Income Slab Prediction

NAME : PULLEPU SAI VENKATA SRINIVAS

CLASS : CSE-2A

ID: S180262

Subject: Data Science with Python

TABLE OF CONTENTS

Contents

Introduction

Income is a very necessary thing to a human in present conditions. That's why every human is worked for to generate income and survive.

In the present situation, a person can generate an income by working or by a business. Mostly everyone earns through by working, some people earn more income and some are earning low based on their skills.

Objective:

My model is aimed to classify a person as less than equal to a $50k income slab or more than it.

Why $50k is selected to classify?

In the US $50k income is necessary to live for a person there, that's why I want to classify the people who meet their basic needs in the US or not.

If a person has less than $50k means, he/she income doesn't meet their needs, and also difficult to live.

If a person is greater than $50k means, he/she earns enough money to survive

In which areas we can use this model:

- This model is helpful to the government because while implementing several schemes, it helps in finding the people really who require that schemes.
- NGOs, because they use this model to generate their donations by selecting donors based on their income conditions.

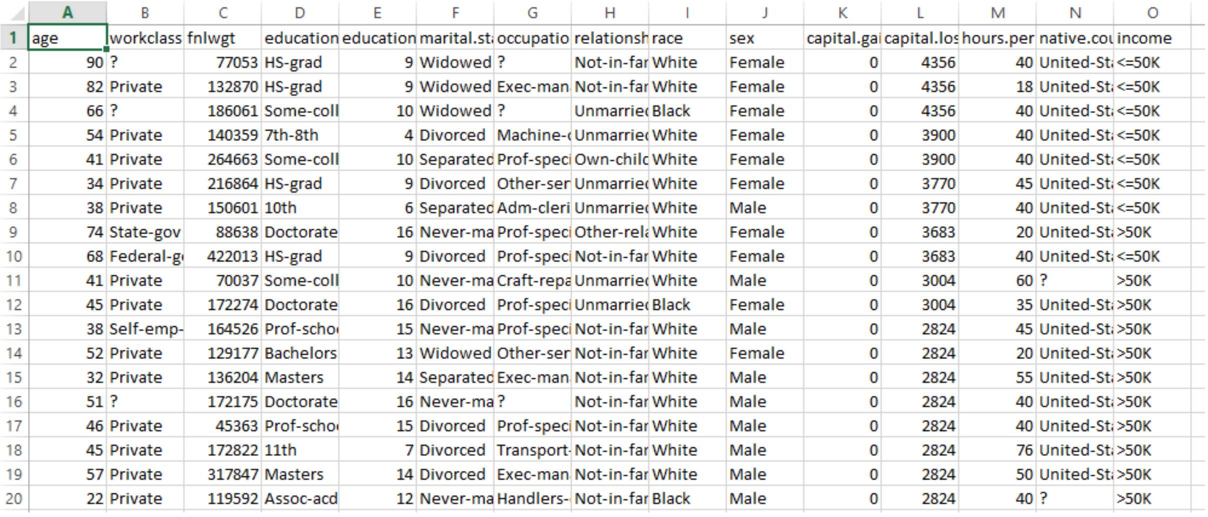
Prerequisites:

- The basic idea of income-related features. (E.g. Capital loss, capital gain)
- Must know how to import Datasets and how to work on them using Pandas in Python.
- Must know how to pre-process the data.
- Must know classifiers and their types for making predictions.
- Must know how to visualize data using different charts and plotting techniques.

Source:

Dataset Information:

The dataset used in this project "Adult Income dataset". Extraction was done by Barry Becker from the 1994 census database and I downloaded it from Kaggle.



