Performance analysis of Machine Learning Models trained on life threatening Diseases

Waqas Kureshy Neeharika Yeluri Jasmine Wang

06 December 2021

Abstract

Humans are plagued by different diseases globally, of these diseases CVD (Cardiovascular diseases), Diabetes and Breast Cancer are known to have a prolonged and detrimental effect. Our task was to work on datasets related to the diseases mentioned above by employing different Machine Learning approaches, consolidate results and provide a comparison to the reader about the models used. For this exercise we used Sequential models, Decision tree Classifiers, Support Vector Machines, LogisticRegression, RandomForestClassifier, and KNeighborsClassifier. Techniques used and comparison for all the models made for each dataset in terms of metrics are presented in this paper.

Introduction

Cardiovascular diseases, diabetes, and breast cancer are top 3 causes of death globally, taking millions of lives each year, which accounts for 50% of all deaths worldwide. It has long been the struggle of Researchers, Scientists and Doctors to come up with comprehensive statistical models to conclude positively the patient's medical ailments. This approach carries benefits for the Medical industry workers as well as the patient. Employing such models that have the ability to classify a patient's health could serve as an early warning system, this enables the patient to get timely medical treatment and advice. This approach as well as proving valuable to a person's health can also potentially save time and effort in the process of Medical diagnosis. Our objective for this project was to take three datasets pertaining to Cardiovascular diseases, Diabetes, as well as Breast Cancer, conduct initial exploratory analysis of the data, clean and preprocess the data and then to use this data to make different Machine Learning models, compile metrics and present an analysis of the performance of each model to the reader.

For this exercise we worked on three different datasets, feature and statistics of each dataset are elaborated as:

Cardiovascular Disease Classification Dataset

Table 1: CVD dataset features

Feature	Data type	Value
Age	Integer	Age of the patient [years]
Sex	Integer	Sex of the patient [1: Male, 0: Female]
Chest Pain Type	Integer	[1: Typical Angina, 2: Atypical Angina, 3: Non-Anginal Pain,4: Asymptomatic]
Resting BP	Integer	Resting blood pressure [mm Hg]
Cholesterol	Integer	Serum cholesterol [mm/dl]
Fasting Blood Sugar	Integer	[1: if FastingBS $>$ 120 mg/dl, 0: otherwise]
Resting ECG	Integer	Resting electrocardiogram results [0: Normal, 1: having ST-T wave abnormality (T wave inversions and/or ST elevation or depression of $>$ 0.05 mV), 2: showing probable or definite left ventricular hypertrophy by Estes criteria]
Max Heart Rate	Integer	Maximum heart rate achieved [Numeric value between 60 and 202]
Exercise Angina	Integer	Exercise-induced angina [1: Yes, 0: No]
Old peak	Float	Oldpeak = ST [Numeric value measured in depression]
ST Slope	Integer	The slope of the peak exercise ST segment [1: upsloping, 2: flat, 3: downsloping]
Target	Integer	Output class [1: heart disease, 0: Normal]

The Correlation of the dataset attributes with respect to the target variable i.e the possibility of having a CVD (Cardiovascular Disease) are depicted in the graph number below.

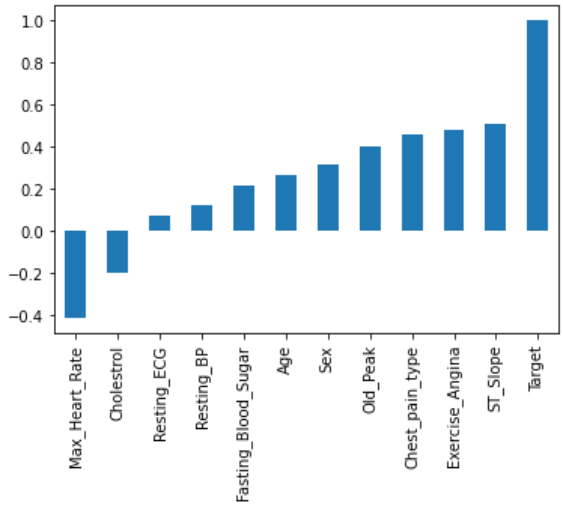

Figure 1: Correlation Graph with Target variable

From this analysis it is evident that 9 different attributes/features have a positive correlation with the target variable.

A heatmap showing the correlation of the features with each other are shown in fig-2 , this image shows graphically how each feature correlates with each other feature. This analysis shows that the following features are positively correlated with the target variable [Age , Sex , Chest pain Type , Resting Blood Pressure, Fasting Blood Sugar , Resting ECG , Exercise Angina , Old Peak , ST_Slope]

With the attribute ST Slope (The ST segment encompasses the region between the end of ventricular depolarization and beginning of ventricular repolarization on the ECG) showing the highest correlation.

