Name: Anwar Qarout.

ID: 1180774.

Instructor: Dr. Ahmad Abu Snena.

Date: May 9th, 2021.

Introduction/Overview

We are required to design an R-Programming or Python program to deal with a dataset, which is a survey of real Palestinian civilians collected in 2011 and 2014. The program will print the summary statistics of all attributes, print the histograms of all features, find the correlation matrix and confusion matrix, split the data into training and testing, and use Decision Tree Classification Model to predict whether the family has a kitchen, which is the target class.

In this project, I tried to print as many plots and accuracy measures as possible, to assure full understanding and implementation of this Machine Learning algorithm, as well as ensure its accuracy.

Here are the **main features** that I chose in order to predict whether the family has a kitchen, which has a column name H14:

- Number of rooms used for sleep (sleep area) - column named H9.
- Is the family on loan, borrowing or debt? - column named C09_1A.
- Dwelling connection to utilities network (Water) - column named H12_1B.
- Dwelling connection to utilities network (Electricity) - column named H12_2B.
- What kind of dwelling unit does the family live in? - column named H1.
- How far is your home from public transportation? - column named T3_1_1.
- How many freezer are available to the house hold? - column named H22_3.
- How far is your home from the nearest shopping center? - column named T3_9_1.
- What are the first major impacts on your family's situation? - column named C12_21.
- How many iPads/Tablets are available to the house hold? - column named H22_31.
- In the last 30 days, has your family sold assets property? - column named C13_2A.
- In the last 30 days, has your family looked for overtime job? - column named C13_8A.

- In general, what do you consider your family? - column named I04_1.
- In the past 30 days, has your family used family savings? - column named C13_3A.
- In the past 30 days, has your family forced to take children out of school? - column named C13_12A.
- How many smart phones available to the house hold? - column named H22_28.
- Over the past week, how many days has the family consumed Red Meat? - column named E801_6A.
- How many Playstation/Xbox are available to the house hold?- column named H22_34.
- How many central heating are available to the house hold? - column named H22_17.

Code Explanation & Outputs

In my code, lines 1 to 14 are used to import useful libraries that allow us to run different tests, as well as print various graphs.

```
1   import pandas
2   import pandas as pd
3   import numpy as np
4   import seaborn as sn
5   import matplotlib.pyplot as plt
6   import sklearn
7   from pandas import DataFrame
8   from sklearn.compose import ColumnTransformer
9   from sklearn.metrics import confusion_matrix
10  from sklearn.model_selection import train_test_split
11  from sklearn import metrics
12  from sklearn.preprocessing import OneHotEncoder, StandardScaler, LabelBinarizer
13  from sklearn.tree import DecisionTreeClassifier
14  from sklearn.preprocessing import OneHotEncoder
```

Next, in line 15, the dataset is read, and put in the variable “data”.

```
data = pd.read_spss("SefSec_2014_HH_weight new.sav")
```

Note that some of the attributes we have in the dataset file are not numerical. In lines 22 to 52, I converted these attributes from non-numerical to numerical, by giving them dummy numerical data, in order to include them in different tests later on; because these tests only apply to numerical data. Here is an example:

```
data['H14'] = data['H14'].cat.codes
```

This piece of code takes the column H14, and assigns numeric values to string labels. So for example, if the class labels are Yes/No, it will give them the values 0 and 1.

Up next, I used the describe() method in pandas library on each of the features, which prints the summary statistics for a column. This helps me understand the data better.

Here is an example summary statistic for column H22_3, how many freezers are available to the house hold. This is in line 54. Here is the output:

```
count    8237.000000
mean        0.070657
std         0.257684
min         0.000000
25%         0.000000
50%         0.000000
75%         0.000000
max         2.000000
Name: H22_3, dtype: float64
```

Next, I defined an array named temp, and subsetted all the features that I want from the original, big dataset. This is done in line 59.