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
1	age	workclass	fnlwgt	education	education	marital.st	occupatio	relationsh	race	sex	capital.gai	capital.los	hours.per	native.cou	income
2	90	?	77053	HS-grad	9	Widowed	?	Not-in-far	White	Female	0	4356	40	United-St	<=50K
3	82	Private	132870	HS-grad	9	Widowed	Exec-man	Not-in-far	White	Female	0	4356	18	United-St	<=50K
4	66	?	186061	Some-coll	10	Widowed	?	Unmarrie	Black	Female	0	4356	40	United-St	<=50K
5	54	Private	140359	7th-8th	4	Divorced	Machine-	Unmarrie	White	Female	0	3900	40	United-St	<=50K
6	41	Private	264663	Some-coll	10	Separated	Prof-speci	Own-chilc	White	Female	0	3900	40	United-St	<=50K
7	34	Private	216864	HS-grad	9	Divorced	Other-ser	Unmarrie	White	Female	0	3770	45	United-St	<=50K
8	38	Private	150601	10th	6	Separated	Adm-cleri	Unmarrie	White	Male	0	3770	40	United-St	<=50K
9	74	State-gov	88638	Doctorate	16	Never-ma	Prof-speci	Other-rel	White	Female	0	3683	20	United-St	>50K
10	68	Federal-g	422013	HS-grad	9	Divorced	Prof-speci	Not-in-far	White	Female	0	3683	40	United-St	<=50K
11	41	Private	70037	Some-coll	10	Never-ma	Craft-repa	Unmarrie	White	Male	0	3004	60	?	>50K
12	45	Private	172274	Doctorate	16	Divorced	Prof-speci	Unmarrie	Black	Female	0	3004	35	United-St	>50K
13	38	Self-emp-	164526	Prof-scho	15	Never-ma	Prof-speci	Not-in-far	White	Male	0	2824	45	United-St	>50K
14	52	Private	129177	Bachelors	13	Widowed	Other-ser	Not-in-far	White	Female	0	2824	20	United-St	>50K
15	32	Private	136204	Masters	14	Separated	Exec-man	Not-in-far	White	Male	0	2824	55	United-St	>50K
16	51	?	172175	Doctorate	16	Never-ma	?	Not-in-far	White	Male	0	2824	40	United-St	>50K
17	46	Private	45363	Prof-scho	15	Divorced	Prof-speci	Not-in-far	White	Male	0	2824	40	United-St	>50K
18	45	Private	172822	11th	7	Divorced	Transport-	Not-in-far	White	Male	0	2824	76	United-St	>50K
19	57	Private	317847	Masters	14	Divorced	Exec-man	Not-in-far	White	Male	0	2824	50	United-St	>50K
20	22	Private	119592	Assoc-acd	12	Never-ma	Handlers-	Not-in-far	Black	Male	0	2824	40	?	>50K

Columns:

Age: The age of the person

Work class: This column contains the working class of the person means which type of work he/she was doing.

Education: It contains education qualifications.

Education number: It contains the number of years of education.

Marital- Status: It contains the marital status of the person.

Relationship: It contains the relationship he/she had.

Race: It contains the race of the person like white or black.

Sex: It contains the gender of the person like male or female.

Capital gain: It contains the amount he earns on previous assets.

Capital loss: It contains the amount he lost on previous assets.

Hours per week: It contains the number of hours he/she worked in a week

Income: It contains the income labels like <=50k and >50k.

Data Pre-processing:

- First of all my dataset contains the “?” values in it. These values are converted into null values and then values are filled by the mode value of it. In my dataset work-class column, occupation column, and native country columns contain the "?" value.
- Using a label encoder categorical values are changed to continuous variables.

Python Tools (Libraries)

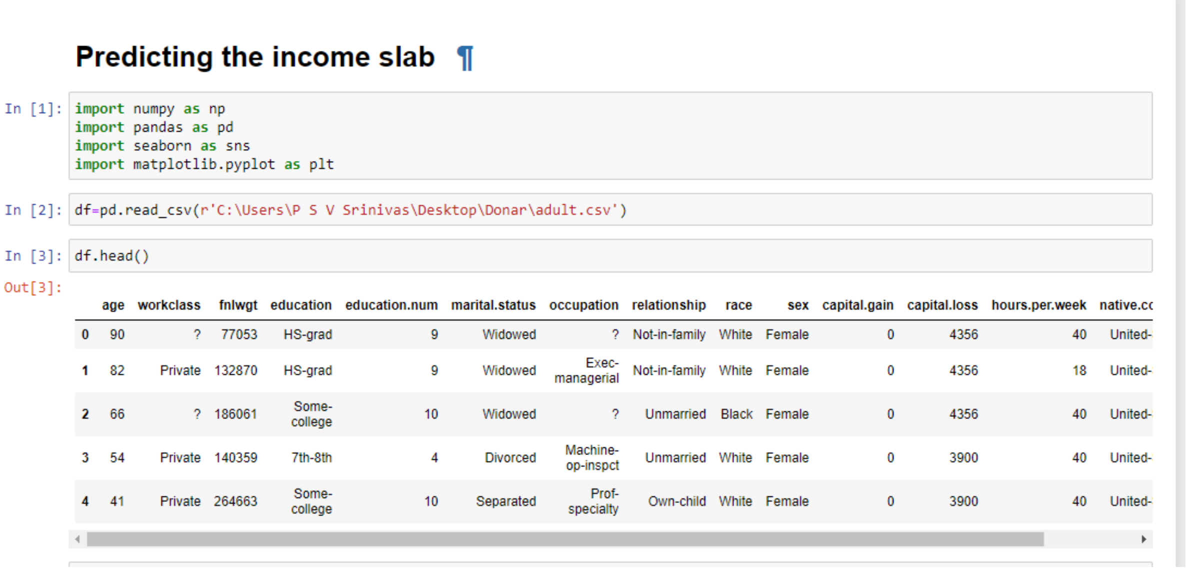
Python Tools (Libraries) used:

