Machine Learning Engineer Nanodegree
Capstone Project Proposal
Karna Venkata Triveni
Feburary2nd,2019
Proposal:
Classifying Heart Disease Dataset
Domain Background:
History:

Across the World, Dead due to Heart disease is very common. About 610,000 people die of heart disease in the United States every year–that's 1 in every 4 deaths. Heart disease is the leading cause of death for both men and women. More than half of the deaths due to heart disease in 2009 were in men.

Heart disease is considered as one of the top preventable causes of the death in the United States. Some genetic factors can contribute, but the disease is largely attributed to poor life style habits

So,I am going to predict the rate of heart diseases in this project

One recent research paper based on the heart disease reference link: https://www.researchgate.net/publication/328031918_Machine_Learning_Classification_Techniques_for_Heart_Disease_Prediction_A_Review#pf7

Applications:

This project is predicting the heart rate of the dataset and it will helps for the government to take prevention methods i.e making awareness and how to follow the diet control programs etc.It helps for the total analysis of the rate of heart disease persons

Problem Statement:

The main aim of my project is to predict the rate of heart disease.For this I selected the data set compiled from a wide range of sources and made publicly available by the United States Department of Agriculture Economic Research Service (USDA ERS).So,My goal is to predict the rate of heart diseases in U.S.Here I am using the classification models to find the accuracy of each model and select the best model which will have high accuracy to predict the rate of heart

diseases.Here the input parameters are the training data and the output will either 0 or 1i.e having heart heart disease or not

DataSets And Inputs:

This datasets contains 76 attributes, but all published experiments refer to using a subset of 14 of them. The "goal" field refers to the presence of heart disease in the patient. It is integer valued from 0 (no presence) to 4.Here the total samples in the data set is 6278(training data and testing data).Here the target variable is The data type of heart_diseases_mortality_per_100k is an integer, Output is integer value.The data set consists of Nan values for some features.I will remove it(clean the data)

Attributes:
> 1. age
> 2. sex
> 3. chest pain type (4 values)
> 4. resting blood pressure
> 5. serum cholestoral in mg/dl
> 6. fasting blood sugar > 120 mg/dl
> 7. resting electrocardiographic results (values 0,1,2)
> 8. maximum heart rate achieved
> 9. exercise induced angina
> 10. oldpeak = ST depression induced by exercise relative to rest
> 11. the slope of the peak exercise ST segment
> 12. number of major vessels (0-3) colored by flourosopy
> 13. thal: 3 = normal; 6 = fixed defect; 7 = reversable defect

Solution Statement:

Here, I trying to predict the rate of the heart disease for the selected data set.For predicting the rate of heart disease we want to use the different classification models.Then, we will find the accuracy score for each classification model. I explore the data set with opencv and matplotlib.pyplot libraries.in this project.By using visualization helps me to better understand the solutions

Benchmark Model:
This step will be important because compare your final model with some of them and see if it got better, same or worse. Here accuracy score will be compared between the models and select the best one
Evaluation Metrics:

I want to use accuracy score as evaluation metric for prediction of rate of heart disease. Here I am predicting the accuracy score fot the selected models.
In multilabel classification, this function computes subset accuracy: the set of labels predicted for a sample must *exactly* match the corresponding set of labels in y_true.
Here accuracy score which model have the high value it is selected as the best model
Here I choosen the classification algorithms.
We use metrics because:
Training Objective is only a proxy for real world Objective
Metrics help capture a business goal into a quantitative target(not all errors are equal)
Helps organize ML team effect towards the target
Useful to quantify the gap between:

- Desired performance and business(estimate effort initially)
- Desired performance and current performance
- Measure progress over time

Useful for lower level tasks and debugging
Ideally training objective should be the metric,but not always possible.Still metrics are useful and important for evaluation

- Here I have chosen the accuracy score as metric because Accuracy score defines the what fractions of all predictions of train labels to test labels and gives the percentage value
- accuracy_score(*y_true*, *y_pred*, *normalize=True*, *sample_weight=None*) is used to calculate the accuracy score

y_true : 1d array-like, or label indicator array / sparse matrix

Ground truth (correct) labels.

y_pred : 1d array-like, or label indicator array / sparse matrix

Predicted labels, as returned by a classifier.

normalize : bool, optional (default=True)

If False, return the number of correctly classified samples. Otherwise, return the fraction of correctly classified samples.

sample_weight : array-like of shape = [n_samples], optional

Sample weights.

-
- By this score we can say that our model is working correctly for predicting the heart disease or not .If the accuracy score is high that means that our model working properly for predicting the selected problem else not working properly for predicting the selected problem

Project Design:

The project is composed of different steps as follows:

Pre-processing:

First task is to read the dataset and perform visualizations on it to get some insights about the data. After reading the data clean the data i.e. removing unwanted data or replacing null values with some constant values or removing duplicates. Then finding the correlation for each features with the heart disease target variable

After Data Exploration, I want to split the total data into training, validation and testing sets and normalize the data to make it suitable for .Then applying the Classifying models and then predicting the accuracy score to the selected models

First step in training:

First, I want to choose a Benchmark model which will at least gives testing accuracy score around 50 % accuracy score.