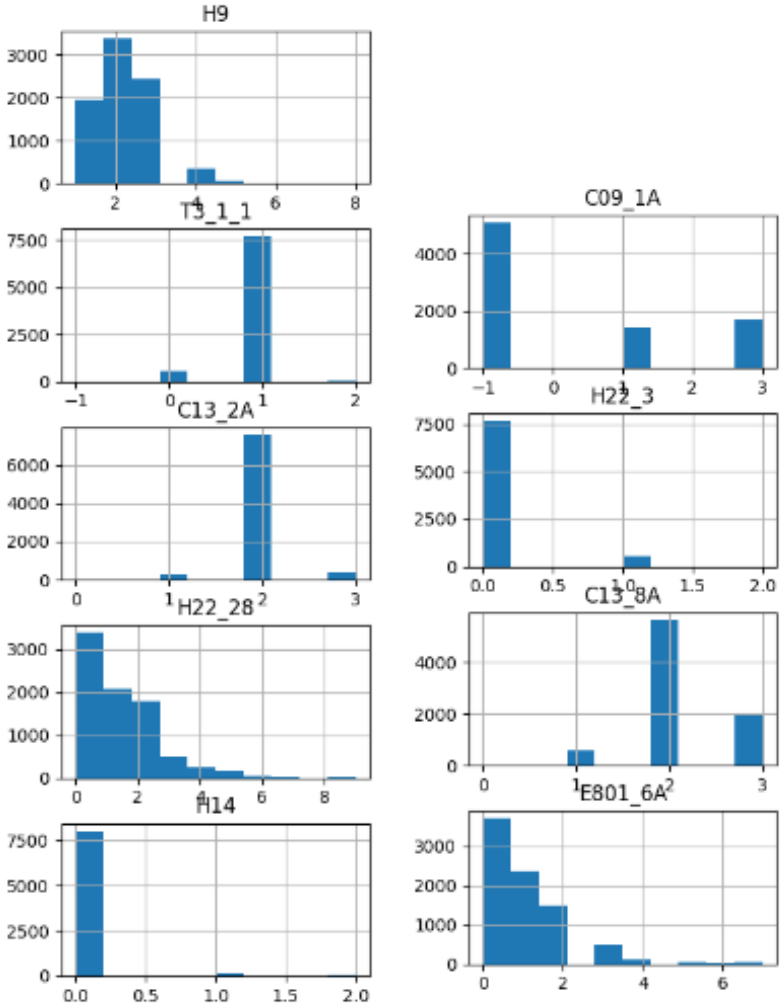
```
temp = data[['H9','C09_1A','H12_1B','H12_2B','H1','T3_1_1',
```