The CVD dataset is composed of 1190 cases which is professionally compiled, by merging datasets from different origins and compilations presented in [1] .The distribution of the dataset with respect to the target variable as shown in fig-3 shows that the dataset is balanced and is adequate for modeling.

A graph plot shown in fig-4 shows the distribution of the dataset with respect to the target variable differentiated on the basis of age (0: for no CVD disease and 1: for having a positive identification of CVD).

The following fig-5 shows a plot of various types of chest pain encountered in the dataset separated by Gender, where 0 represents female cases and 1 represents male cases.

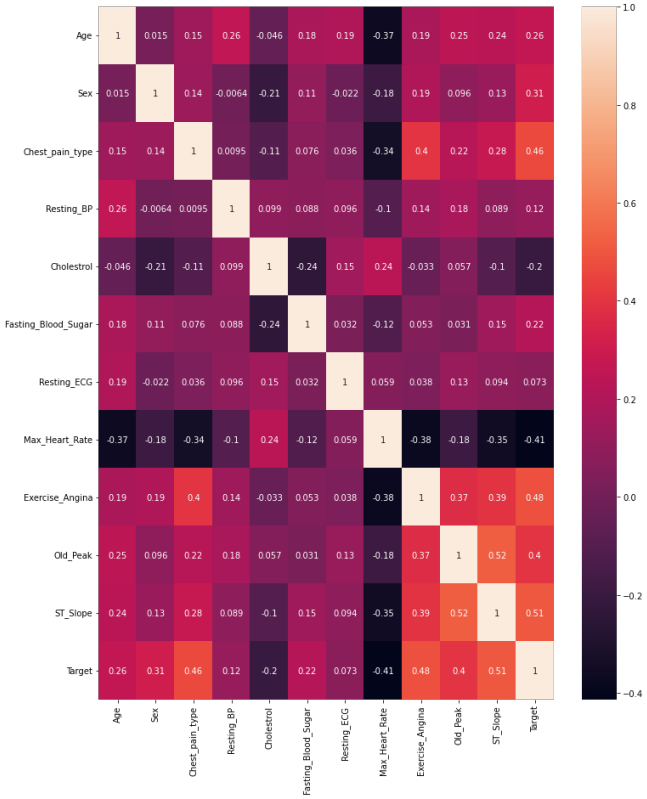

Figure 2: Heatmap of correlation

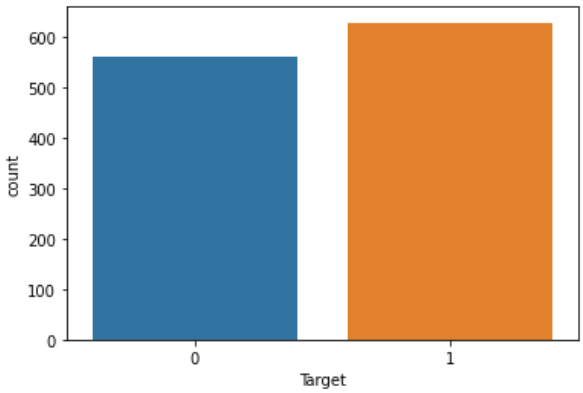

Figure 3: Distribution w.r.t the target