Second step in training:

I want to apply classification models of my own and use on the data. I want to apply Support Vector Machine and Logistic Regression model then find the Accuracy score for both the models

Finally, I will declare the model which highest accuracy score on both training and testing data sets concluded as the best model for detecting the rate of heart diseases

ANALYSIS

Data Exploration:

The data set loaded in memory using read_csv()

This datasets contains 76 attributes, but all published experiments refer to using a subset of 14 of them. The "goal" field refers to the presence of heart disease in the patient. It is integer valued from 0 (no presence) to 4.Here the total samples in the data set is 6278(training data and testing data).Here the target variable is The data type of heart_diseases_mortality_per_100k is an integer, Output is integer

value.The data set consists of Nan values for some features.I will remove it(clean the data)

Attributes:
> 1. age
> 2. sex
> 3. chest pain type (4 values)
> 4. resting blood pressure
> 5. serum cholestoral in mg/dl
> 6. fasting blood sugar > 120 mg/dl
> 7. resting electrocardiographic results (values 0,1,2)
> 8. maximum heart rate achieved
> 9. exercise induced angina
> 10. oldpeak = ST depression induced by exercise relative to rest
> 11. the slope of the peak exercise ST segment
> 12. number of major vessels (0-3) colored by flourosopy
> 13. thal: 3 = normal; 6 = fixed defect; 7 = reversable defect

The sample values of the dataset is:

	row_id	area__rucc	area__urban_influence	econ__economic_typology	econ__pct_civilian_labor	econ__pct_unemployment	econ__pct_uninsured_adults	econ
0	0	Metro - Counties in metro areas of fewer than ...	Small-in a metro area with fewer than 1 millio...	Manufacturing-dependent	0.408	0.057	0.254	
1	1	Metro - Counties in metro areas of fewer than ...	Small-in a metro area with fewer than 1 millio...	Mining-dependent	0.556	0.039	0.260	
2	4	Metro - Counties in metro areas of 1 million p...	Large-in a metro area with at least 1 million ...	Nonspecialized	0.541	0.057	0.070	
3	5	Nonmetro - Urban population of 2,500 to 19,999...	Noncore adjacent to a small metro with town of...	Nonspecialized	0.500	0.061	0.203	
4	6	Nonmetro - Urban population of 2,500 to 19,999...	Noncore not adjacent to a metro/micro area and...	Nonspecialized	0.471	0.050	0.225	

5 rows × 34 columns

And features in the data set contains the missing values

```
In [86]: full_NA = full.isnull().sum()
         full_NA = full_NA.drop(full_NA[full_NA == 0].index).sort_values(ascending = False)

         print(full_NA)
```

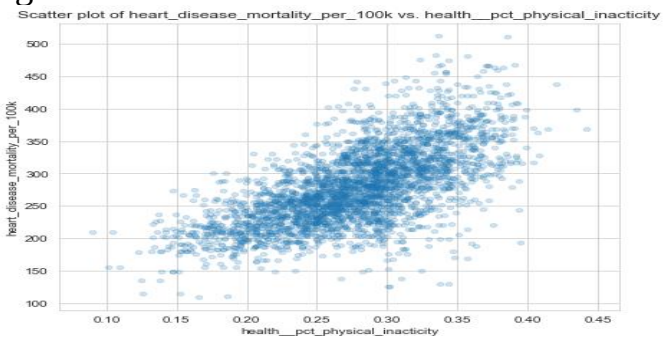
```
health__homicides_per_100k                    3797
health__pct_excessive_drinking                1838
health__pct_adult_smoking                      866
health__motor_vehicle_crash_deaths_per_100k    762
health__pop_per_dentist                        442
health__pop_per_primary_care_physician         414
health__pct_low_birthweight                    329
health__air_pollution_particulate_matter        66
demo__pct_non_hispanic_african_american         10
econ__pct_uninsured_children                    10
demo__pct_female                                10
demo__pct_below_18_years_of_age                 10
demo__pct_aged_65_years_and_older               10
demo__pct_hispanic                              10
health__pct_adult_obesity                       10
demo__pct_non_hispanic_white                    10
demo__pct_american_indian_or_alaskan_native     10
demo__pct_asian                                 10
health__pct_diabetes                            10
health__pct_physical_inacticity                 10
econ__pct_uninsured_adults                      10
dtype: int64
```

The above image defines the missing value features of the dataset

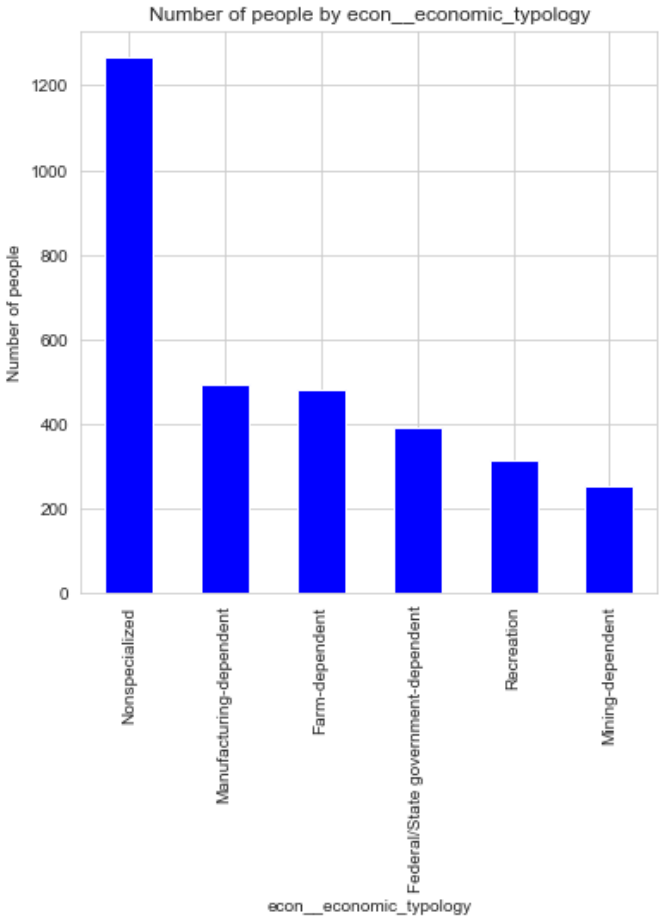
Data Visualization:

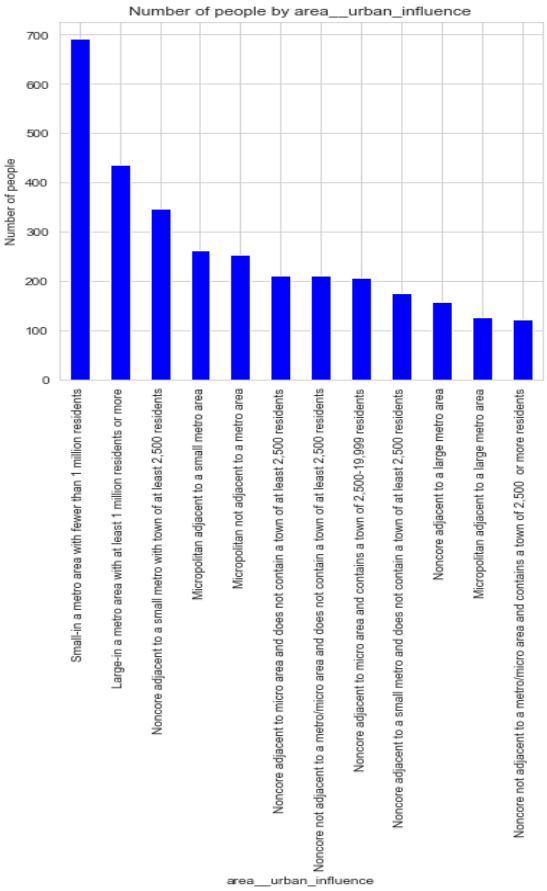
The below image shows the relationship of feature with the heart disease.This is helped for prediciting the value of heart disease for each feature after the data is cleaned.

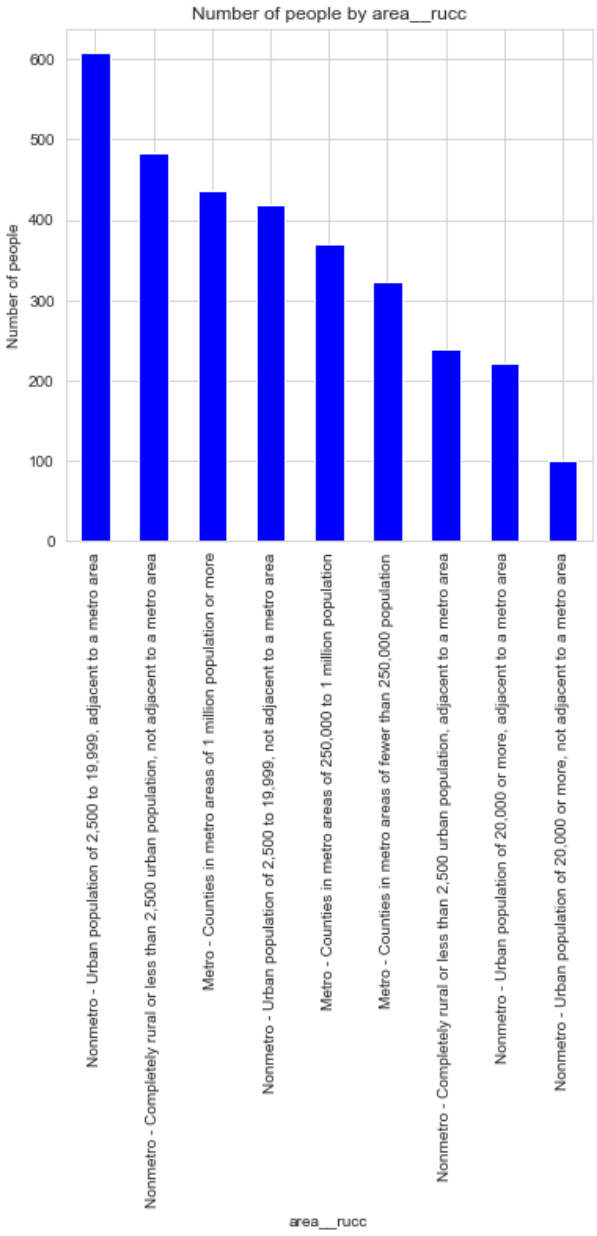
Eg:for first feature is:

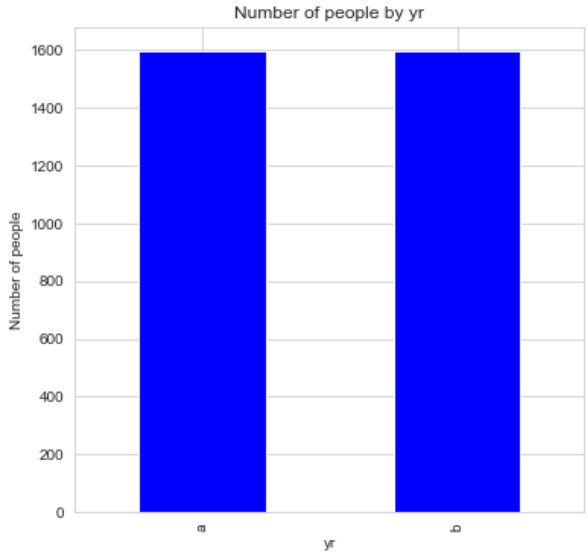


The above visualization helps to find the which feature has the highest categorical value .This is fined after cleaning the data and splitting the data into traning and testing. And finding thecategorical value for each feature


Number of people by econ__economic_typology
Number of people
0
200
400
600
800
1000
1200
Nonspecialized
Manufacturing-dependent
Farm-dependent
Federal/State government-dependent
Recreation
Mining-dependent
econ__economic_typology


