COVID-19 PREDICTIVE ANALYSIS – CRISP DM

1. Business Understanding:

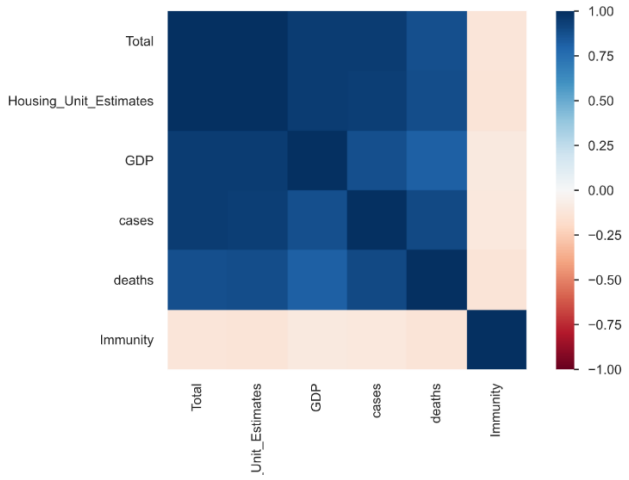
- As the head of a research team of a Pharmaceutical Company, I am conducting a COVID-19 predictive analysis by county considering major factors; total population of the county, GDP of the county and the housing unit estimates of each county
- The objective of the analysis is to determine which factor are responsible for immunity of people living in a county
- The analysis will assist in determining ways and amount vaccines to be distributed

2. Data Understanding:

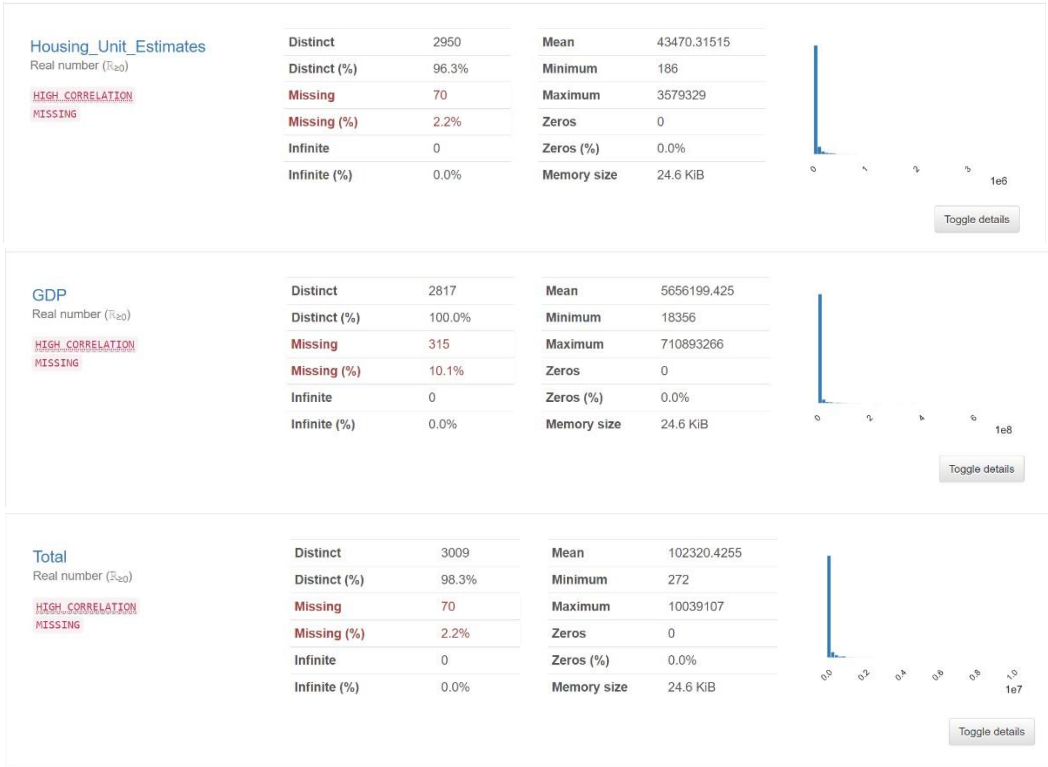
- A total of 4 datasets are available to be merged and generate one dataset to perform predictive analysis
- The first dataset includes number of COVID-19 cases and deaths per county for each state
- The second dataset includes GDP of each county for year 2018
- The third dataset includes population for different age groups, races and gender for all counties
- The fourth dataset includes housing unit estimates for all counties