- pandas
- NumPy

- seaborn
- matplotlib
- sklearn

Python Code

Importing the all necessary libraries and the read the dataset using pandas



Predicting the income slab ¶

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

```
In [2]: df=pd.read_csv(r'C:\Users\P S V Srinivas\Desktop\Donar\adult.csv')
```

```
In [3]: df.head()
```

Out[3]:

	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-

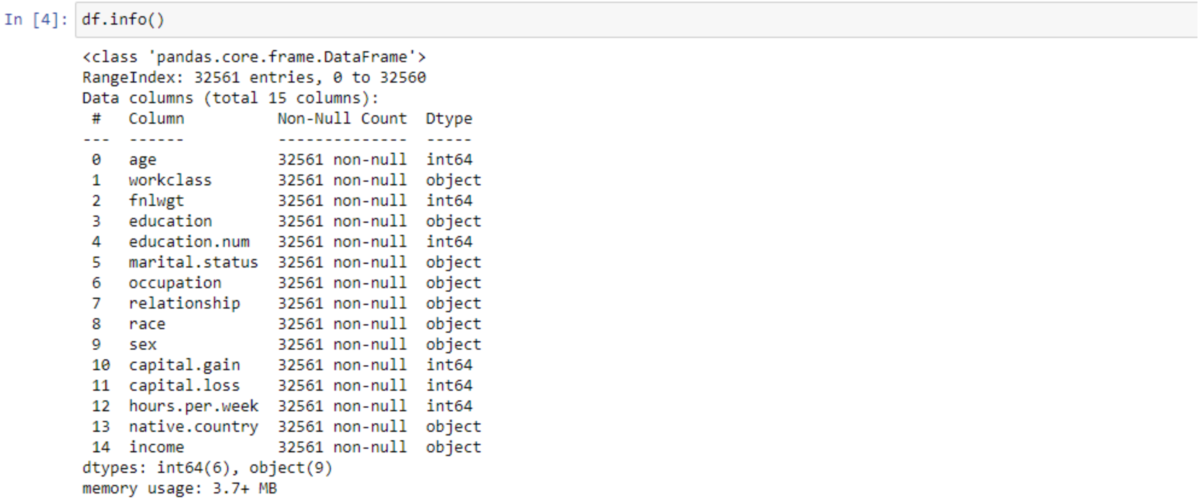
Using df.info() method checks the details of the dataset.

It contains

—— Range Index which tells the number of rows and columns

——Nonnull count in the columns

—Dtype which shows the datatype the column



```
In [4]: df.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 32561 entries, 0 to 32560
Data columns (total 15 columns):
 #   Column          Non-Null Count  Dtype 
---  ------          --------------  ----- 
 0   age             32561 non-null  int64 
 1   workclass       32561 non-null  object
 2   fnlwgt          32561 non-null  int64 
 3   education       32561 non-null  object
 4   education.num   32561 non-null  int64 
 5   marital.status  32561 non-null  object
 6   occupation      32561 non-null  object
 7   relationship    32561 non-null  object
 8   race            32561 non-null  object
 9   sex             32561 non-null  object
 10  capital.gain    32561 non-null  int64 
 11  capital.loss    32561 non-null  int64 
 12  hours.per.week  32561 non-null  int64 
 13  native.country  32561 non-null  object
 14  income          32561 non-null  object
dtypes: int64(6), object(9)
memory usage: 3.7+ MB
```

From the above command results:

Columns: 15

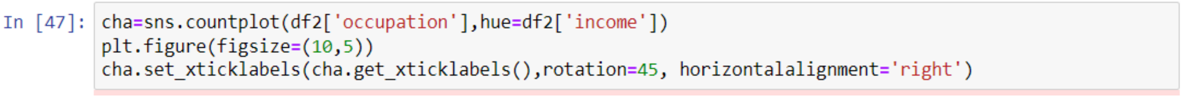
Rows: 32561

Dtypes: int64 – 6

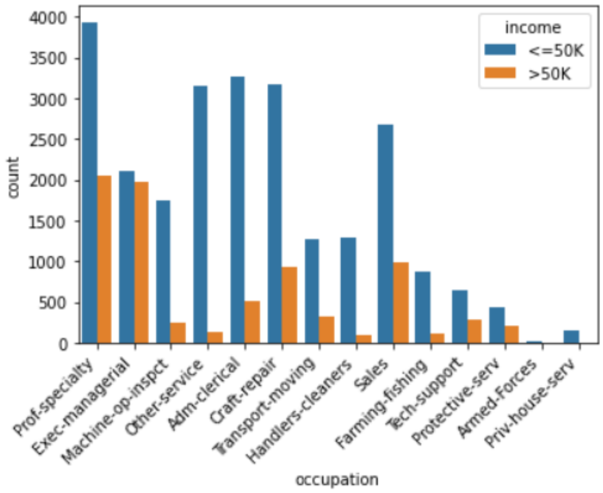
Object – 9

Visualization:

Count of each income slab based on occupation:



```
In [47]: cha=sns.countplot(df2['occupation'],hue=df2['income'])
         plt.figure(figsize=(10,5))
         cha.set_xticklabels(cha.get_xticklabels(),rotation=45, horizontalalignment='right')
```



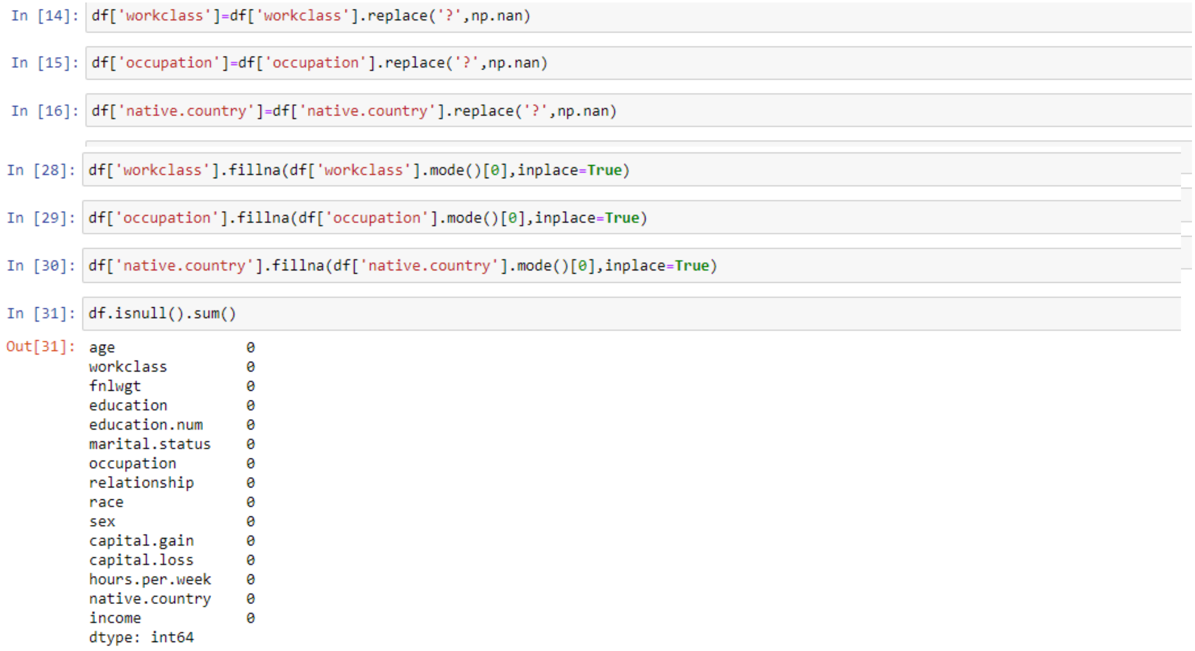
```
<Figure size 720x360 with 0 Axes>
```

Finding the Null values:

Now I am found the "?" values in the dataset. In my dataset work class column, occupation column and native country columns contain "?" values.

All the "?" values are replaced with Null values using NumPy.

Then those null values are replaced with the mode value of that particular column using pandas.

