Data Mining Assignment 1

Data mining CSE-5334

Report on Python assignment

INTRODUCTION:

Python is an Object Oriented Language and also an interpreted language.
The real world entities can be represented using python. DataMining is used to create value from data. We assume some hypothesis on data , perform statistical testing to check our hypothesis and finally draw conclusions and finding patterns in data.
Python can be used to draw conclusions from the data as we have many libraries and modules like matplotlib and seaborn to visualize data.
The integrated Development Environment used is Jupyter Notebook by ANACONDA. Numpy is used to implement multidimensional arrays in python which can go beyond some operations of a LIST in Python
Pandas is used to create DataFrames so that the data can be represented in the form of tables which is easy to understand and also helps in analysis and manipulation of Data.

```
# special IPython command to prepare the notebook for matplotlib
%matplotlib inline

#Array processing
import numpy as np
#Data analysis, wrangling and common exploratory operations
import pandas as pd
from pandas import Series, DataFrame
from itertools import chain

#For visualization. Matplotlib for basic viz and seaborn for more stylish figures
import matplotlib.pyplot as plt
import seaborn as sns
```

%matplotlib inline
Import numpy as np
Import pandas as pd
From pandas import series, dataframe
From itertools import chain
Import matplotlib.pyplot as plt
Import seaborn as sns

Retrieving the Data:
We use pandas to read the data in the jupyter notebook and we can see the data in dataframes which shows in rows and columns. The python command shown below is used to see the dataset.

Glimpse of Data:

The dataset contains of 15 columns and 43958 rows The attributes or the columns namely: df_data.info()

```
#   Column            Non-Null Count   Dtype
---  ------            --------------   -----
0   age               43957 non-null   int64
1   employment        41459 non-null   object
2   fw                43957 non-null   int64
3   education         43957 non-null   object
4   years-education   43957 non-null   int64
5   marital-status    43957 non-null   object
6   job               41451 non-null   object
7   bond              43957 non-null   object
8   race              43957 non-null   object
9   gender            43957 non-null   object
10  capital-gain      43957 non-null   int64
11  capital-loss      43957 non-null   int64
12  hours-per-week    43957 non-null   int64
13  native-country    43957 non-null   object
14  income > 50K      43957 non-null   object
```

	age	employment	fw	education	years-education	marital-status	job	bond	race	gender	capital-gain	capital-loss	hours-per-week	native-country	income > 50K
0	67	Private	366425	Doctorate	16	Divorced	Exec-managerial	Not-in-family	White	Male	99999	0	60	United-States	Yes
1	17	Private	244602	12th	8	Never-married	Other-service	Own-child	White	Male	0	0	15	United-States	No
2	31	Private	174201	Bachelors	13	Married-civ-spouse	Exec-managerial	Husband	White	Male	0	0	40	United-States	Yes
3	58	State-gov	110199	7th-8th	4	Married-civ-spouse	Transport-moving	Husband	White	Male	0	0	40	United-States	No
4	25	State-gov	149248	Some-college	10	Never-married	Other-service	Not-in-family	Black	Male	0	0	40	United-States	No

- The file can be read using pd.read.csv(),we can pass the path of the file in the function.

we can print the first 5 rows of the data using df_data.head() command.

Check for Missing data:

We need to check the missing values or the null values because without filling out or removing null values it will cause overfitting or underfitting. In order to avoid this and extract good patterns we need to identiy the missing or null values using the command. • df_data[column].isnull()

Data Exploration:

Task 1-a: Print the details of the df_data data frame (information such as number of rows, columns, name of the columns, etc)

- Using the command df_data.info()

- It gives the details of the column name , the non null values present in the columns and the data type of the values in the column.
- It also gives us brief idea of the null values present and helps us to fill those null values based on the data type and gives us the idea on how many values should be filled and how many rows should be dropped.

Task 1-b: Print name of all the countries used in the dataset:

- df_data['native-country'].
- set(df_data['native-country']).
- The above command is used to print the name of all the countries present in the dataset.
- The set is used to create the names where we don't have duplicate names of the countries since set doesn't give duplicate values.

Task 1-c: print descriptive deatils for "employment column of the df_data"

- df_data['employment'].describe().
- the above command is used to give the details of the column which the describe() command gives top row value , datatype and number of rows in total.

Task 1-d: Some of the entries in the columns are undefined. Determine which columns contain these undefined entries and print the count of these undefined entries for each column respectively.

- The command df_data[column].isnull())gives us the empty values or null values.

Number of doctorates that they are married Task 2-a: .

- To find doctorates we need to use str.comtains and check 2 conditions in which both hould be true that is they are married and educational qualification is doctorate.
- We can use the below command that is len() fun and 2 conditions in the length function.
- len(df_data[df_data['education'].str.contains('Doctorate')&df_data['maritalstatus'].str.contains('Married-civ-spouse')])).

Task 2-b: Retrieve and print the records of Americans over 40 years who have been putting in more than 40 hours a week of work

- By using below code, we have got the details of americans over 40years((df_data['age']>40) who have been putting in more than 40 hours a week of work(df_data['hours-per-week']>40).
- df_data[(df_data['native-country'].str.contains('United-States'))& (df_data['age']>40) & (df_data['hours-per-week']>40)].

Discover the Task 2-c: top 10 nations with the highest incomes and print a list of them.:

- We have utilized the following code below.
- df_data.sort_values(by=['capital-gain'],ascending=False).head(11).
- By using this function we have printed the top 10 country according to capital gain in descending order.

Task 2-d: Fetch and print the records of people having doctoral degrees and are divorced.:

- By using the code below, we can display doctorate in education column whose martial status is divorced.
- df_data[df_data['education'].str.contains('Doctorate')&(df_data['maritalstatus'].str.contains('Divorced'))] .

Task 3-a: Display the countplot for education level of those with an income over 50K.:

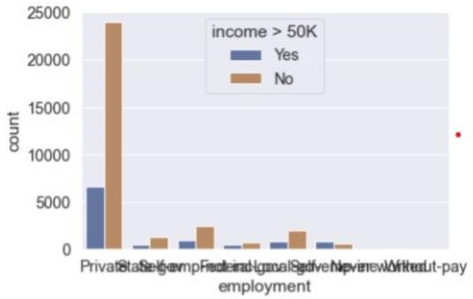
- By using the code below, we have visualized the data using countplot(sns.countplot) for education level of those with an income over 50K(x='education',hue='income > 50K', data=df_data).
- sns.countplot(x='education',hue='income > 50K', data=df_data) plt.show().

Task 3-b: Display a pie chart that represents native country and display percentages in legend respectively.:

- By using the code below, we have visualized the data using pie chart that represents native country and display percentages in legend(autopct='%1.1f%%) respectively.
- threeb=df_data['nativecountry'].value_counts().plot(kind='pie',autopct='%1.1f%%') plt.show().

Task 4: Find out 'interesting' information from the dataset. Give two insights and Create a visualization for each of the insights. Explain in a few lines your reasoning.Your work's uniqueness and quality will be taken into account when evaluating your work (having a meaningful result and an aesthetic visualization).:

In [36]:
```
#4-a
sns.countplot(x='employment',hue='income > 50K', data=df_data)
plt.show()
```



In [24]:
```
#4-b
sns.countplot(x='bond',hue='income > 50K', data=df_data)
plt.show()
```