♦ **Exploratory data analysis:**

- The merged dataset includes housing unit estimates, total population, GDP, COVID-19 cases, and deaths
- The data included daily cases and deaths from March to 18th October, the cumulative data was reduced to number of cases on 18th October
- The table below shows the correlation between each variable



- The image shows total population is highly related housing unit and GDP
- GDP is more correlated to number of cases than deaths. In this case we can say that where there is more development or infrastructure, people are less likely to die.
- There were several missing values as shown in the images below



3. Data Preparation:

- For the census dataset, a filter was applied to extract population of year 2019. After eliminating all the other variables, state, county, and total population remained
- The other dataset required not much of transformation
- All four datasets were merged using the pandas merge() function

4. Modeling:

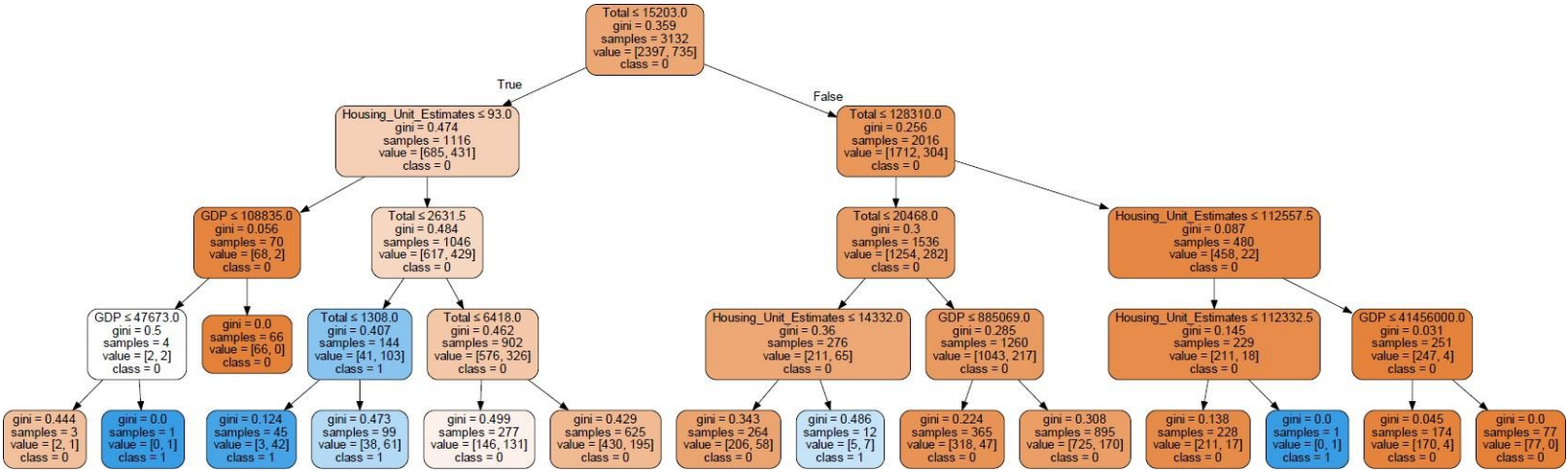
- **Decision Tree Classification:**
- We are going to use two data mining models for this data set, Decision Tree and Support vector machine
- Scikit learn library will be used to apply the model to the transformed dataset
- The first step is to determine the target variable, in this case the target variable was achieved from number of deaths and cases