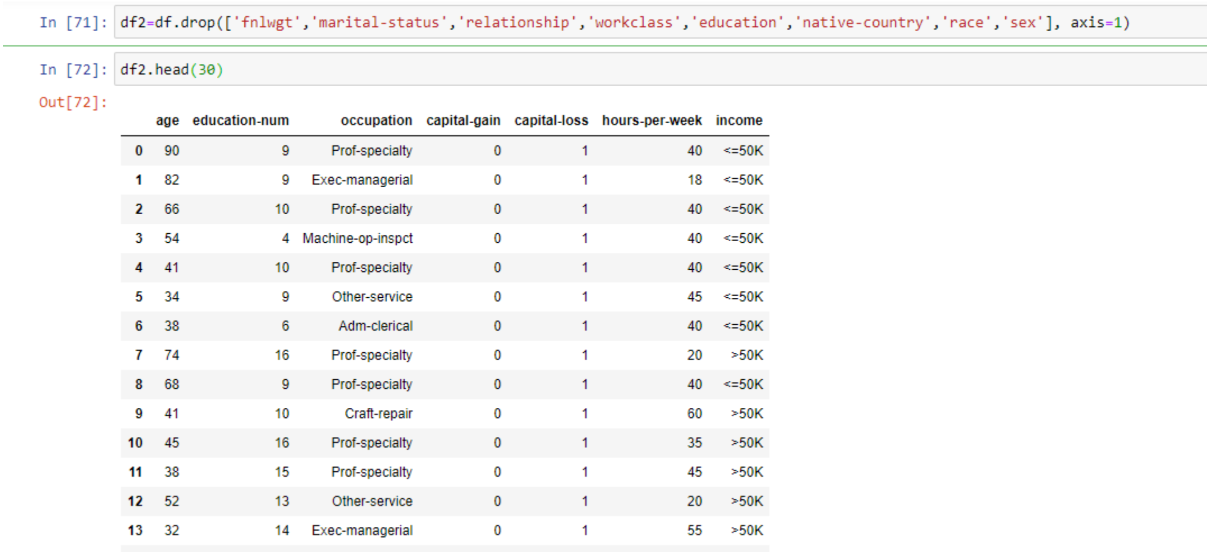


```
In [14]: df['workclass']=df['workclass'].replace('?',np.nan)

In [15]: df['occupation']=df['occupation'].replace('?',np.nan)

In [16]: df['native.country']=df['native.country'].replace('?',np.nan)

In [28]: df['workclass'].fillna(df['workclass'].mode()[0],inplace=True)

In [29]: df['occupation'].fillna(df['occupation'].mode()[0],inplace=True)

In [30]: df['native.country'].fillna(df['native.country'].mode()[0],inplace=True)

In [31]: df.isnull().sum()

Out[31]: 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
```

Finally no null values in my dataset.

Dropping the columns:

I dropped some columns which are not useful for my model.



```
In [71]: df2=df.drop(['fnlwgt','marital-status','relationship','workclass','education','native-country','race','sex'], axis=1)

In [72]: df2.head(30)
Out[72]:
```

	age	education-num	occupation	capital-gain	capital-loss	hours-per-week	income
0	90	9	Prof-specialty	0	1	40	<=50K
1	82	9	Exec-managerial	0	1	18	<=50K
2	66	10	Prof-specialty	0	1	40	<=50K
3	54	4	Machine-op-inspct	0	1	40	<=50K
4	41	10	Prof-specialty	0	1	40	<=50K
5	34	9	Other-service	0	1	45	<=50K
6	38	6	Adm-clerical	0	1	40	<=50K
7	74	16	Prof-specialty	0	1	20	>50K
8	68	9	Prof-specialty	0	1	40	<=50K
9	41	10	Craft-repair	0	1	60	>50K
10	45	16	Prof-specialty	0	1	35	>50K
11	38	15	Prof-specialty	0	1	45	>50K
12	52	13	Other-service	0	1	20	>50K
13	32	14	Exec-managerial	0	1	55	>50K

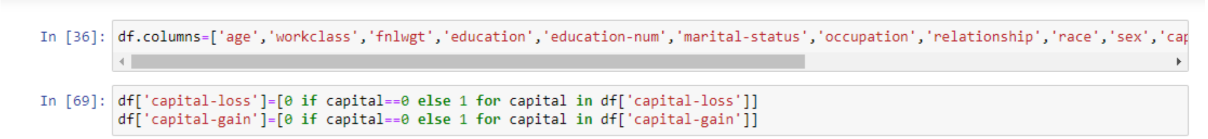
Label Encoding:

It is a process through which we can transform categorical values into numeric values.

This process is helpful while classification, since the model, can easily classify the numeric values.