Number of people by area__urban_influence
Number of people
0
100
200
300
400
500
600
700
Small-in a metro area with fewer than 1 million residents
Large-in a metro area with at least 1 million residents or more
Noncore adjacent to a small metro with town of at least 2,500 residents
Micropolitan adjacent to a small metro area
Micropolitan not adjacent to a metro area
Noncore adjacent to micro area and does not contain a town of at least 2,500 residents
Noncore not adjacent to a metro/micro area and does not contain a town of at least 2,500 residents
Noncore adjacent to micro area and contains a town of 2,500-19,999 residents
Noncore adjacent to a small metro and does not contain a town of at least 2,500 residents
Noncore adjacent to a large metro area
Micropolitan adjacent to a large metro area
Noncore not adjacent to a metro/micro area and contains a town of 2,500 or more residents
area__urban_influence





Algorithm Techniques:

Two algorithm techniques I am using is:

1.Logistic Regression

2.Support Vector Machine

Logistic Regression:

Logistic regression is the classification counterpart to linear regression. Predictions are mapped to be between 0 and 1 through the logistic function, which means that predictions can be interpreted as class probabilities.

The models themselves are still "linear," so they work well when your classes are linearly separable (i.e. they can be separated by a single decision surface). Logistic regression can also be regularized by penalizing coefficients with a tunable penalty strength.

- **Strengths:** Outputs have a nice probabilistic interpretation, and the algorithm can be regularized to avoid overfitting. Logistic models can be updated easily with new data using stochastic gradient descent.
- **Weaknesses:** Logistic regression tends to underperform when there are multiple or non-linear decision boundaries. They are not flexible enough to naturally capture more complex relationships.

2.**Support Vector Machine:**

Support vector machines (SVM) use a mechanism called kernels, which essentially calculate distance between two observations. The SVM algorithm then finds a decision boundary that maximizes the distance between the closest members of separate classes.

For example, an SVM with a linear kernel is similar to logistic regression. Therefore, in practice, the benefit of SVM's typically comes from using non-linear kernels to model non-linear decision boundaries.

- **Strengths:** SVM's can model non-linear decision boundaries, and there are many kernels to choose from. They are also fairly robust against overfitting, especially in high-dimensional space.
- **Weaknesses:** However, SVM's are memory intensive, trickier to tune due to the importance of picking the right kernel, and don't scale well to larger datasets. Currently in the industry, random forests are usually preferred over SVM's.

Bench Mark:

To create brench mark for this project ,I used the comaprision of accuracy score for the two models and deciding which model is best for predicting the rate of heart diseases.The acuuracy score which has the highest value is the best model.Here I got the accuracy score of 0.8 for the logistic regression model

METHODOLOGY

DATA Preprocessing:

The preprocessing done in the “pre data”notebook consists of the following steps:

Data Cleaning:

- Checking the datasets contains any duplicate values or not: train data consists of 3198 uniques values and test data consist of 3080 unique values

Checking the data sets contains the duplicate values or not

```
In [83]: print(train.shape)
         print(train.row_id.unique().shape)

         (3198, 34)
         (3198,)

In [84]: print(test.shape)
         print(test.row_id.unique().shape)

         (3080, 34)
         (3080,)
```

Content:

- After combining the train data and test the the original dataset consists of 6728 rows
- It consists of 14 attributes.Here the data is cleaned so duplicate values are in this data

- Here the data set contains the missing values for some features.Here I am finding the missing count for each feature

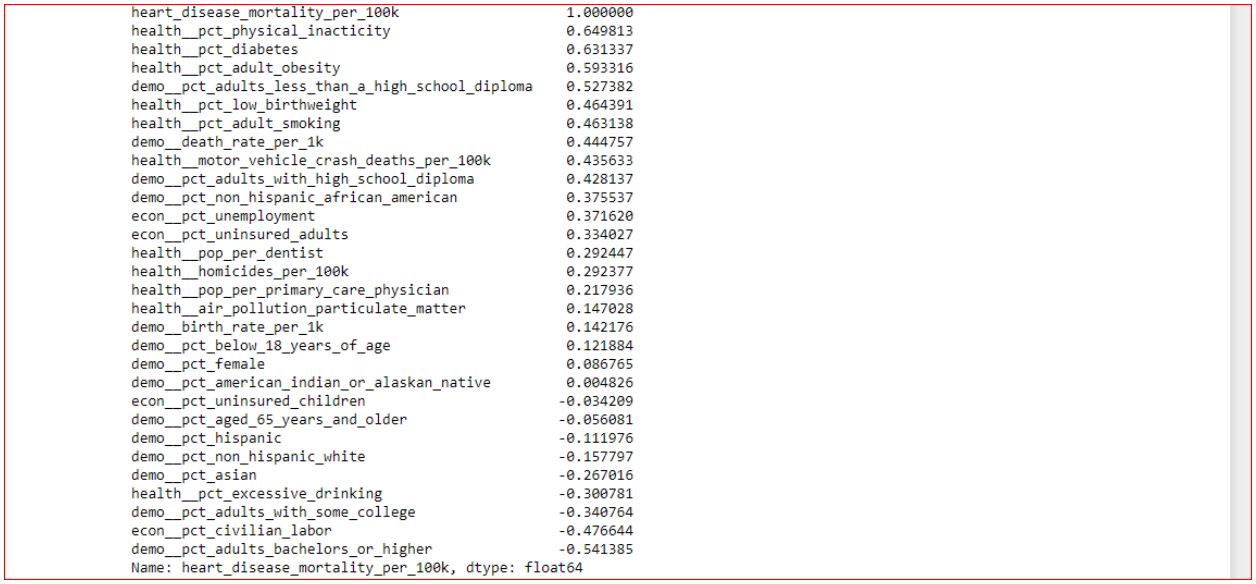
```
In [86]: full_NA = full.isnull().sum()
         full_NA = full_NA.drop(full_NA[full_NA == 0].index).sort_values(ascending = False)

         print(full_NA)

         health__homicides_per_100k                   3797
         health__pct_excessive_drinking               1838
         health__pct_adult_smoking                     866
         health__motor_vehicle_crash_deaths_per_100k   762
         health__pop_per_dentist                       442
         health__pop_per_primary_care_physician        414
         health__pct_low_birthweight                   329
         health__air_pollution_particulate_matter       66
         demo__pct_non_hispanic_african_american        10
         econ__pct_uninsured_children                   10
         demo__pct_female                               10
         demo__pct_below_18_years_of_age                10
         demo__pct_aged_65_years_and_older              10
         demo__pct_hispanic                             10
         health__pct_adult_obesity                      10
         demo__pct_non_hispanic_white                   10
         demo__pct_american_indian_or_alaskan_native    10
         demo__pct_asian                                10
         health__pct_diabetes                           10
         health__pct_physical_inacticity                10
         econ__pct_uninsured_adults                     10
         dtype: int64
```