$$\text{Death ratio} = \frac{\text{deaths}}{\text{cases}}$$

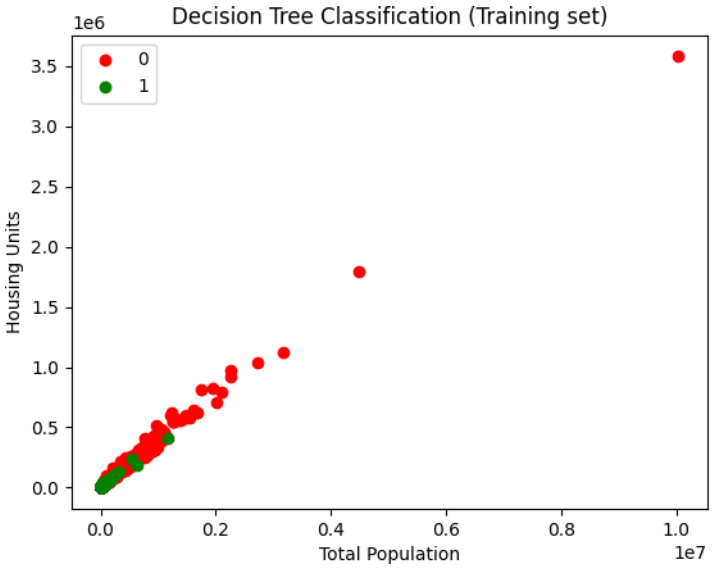
- The threshold value will be the product of maximum value of death ratio obtained and 0.05
- The next step is to import modules and packages from scikit learn library as shown in the image below

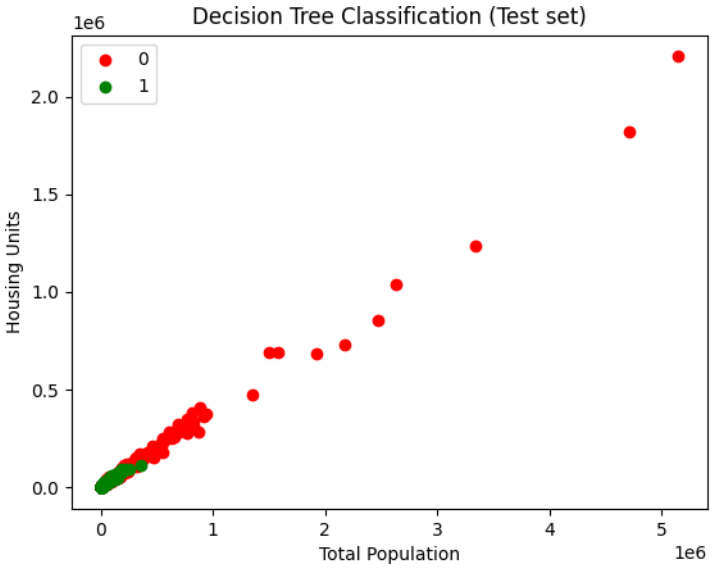
```
from sklearn.datasets import load_iris
from sklearn.model_selection import train_test_split
from sklearn import tree
from sklearn.metrics import confusion_matrix,accuracy_score, classification_report
from sklearn.metrics import roc_curve, roc_auc_score
from graphviz import Source
from matplotlib import pyplot as plt
import numpy as np
import pandas as pd
```

- The train_test_split package will be used to split the dataset into training set and testing set
- The tree package is used to generate decision tree analysis
- After importing dataset, we define the X and Y variables, into X we write Total population and housing units, into Y we have our target variable
- Fit the decision tree classifier onto the dataset using DecisionTreeClassifier() function
- The next step is to split the dataset into training set and testing set and predict the test dataset
- In this case we generated the decision tree as shown in the figure below