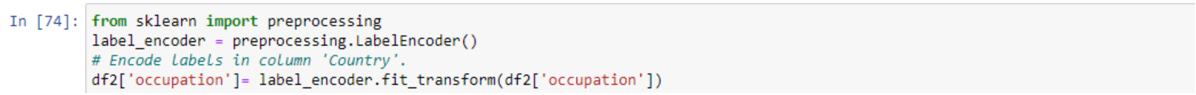
Using sklearn library we perform label encoding.

Without sklearn:

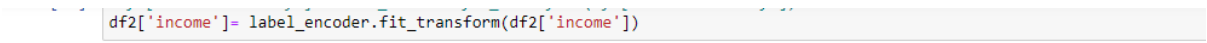


```
In [36]: df.columns=['age','workclass','fnlwgt','education','education-num','marital-status','occupation','relationship','race','sex','cap

In [69]: df['capital-loss']=[0 if capital==0 else 1 for capital in df['capital-loss']]
         df['capital-gain']=[0 if capital==0 else 1 for capital in df['capital-gain']]
```

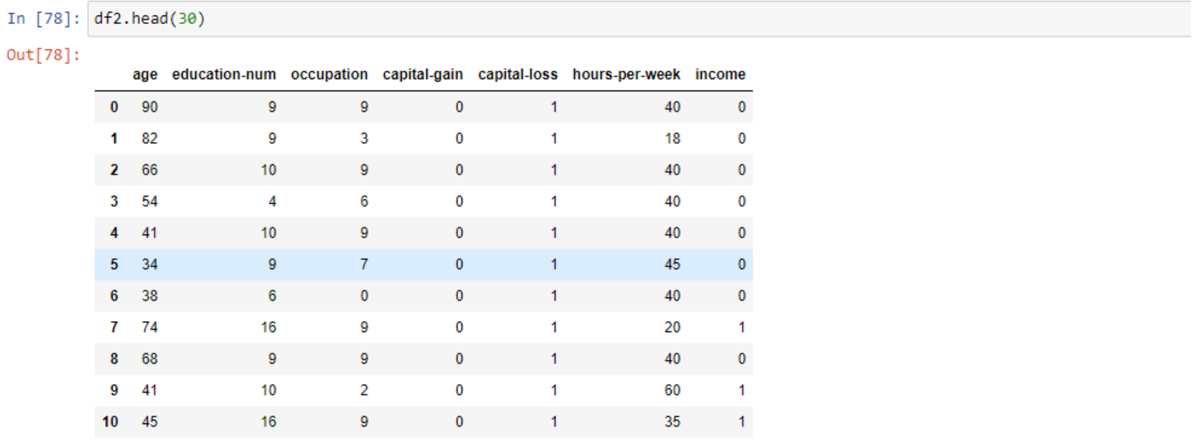
With sklearn:



```
In [74]: from sklearn import preprocessing
         label_encoder = preprocessing.LabelEncoder()
         # Encode labels in column 'Country'.
         df2['occupation']= label_encoder.fit_transform(df2['occupation'])
```



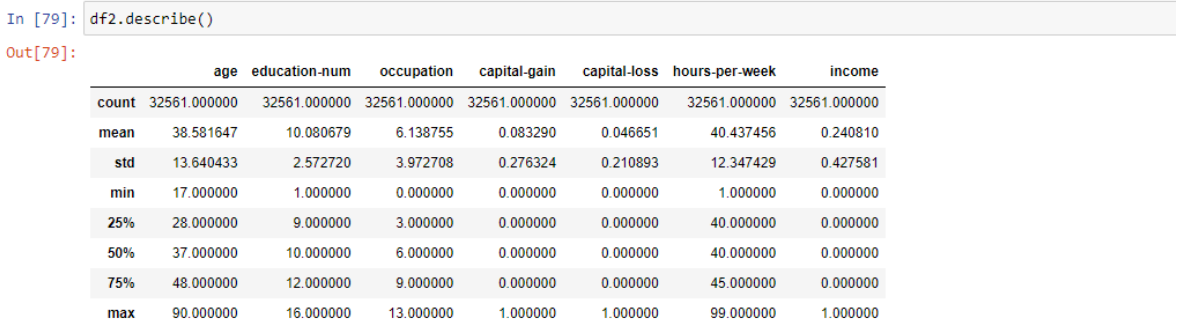
```
         df2['income']= label_encoder.fit_transform(df2['income'])
```



```
In [78]: df2.head(30)
```