Now replacing the each feature missing values with their respective feature median value

For further analysis the full data is further divided into the train data and test data

Correlation:

The statistical relationship of two variable is called correlation.If the coorelation is positive it has strong relation ship both move in one direction only and if the negative value is their with increasing of variable value the other variable decreases.The correlation value of each feature with the heart disease is:



```
heart_disease_mortality_per_100k                1.000000
health__pct_physical_inacticity                 0.649813
health__pct_diabetes                            0.631337
health__pct_adult_obesity                       0.593316
demo__pct_adults_less_than_a_high_school_diploma    0.527382
health__pct_low_birthweight                     0.464391
health__pct_adult_smoking                       0.463138
demo__death_rate_per_1k                         0.444757
health__motor_vehicle_crash_deaths_per_100k     0.435633
demo__pct_adults_with_high_school_diploma       0.428137
demo__pct_non_hispanic_african_american         0.375537
econ__pct_unemployment                          0.371620
econ__pct_uninsured_adults                      0.334027
health__pop_per_dentist                         0.292447
health__homicides_per_100k                      0.292377
health__pop_per_primary_care_physician          0.217936
health__air_pollution_particulate_matter        0.147028
demo__birth_rate_per_1k                         0.142176
demo__pct_below_18_years_of_age                 0.121884
demo__pct_female                                0.086765
demo__pct_american_indian_or_alaskan_native     0.004826
econ__pct_uninsured_children                   -0.034209
demo__pct_aged_65_years_and_older              -0.056081
demo__pct_hispanic                             -0.111976
demo__pct_non_hispanic_white                   -0.157797
demo__pct_asian                                -0.267016
health__pct_excessive_drinking                 -0.300781
demo__pct_adults_with_some_college             -0.340764
econ__pct_civilian_labor                       -0.476644
demo__pct_adults_bachelors_or_higher           -0.541385
Name: heart_disease_mortality_per_100k, dtype: float64
```

Now finding the relationship between the no.of categorical values for each feature.

IMPELMENTATION:

The Implementation project is divided into two main stages:

1.The classifier training stage

2.The classification model development stage

Classifier training Stage:(Described in the jupyter Notebook)

The classifier was trained on the preprocessed data.Steps followed here is:

- Load the training and testing data into the memory and preprocess (means cleaning the data)Here the data is preprocessed by finding the unique values and finding the missing values then replacing the missing values with their respective feature median value
- Using displot() function for defining the visualization of distribution of target label
- Finding the correlation the every features with labels to know which feature has the strongest relationship with the target label
- Now split the data into training and test for further implemention
- By using StandardScaler() standardizing the train and test data

Model Development Stage:

- Here we want to define which models we are using for predicting the rate of heart disease
- Then fit the train and test to the selected models

- Then finding the accuracy score for the selected models(based on this we decide which model is best for predicting the rate of the heart disease)

Complications faced here:

- cleaning and Spliting the data for testing and training
- Selecting the tuning parameters for the model to get the best accuracy score

Refinement:

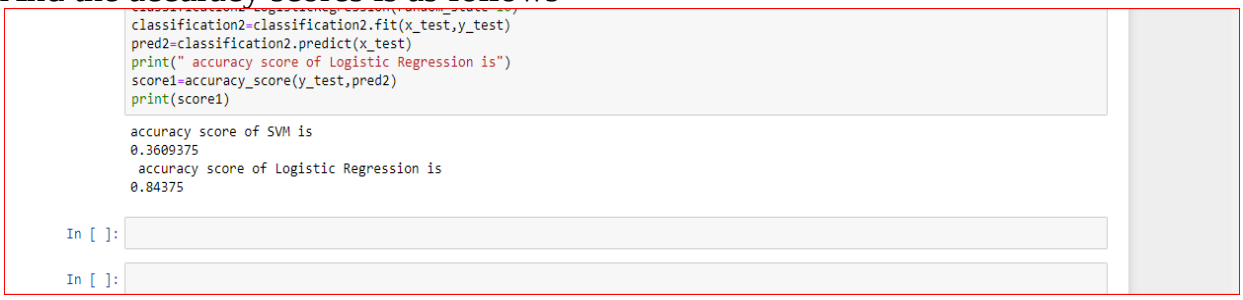
As mentioned in the benchmark section, comparing the accuracy score of the two models

Here for the improving the results I used the hyper paremter is random_state()

If you don't specify the random_state in your code, then every time you run(execute) your code a new random value is generated and the train and test datasets would have different values each time.

However, if a fixed value is assigned like random_state =10 then no matter how many times you execute your code the result would be the same .i.e, same values in train and test datasets.

And the accuracy scores is as follows



The accuracy score I got for the logistic regression is 91%

The accuracy score I got for the SVM is 36.09%

Here the tuned parameters for making the model best in logistic regression is C and max_iters

- C : float, optional (default=1.0) Inverse of regularization strength; must be a positive float. Like in support vector machines, smaller values specify stronger regularization.