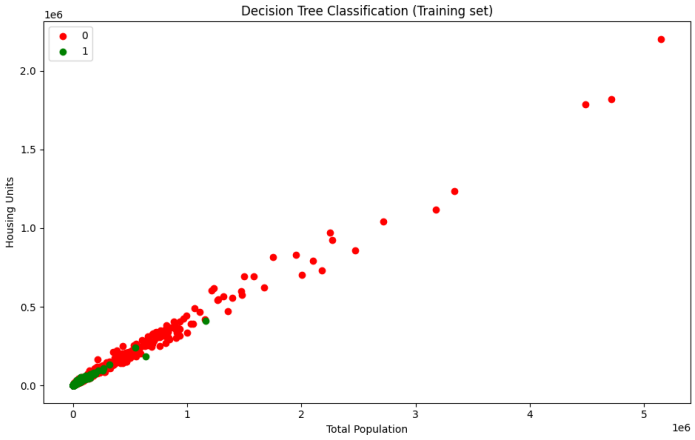


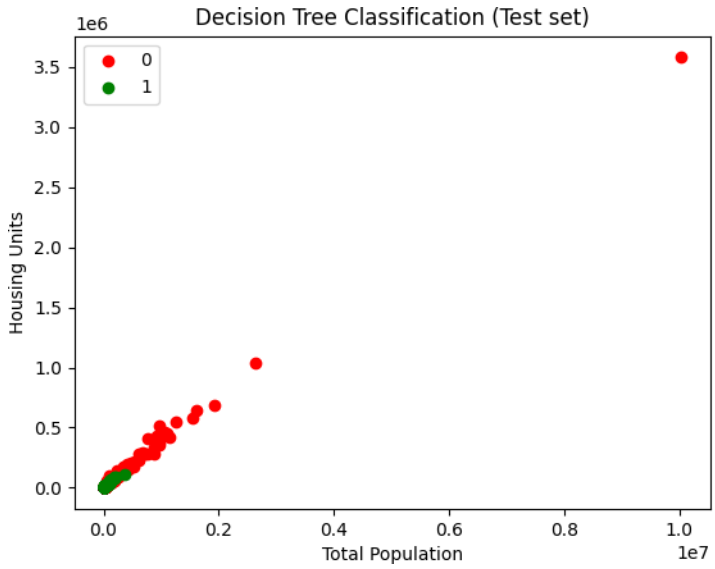
- From the tree above we can conclude that counties having less than 93 housing unit, a population less than 15203 people and a GDP between 47673 has less deaths.
- The two figures below show classification of the target variable plotted for training set and testing set


Decision Tree Classification (Training set)
1e6
0
1
3.5
3.0
2.5
2.0
1.5
1.0
0.5
0.0
Housing Units
0.0
0.2
0.4
0.6
0.8
1.0
1e7
Total Population


Decision Tree Classification (Test set)
1e6
0
1
2.0
1.5
1.0
0.5
0.0
Housing Units
0
1
2
3
4
5
1e6
Total Population

- **Support Vector Machine:**
- The only difference between the python code of decision tree and support vector machine is the use of the package SVC from module SVM of scikit learn library
- The image below shows classification of the target variable plotted for training set and testing set using Support Vector Machine





5. Evaluation:

- The accuracy of the decision tree is 69% where as the accuracy of the support vector machine is 75%
- Below are the error matrix for SVM and Decision Tree respectively

SVM:

[[593 0]

[190 0]]

Decision Tree:

[[636 150]

[154 94]]

There are more errors in the decision tree model than SVM model which is the reason the accuracy of SVM model is higher than the decision tree.

6. Conclusion:

- Since the SVM model has a higher accuracy for the COVID-19 dataset, I would recommend the team to use this model to perform predictive models on similar datasets.
- The results of the analysis will be used by the Research team to group counties based on the immunity level of majority of people of the county.