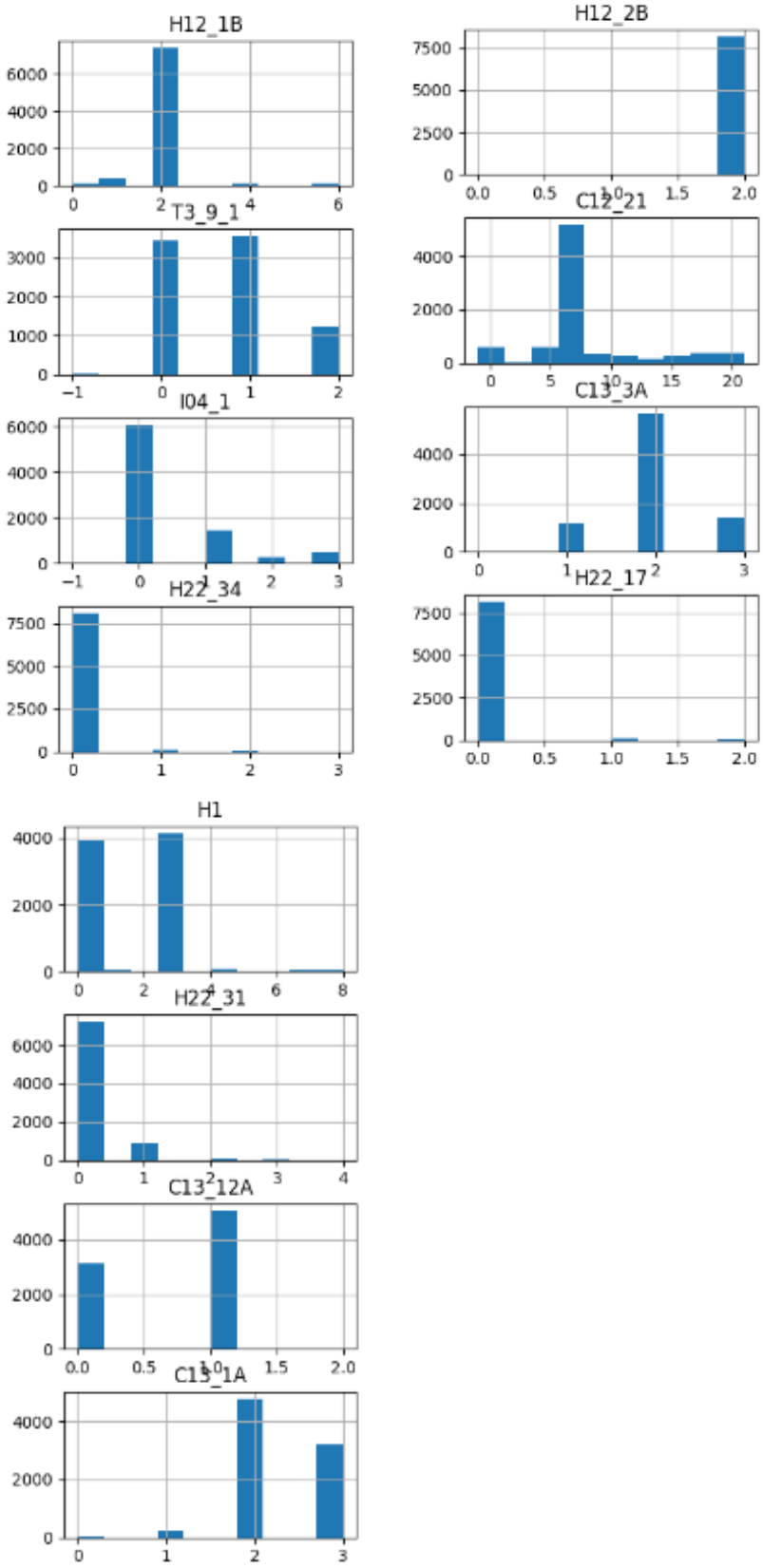
.....

For all the features that I've chosen, I plotted histograms for better understanding of the data. This is in lines 61 and 62.

```
temp.hist( figsize = (20,10))
plt.show()
```

Here are the histograms generated by my Python program:

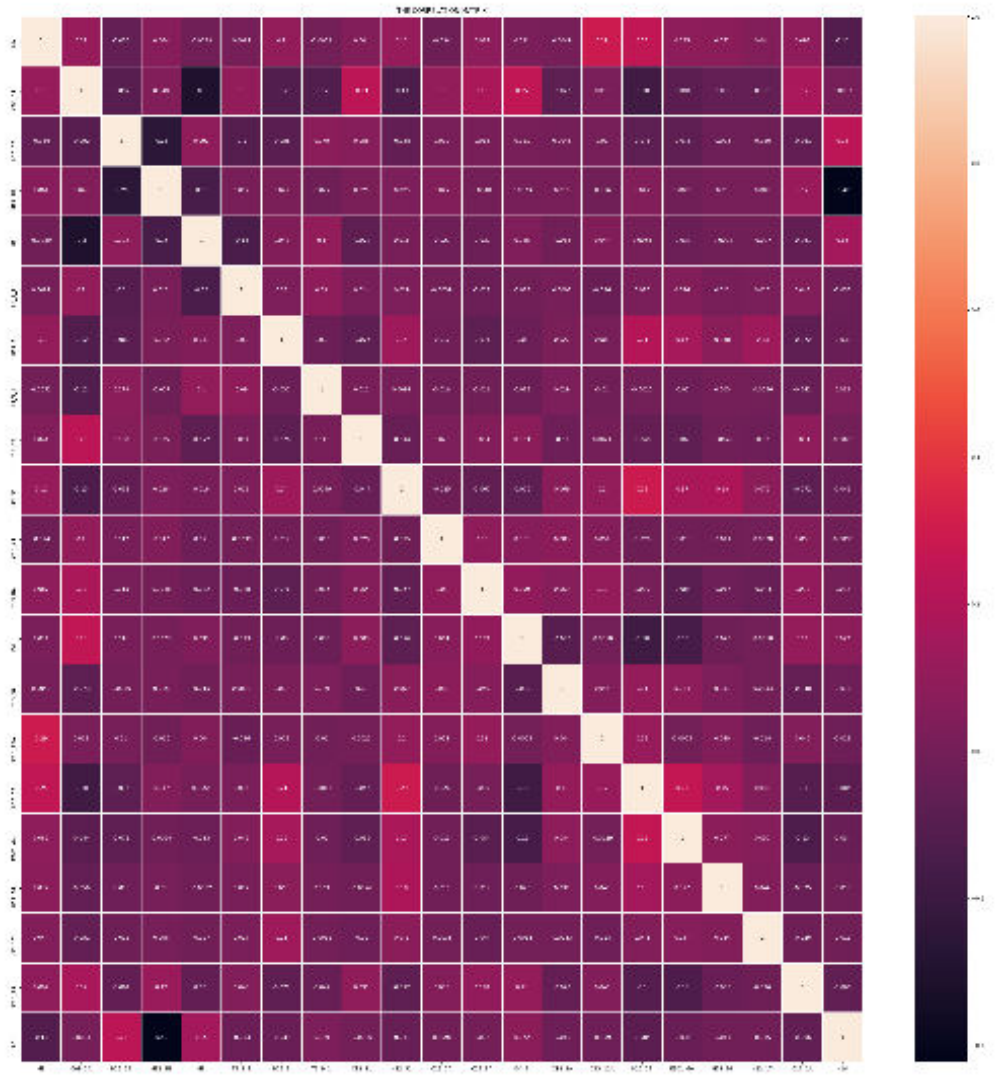




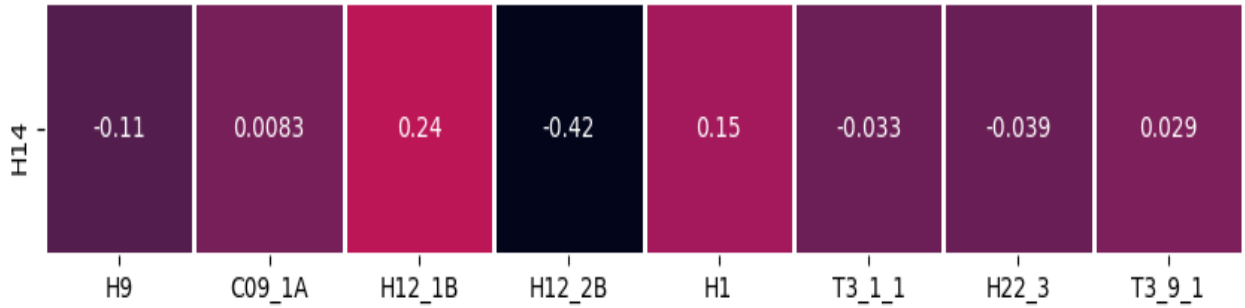
These histograms are obviously plotted after converting all String data to numerical by giving them dummy values.