- Regularization is applying a penalty to increasing the magnitude of parameter values in order to reduce overfitting. When you train a model such

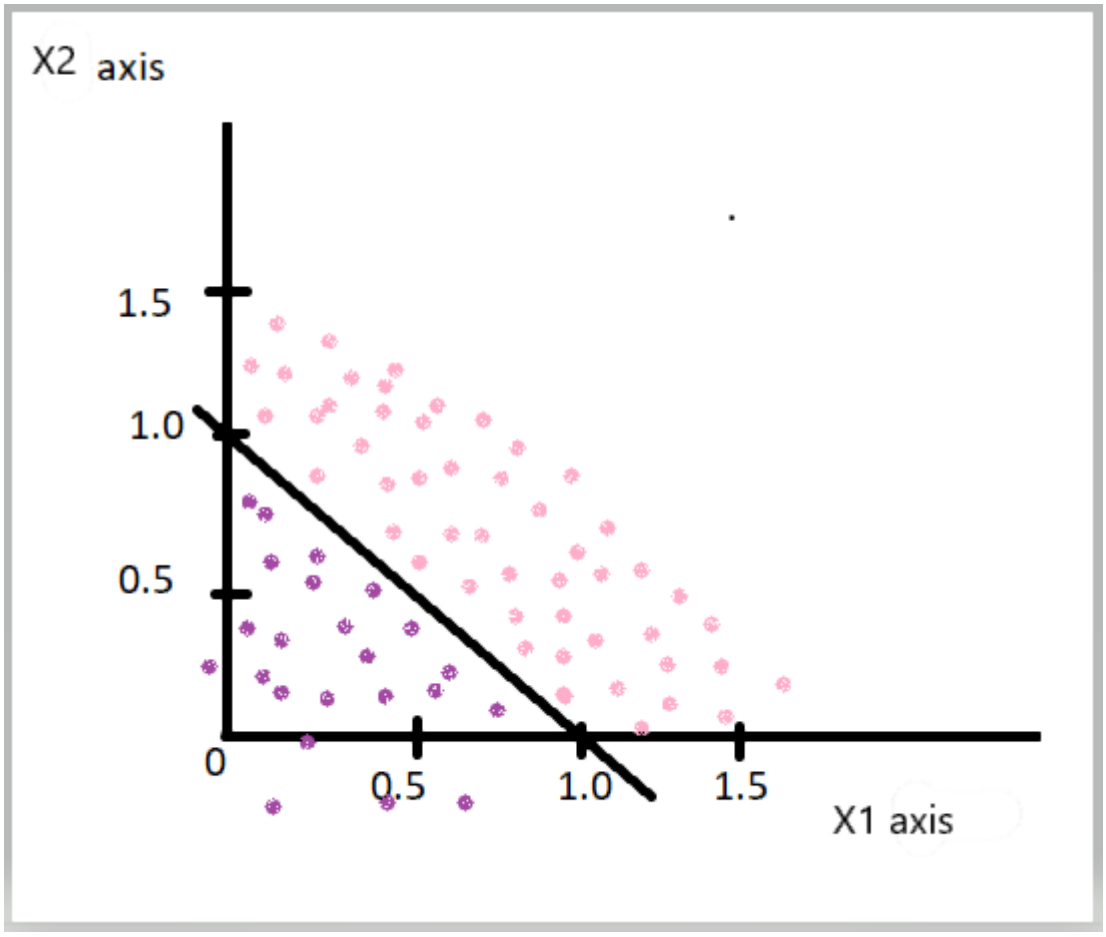
as a logistic regression model, you are choosing parameters that give you the best fit to the data. This means minimizing the error between what the model predicts for your dependent variable given your data compared to what your dependent variable

- Max_tiers: Useful only for the newton-cg, sag and lbfgs solvers. Maximum number of iterations taken for the solvers to converge.
- Here before using,I used the parameter penalty='l1' then I got the accuracy of 66 and then after using the C=2.0 I got the accuracy of 91
- I used it the C parameter to minimize the errors and overfitting

Results

Model Evaluation and Validation:

I used accuracy score as evaluation metric for prediction of rate of heart disease.
Here I am predicting the accuracy score for the selected models.
In multilabel classification, this function computes subset accuracy: the set of labels predicted for a sample must *exactly* match the corresponding set of labels in y_true.
Here accuracy score which model have the high value it is selected as the best model
So,Based on the highest accuracy score which model is I get is the best model for my project.
The accuracy score I got for the logistic regression is 91%
The accuracy score I got for the SVM is 36.09% and the best model for this project is logistic regression because it has the highest accuracy value
And the architecture of the logistic regression is example:



This line is known as the **decision boundary** because it separates the regions that are used to predict outcomes 1 and 0.

When the prediction is done to find the probability of an outcome as 1, the region to the right of the line, which is shaded in **pink** is considered. On the other hand, when we need to find values for the outcome of 0, we consider the region below the line, which is shaded in **purple**. In general, after the values for theta is found, there is no need to plot this graph, because the values for theta itself helps us define the decision boundary

```
In [44]: classification1 = SVC(kernel="rbf",random_state=10)
         classififcation1=classification1.fit(x_test, y_test)
         pred1=classification1.predict(x_test)
         print("accuracy score of SVM is")
         score=metrics.accuracy_score(y_test,pred1)
         print(score)
         classification2=LogisticRegression(random_state=10,max_iter=10,C=2.0)
         classification2=classification2.fit(x_test,y_test)
         pred2=classification2.predict(x_test)
         print(" accuracy score of Logistic Regression is")
         score1=metrics.accuracy_score(y_test,pred2)
         print(score1)

         accuracy score of SVM is
         0.3609375
          accuracy score of Logistic Regression is
         0.9109375
```

