupation        1843
relationship         0
race                 0
sex                  0
capital.gain         0
capital.loss         0
hours.per.week       0
native.country     583
income               0
dtype: int64
```

In [20]:
```
df=df.dropna()
```

In [21]:
```
df.isnull().sum()
```

Out[21]:
```
age               0
workclass         0
fnlwgt            0
education         0
education.num     0
marital.status    0
occupation        0
relationship      0
race              0
sex               0
capital.gain      0
capital.loss      0
hours.per.week    0
native.country    0
income            0
dtype: int64
```

Scaling income column

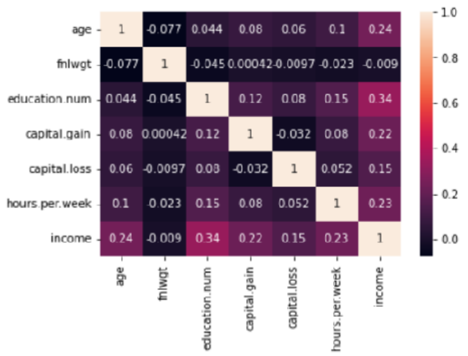
In [22]:
```
df['income'] = df['income'].replace('<=50K' , 0)
df['income'] = df['income'].replace('>50K' , 1)
```

Checking Multicollinearity

In [24]:
```
corr =df.corr()
```

In [25]:
```
sns.heatmap(corr , annot=True)
```

Out[25]: <AxesSubplot:>



Droping column 'fnlwgt' as correlation is very low

In [26]:
```
df = df.drop('fnlwgt', axis=1)
```

Labeling or Scaling the data

In [27]:
```
from sklearn.preprocessing import LabelEncoder
```

In [28]:
```
le = LabelEncoder()
```

In [29]:
```
df['workclass'] = le.fit_transform(df['workclass'])
df['education'] = le.fit_transform(df['education'])
df['marital.status'] = le.fit_transform(df['marital.status'])
df['occupation'] = le.fit_transform(df['occupation'])
df['relationship'] = le.fit_transform(df['relationship'])
df['race'] = le.fit_transform(df['race'])
df['sex'] = le.fit_transform(df['sex'])
df['native.country'] = le.fit_transform(df['native.country'])
```

In [ ]:

In [30]:
```
X = df.drop('income' , axis =1 )
y = df['income']
```

Spliting the data in Train and Test

In [31]:
```
from sklearn.model_selection import train_test_split
```

In [32]:
```
X_train , X_test ,  y_train , y_test = train_test_split(X, y ,test_size=0.3 ,random_state=50)
```

Logistic Reggression Approach

In [37]:
```
from sklearn.linear_model import LogisticRegression
from sklearn import metrics
classifier = LogisticRegression(random_state = 0)
classifier.fit(X_train, y_train)
y_pred = classifier.predict(X_test)
print("ACCURACY OF THE MODEL: ", metrics.accuracy_score(y_test, y_pred))
```

```
ACCURACY OF THE MODEL:  0.7989833130732678
```

RandomForest Approach

In [38]:
```
from sklearn.ensemble import RandomForestClassifier
from sklearn import metrics
rfclassifier = RandomForestClassifier(random_state = 0)
rfclassifier.fit(X_train, y_train)
y_pred = rfclassifier.predict(X_test)
print("ACCURACY OF THE MODEL: ", metrics.accuracy_score(y_test, y_pred))
```

```
ACCURACY OF THE MODEL:  0.8447342247762184
```

Bernoulli Naive Bayes Approach

In [40]:
```
from sklearn.naive_bayes import BernoulliNB
from sklearn import metrics
NBclassifier = BernoulliNB()
NBclassifier.fit(X_train, y_train)
y_pred = NBclassifier.predict(X_test)
print("ACCURACY OF THE MODEL: ", metrics.accuracy_score(y_test, y_pred))
```

```
ACCURACY OF THE MODEL:  0.7261023317493646
```

Support Vector Approach

In [*]:
```
from sklearn.svm import SVC
from sklearn import metrics
SVCclassifier = SVC()
SVCclassifier.fit(X_train, y_train)
y_pred = SVCclassifier.predict(X_test)
print("ACCURACY OF THE MODEL: ", metrics.accuracy_score(y_test, y_pred))
```

Conclusion

```
The performances of the models are compared on the basis of classification report.
The RandomForest Classification comes to
be the best performing algorithm above all other models
with an accuracy of 84.4% and over all generalizing well.
```

7 REFERENCES

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

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