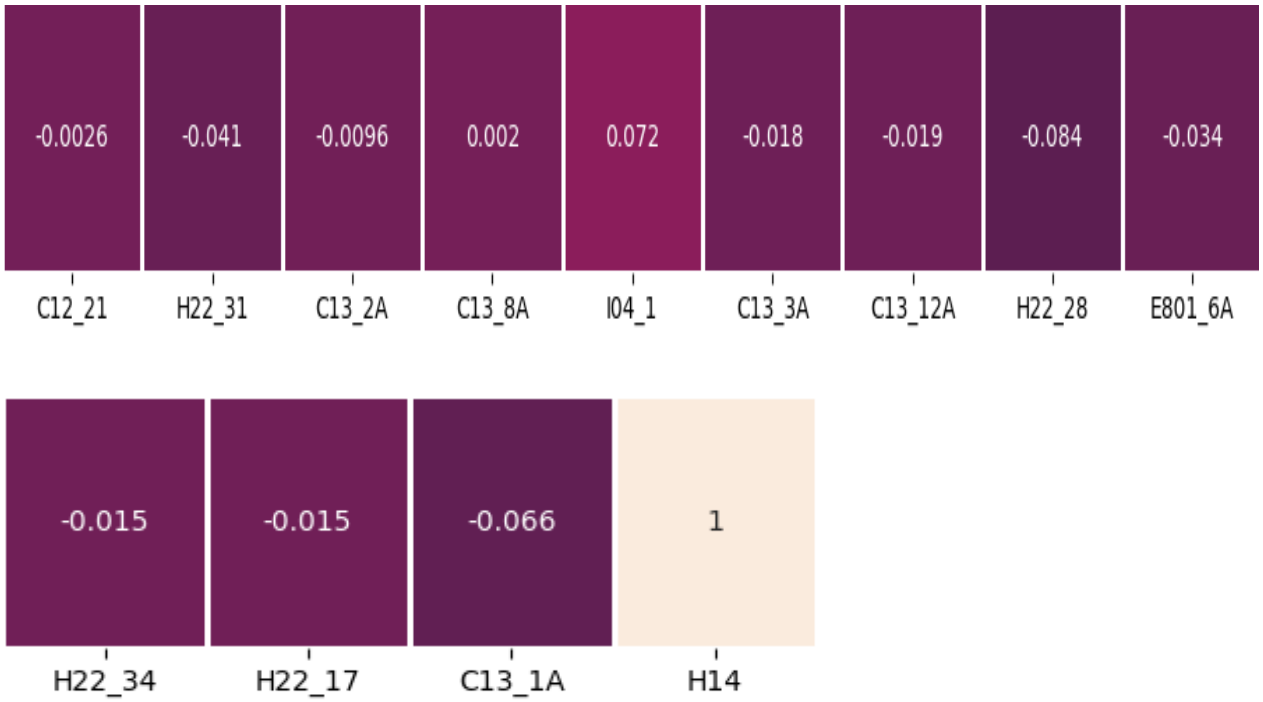
Up next, I plotted the Correlation Matrix, to make sure that the features I have chosen are related to each other and to the target class. This is in lines 63 to 66.

Here is the result Correlation Matrix:



As we can see, its a big, 20x20 matrix. However, I'm interested in one row (or column), the one that gives me the correlation between my target class and the rest of the features. This is the last row in this matrix.





Correlation Matrix Summary:

The correlation matrix shows the relationship between all features. It is always between -1 and 1. If the correlation between two variables is positive, it means there is a positive relationship between these two features, so when one increases, the other one increases as well. Similarly, if the correlation is negative, it means there is a negative relationship, meaning if one decreases, the other one increases, and vise versa. If the correlation is zero, it means there is no relationship between them, meaning they don't affect each other.

The correlations are mostly good, although some of them are close to zero which won't affect the learning process very well. We can also notice there is a correlation of 1, which is the maximum correlation value. This is between the target value and itself, so that is why it is 1.