The above defines the accuracy score of the optimized model here used the tuning parameters for best score

The accuracy score of the unoptimised model is 34% for SVM and 84% for logistic regression

Here I am using the C parameter in the logistic regression.By using this paramaeter the model minimize the overfitting and reduces the errors.And I can said that my models is helpful for predicting the rate of the heart disease for the same features od datasets.And here I am cleaning the data and replacing the null values with the median values and then fitting the training and testing data into the model.SO I can believe that my project works correctly for predict the value of heart disease rate

JUSTIFICATION:

Here by using SVM I got the accuracy score is 36.09%
And by using Logistic Regression I got the accuracy score is 91%
As mentioned in the bench mark By comparing the two models accuracy score ,the Logistic Regression has the high accuracy score than the SVM So,Logistic Regression is the best model for predicting the rate of heart diseases

Conclusion

Free-Form Visualization:

One thing I see quite often is a visual examination of the feature importance metrics, to investigate what information in the dataset matters. But again, there are many options here, so just get creative.For example:

Suppose a logistic regression model is used to predict whether an online shopper will purchase a product (outcome: purchase), after he clicked a set of online adverts (predictors: Ad1, Ad2, and Ad3).

The outcome is a binary variable: 1 (purchased) or 0 (not purcahsed). The predictors are also binary variables: 1 (clicked) or 0 (not clicked). So all variables are on the same scale.

If the resulting coefficients of Ad1, Ad2, and Ad3 are 0.1, 0.2, and 03, we can conclude that Ad3 is more important than Ad2, and Ad2 is more important than Ad1. Furthermore, since all variables are on the same scale, the standardized and un-standardized coefficients should be same, and we can further conclude that Ad2 is twice important than Ad1 in terms of its influence on the logit (log-odds) level.

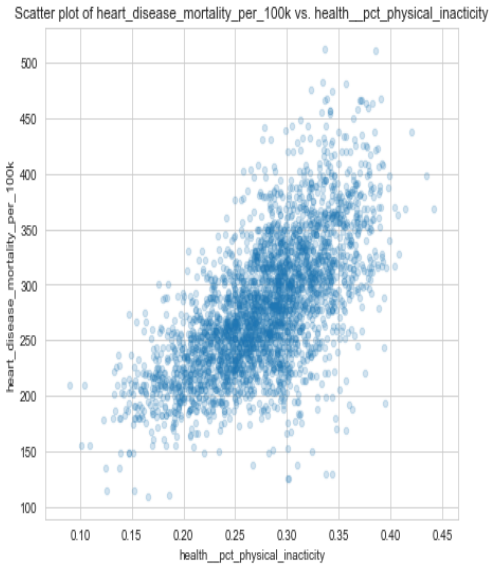
Here

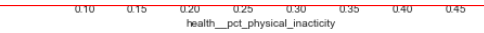
The feature importance is having the heart disease are not is in the form:

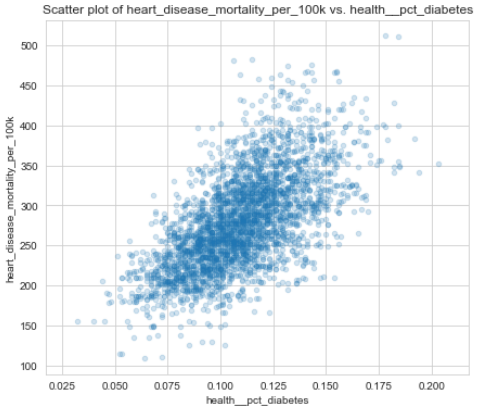
```
heart_disease_mortality_per_100k                  1.000000
health__pct_physical_inacticity                   0.649813
health__pct_diabetes                              0.631337
health__pct_adult_obesity                         0.593316
demo__pct_adults_less_than_a_high_school_diploma  0.527382
health__pct_low_birthweight                       0.464391
health__pct_adult_smoking                         0.463138
demo__death_rate_per_1k                           0.444757
health__motor_vehicle_crash_deaths_per_100k       0.435633
demo__pct_adults_with_high_school_diploma         0.428137
demo__pct_non_hispanic_african_american           0.375537
econ__pct_unemployment                            0.371620
econ__pct_uninsured_adults                        0.334027
health__pop_per_dentist                           0.292447
health__homicides_per_100k                        0.292377
health__pop_per_primary_care_physician            0.217936
health__air_pollution_particulate_matter          0.147028
demo__birth_rate_per_1k                           0.142176
demo__pct_below_18_years_of_age                   0.121884
demo__pct_female                                  0.086765
demo__pct_american_indian_or_alaskan_native       0.004826
econ__pct_uninsured_children                     -0.034209
demo__pct_aged_65_years_and_older                -0.056081
demo__pct_hispanic                               -0.111976
demo__pct_non_hispanic_white                     -0.157797
demo__pct_asian                                  -0.267016
health__pct_excessive_drinking                   -0.300781
demo__pct_adults_with_some_college               -0.340764
econ__pct_civilian_labor                         -0.476644
demo__pct_adults_bachelors_or_higher             -0.541385
Name: heart_disease_mortality_per_100k, dtype: float64
```

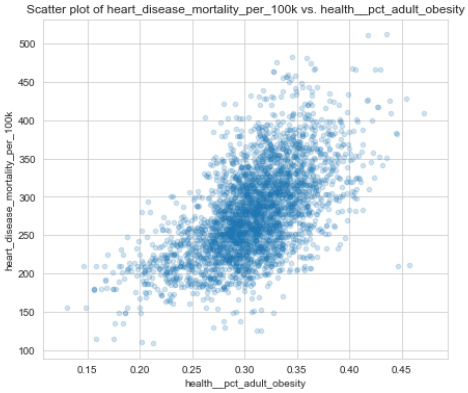
The relationship of each feature with the heart disease target label as follows:

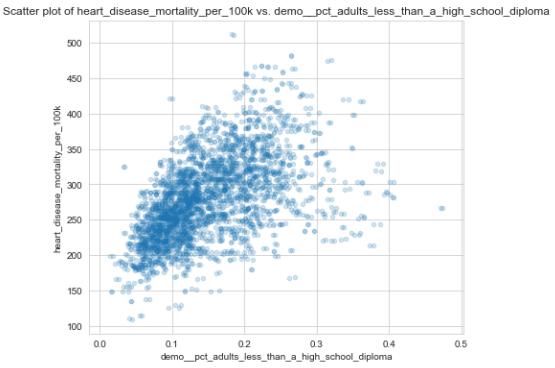
```
plot_scatter_t(train, num_cols, alpha = 0.2)
```

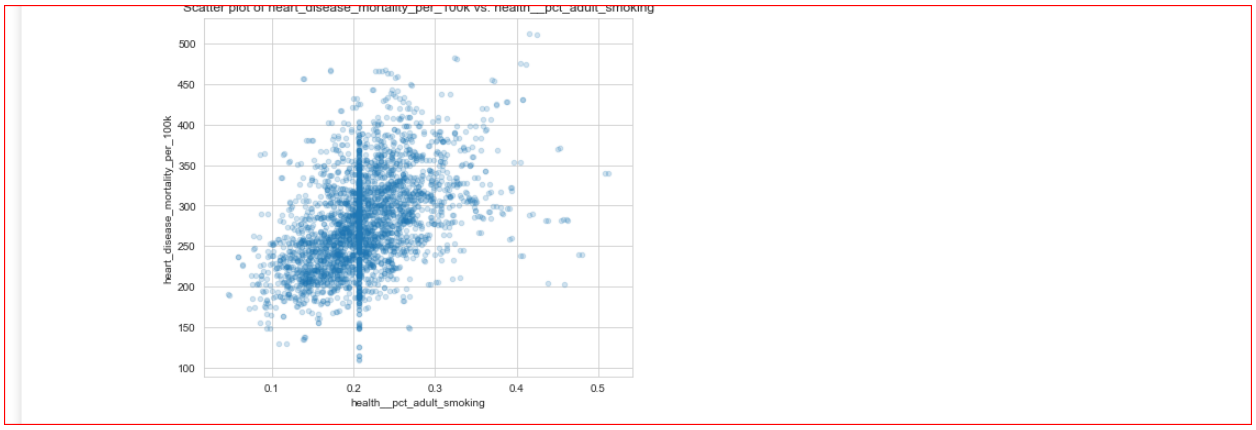



0.10
0.15
0.20
0.25
0.30
0.35
0.40
0.45
health__pct_physical_inactivity


Scatter plot of heart_disease_mortality_per_100k vs. health__pct_diabetes
heart_disease_mortality_per_100k
500
450
400
350
300
250
200
150
100
0.025
0.050
0.075
0.100
0.125
0.150
0.175
0.200
health__pct_diabetes


Scatter plot of heart_disease_mortality_per_100k vs. health__pct_adult_obesity
heart_disease_mortality_per_100k
500
450
400
350
300
250
200
150
100
0.15
0.20
0.25
0.30
0.35
0.40
0.45
health__pct_adult_obesity


Scatter plot of heart_disease_mortality_per_100k vs. demo__pct_adults_less_than_a_high_school_diploma
heart_disease_mortality_per_100k
500
450
400
350
300
250
200
150
100
0.0
0.1
0.2
0.3
0.4
0.5
demo__pct_adults_less_than_a_high_school_diploma



From correlation and from scatter plots we can say that the paitent who has the health_pct_physcial disease may have the heart because because this feature has the highest correlation with the target label

Reflection:

The process used for this project is summarized in the following steps:

1.First thing I have learned about data retrieval processes. When I am doing research about retrieval process, I have come across a lot of surprising methods try. Out of all I decided to use load files method in cv2

2.I have also learn how to use kaggle kernals and how to commit it and reproduce my work.

3. When I started loading data, I used the useful method is pd.read_csv( )

4. Then I used my skills on representation of trained data with heart disease, and also category wise in both training and testing data by using Matplotlib library.

5. I am also learn how to handle the corrupted data in dataset

6. Then I have learned about using density function how the features are distributed is defined.
7.Then I learned that How we divide the data into training and testing using traint_test_split method. And to find the correlation between the features and the heart disease
8.I learned how to standardize the data using standardscaler
9.Here comes the heart of project, creating a SVM Model and Logistic Regression Model, fitting it to training data and testing on test data and I learn the finding the accuracy score .
I am learned the intreseted topic is visualization of the distribution function for every feature
I found the difficulty is the classifier trained using the data with good results

Improvement:
One thing, that can be improved from my models is we can use corr( ) method for finding the correlation and train_test_split for splitting data while fitting model. And also we can use navie bayes but the accuracy score is low.

Final Note:
But, when I trying to improve my two models, I have done two things. Instead of train_test_split method, I have used Kfold split method, soon I came to observe that, both models are taking infinite amount of time to fit training data, so to speak I kind of drop the idea of using it. When I started this project I used Navie Bayes and SVM model with on the training and testing data I got the accuracy score same for both models and accuracy score is less than 50% for both models So I quit the idea.

So here based on the accuracy score of two models i.e Logistic Regression and SVM
I would preferred the Logistic Regression model because accuracy is high for this model compare to SVM
one of the benefits of logistic regression in that it calculates a probability for each prediction.In order to calculate these probabilities, the model looks at how each input variable correlates with the target variable across a number of samples.Trainig and testing time is also low compare to SVM
So,I considered Logistic Regression is my model