Out[78]:

	age	education-num	occupation	capital-gain	capital-loss	hours-per-week	income
0	90	9	9	0	1	40	0
1	82	9	3	0	1	18	0
2	66	10	9	0	1	40	0
3	54	4	6	0	1	40	0
4	41	10	9	0	1	40	0
5	34	9	7	0	1	45	0
6	38	6	0	0	1	40	0
7	74	16	9	0	1	20	1
8	68	9	9	0	1	40	0
9	41	10	2	0	1	60	1
10	45	16	9	0	1	35	1



```
In [79]: df2.describe()
```

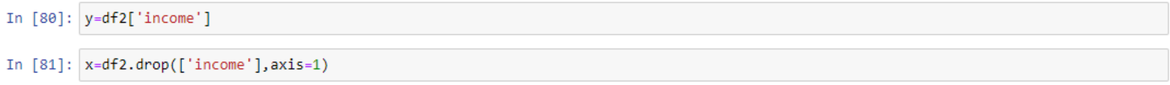
Out[79]:

	age	education-num	occupation	capital-gain	capital-loss	hours-per-week	income
count	32561.000000	32561.000000	32561.000000	32561.000000	32561.000000	32561.000000	32561.000000
mean	38.581647	10.080679	6.138755	0.083290	0.046651	40.437456	0.240810
std	13.640433	2.572720	3.972708	0.276324	0.210893	12.347429	0.427581
min	17.000000	1.000000	0.000000	0.000000	0.000000	1.000000	0.000000
25%	28.000000	9.000000	3.000000	0.000000	0.000000	40.000000	0.000000
50%	37.000000	10.000000	6.000000	0.000000	0.000000	40.000000	0.000000
75%	48.000000	12.000000	9.000000	0.000000	0.000000	45.000000	0.000000
max	90.000000	16.000000	13.000000	1.000000	1.000000	99.000000	1.000000

Extraction of features:

x contains all the columns in the dataset except the income column.

Y contains the only income column of the dataset.

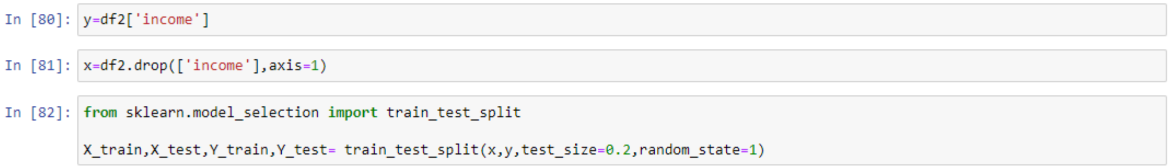


```
In [80]: y=df2['income']

In [81]: x=df2.drop(['income'],axis=1)
```

Splitting the dataset:

Here I am splitting the dataset into the training set and testing set. Using sklearn.modelselction module.

And importing the train test split feature.



```
In [80]: y=df2['income']

In [81]: x=df2.drop(['income'],axis=1)

In [82]: from sklearn.model_selection import train_test_split

         X_train,X_test,Y_train,Y_test= train_test_split(x,y,test_size=0.2,random_state=1)
```

Classification:

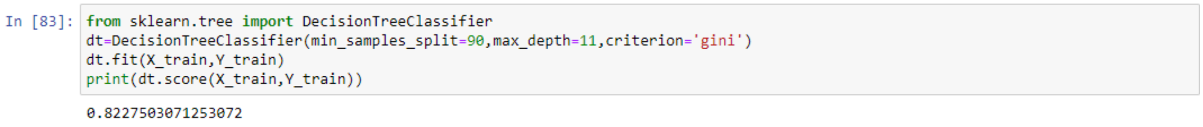
Using the Decision tree classifier I classified my model.

Decision Tree Classifier:

A Decision Tree is a Supervised learning technique that can be used for both classification and Regression problems, but mostly it is preferred for solving Classification problems.

The Gini index is a measure of impurity or purity used while creating a decision tree in the CART(Classification and Regression Tree) algorithm.

It only creates binary splits, and the CART algorithm uses the Gini index to create binary splits.