Up next, I split the data to 70% training and 30% testing. This is executed in lines 68 to 71.

```
X = temp[['C13_1A','H9','C09_1A','H12_1B','H12_2B','H1','T3_1_1','H22_3','T3_9_1','C12_21'
y=temp['H14']

X_train, X_test, y_train, y_test = train_test_split(X, y, random_state=1,test_size=0.3)
```

Then I defined my decision tree classifier, and trained it in lines 72 and 73.

```
clf = DecisionTreeClassifier()
clf.fit(X_train,y_train)
```

Next, I predicted the output of the testing set, and compared it with the actual value, and calculated the accuracy. This is in lines 75 and 79.

```
y_pred = clf.predict(X_test)
y_test = np.array(list(y_test))
y_pred = np.array(y_pred)

print("accuracy : ",metrics.accuracy_score(y_test,y_pred))
```

Here is the output:

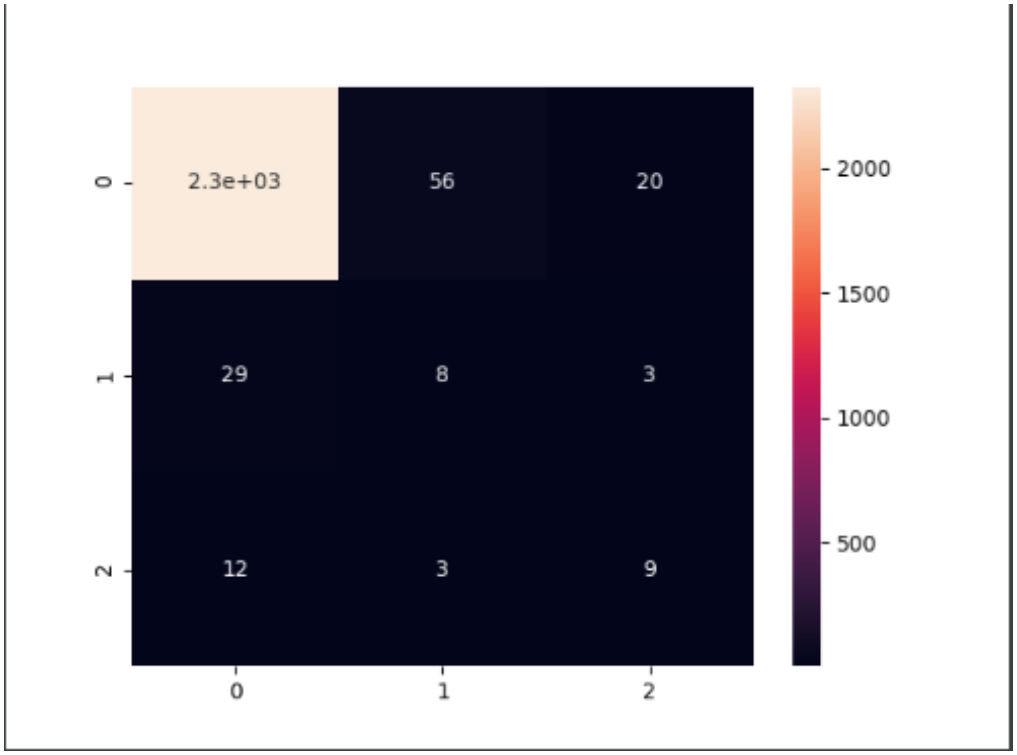
```
accuracy :  0.9501822600243013
```

As we can see, my model has an accuracy of 95%, which is very good.

Finally, I plotted the Confusion Matrix, in lines 80 to 82:

```
cf_matrix = confusion_matrix(y_test, y_pred)
sn.heatmap(cf_matrix,annot=True)
plt.show()
```

Here is the result:



On the X-axis, the prediction is shown, while on the Y-axis, the actual is shown. So, in the first block, it is comparing value 0 with 0, so it is the number of results predicted as 0, and it is actually 0, and the count is 2.3e+03. In the second block is the predicted 0 but actual 1, so they are wrongly predicted, and their count is 56, etc.

References

https://www.youtube.com/watch?v=9P4T7SU3SZc

https://www.datacamp.com/community/tutorials/decision-tree-classification-python

https://scikit-learn.org/stable/modules/tree.html

https://medium.com/pursuitnotes/decision-tree-classification-in-9-steps-with-python-600c85ef56de

https://www.geeksforgeeks.org/decision-tree-implementation-python/