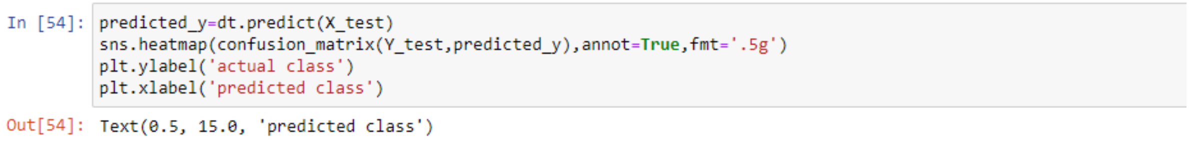


```
In [83]: from sklearn.tree import DecisionTreeClassifier
         dt=DecisionTreeClassifier(min_samples_split=90,max_depth=11,criterion='gini')
         dt.fit(X_train,Y_train)
         print(dt.score(X_train,Y_train))

         0.8227503071253072
```

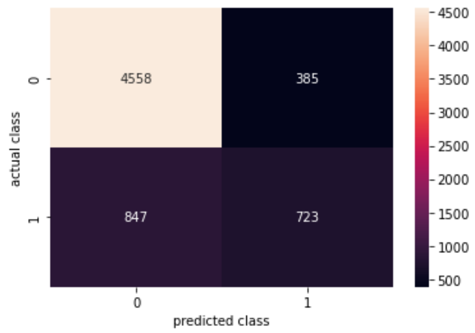
Here I got an accuracy of 82% for my model.

Evaluation:

Here I evaluated my model using the confusion matrix, which was imported from the sklearn module.



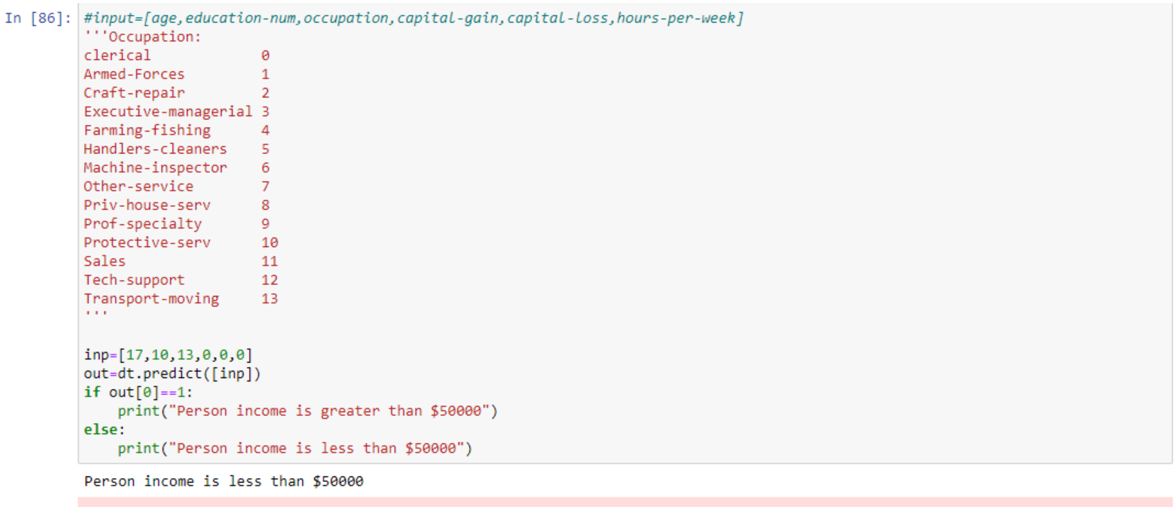
```
In [54]: predicted_y=dt.predict(X_test)
         sns.heatmap(confusion_matrix(Y_test,predicted_y),annot=True,fmt='.5g')
         plt.ylabel('actual class')
         plt.xlabel('predicted class')

Out[54]: Text(0.5, 15.0, 'predicted class')
```



Correct Prediction: 4558 + 723

Incorrect Prediction: 847 +385

Prediction:



```
In [86]: #input=[age,education-num,occupation,capital-gain,capital-loss,hours-per-week]
         '''Occupation:
         clerical            0
         Armed-Forces        1
         Craft-repair        2
         Executive-managerial 3
         Farming-fishing     4
         Handlers-cleaners   5
         Machine-inspector   6
         Other-service       7
         Priv-house-serv     8
         Prof-specialty      9
         Protective-serv     10
         Sales               11
         Tech-support        12
         Transport-moving    13
         '''

         inp=[17,10,13,0,0,0]
         out=dt.predict([inp])
         if out[0]==1:
             print("Person income is greater than $50000")
         else:
             print("Person income is less than $50000")

         Person income is less than $50000
```

Resources & References

https://www.kaggle.com/datasets/rdcmdev/adult-income-dataset

https://medium.com/

https://towardsdatascience.com/

Github link:

https://github.com/PSVS/Data-Science-With-Python/

Subject Faculty:

Asst. Prof. Ch. Satish Kumar,

Department of Computer Science and Engineering,

RGUKT Srikakulam.