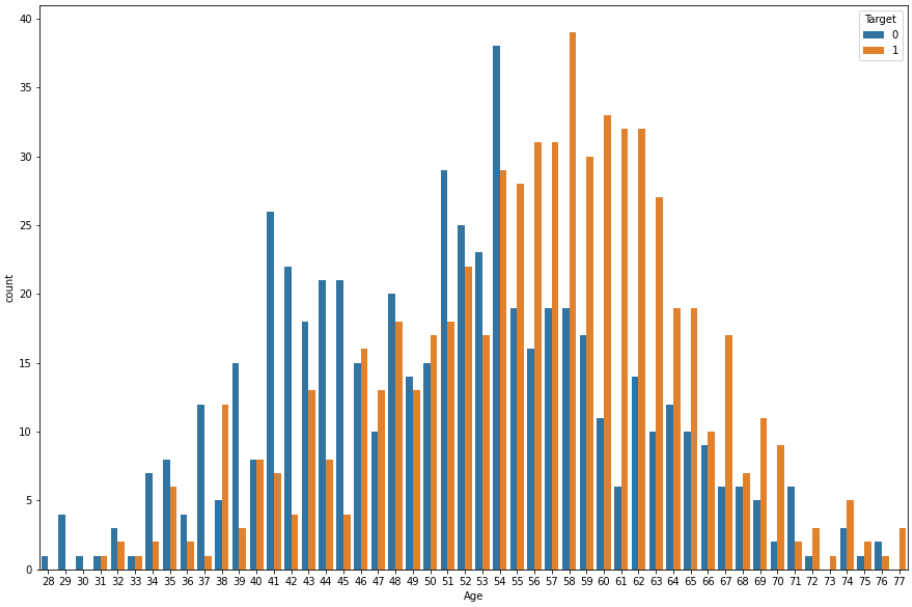

Figure 4: Distribution w.r.t target differentiated by age

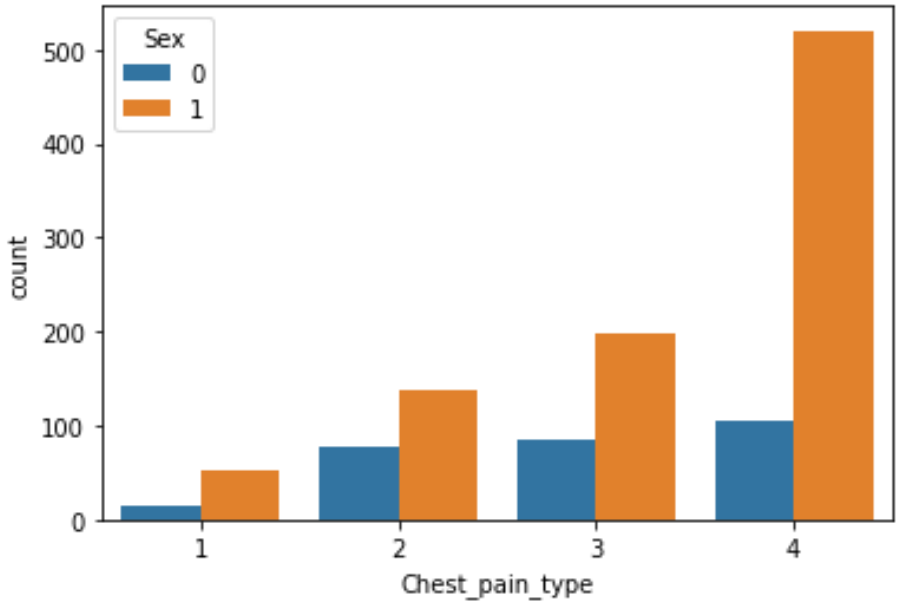

Figure 5: Plot of various types of chest pain encountered in the dataset separated by Gender

The following fig-6 shows a plot of the distribution of Age across the Sex feature, it can be observed that males tend to contract Heart Diseases earlier than females. In the figure 0 represents female cases and 1 represents male cases.

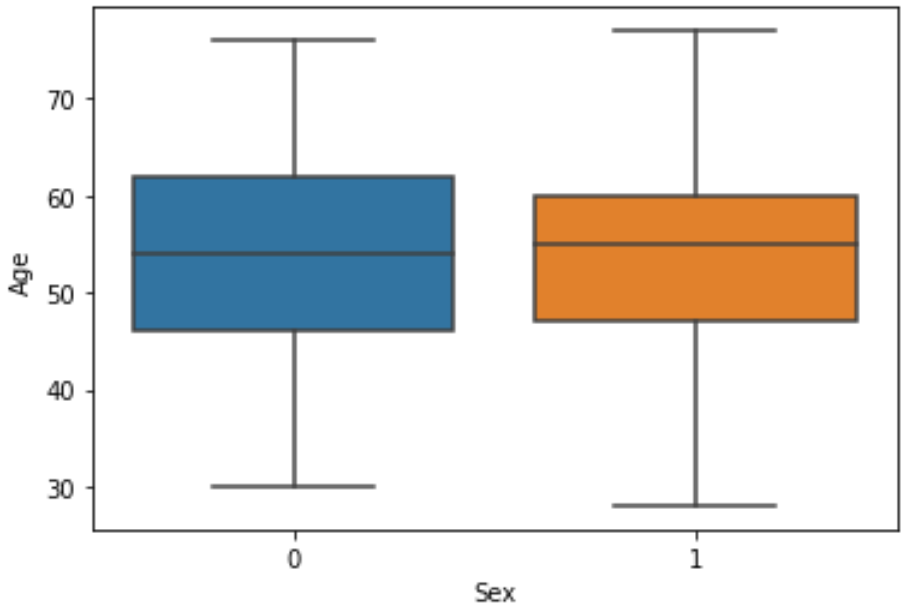

Figure 6: Distribution of Age across the Sex feature

The plot in fig-7 shows interesting information about the trends of heart rate of people having CVD and not having CVD separated by Gender. It can be observed that Males who have suffered from CVD tend to have a lower Heart Beat rate than their female counterparts who also have CVDs.

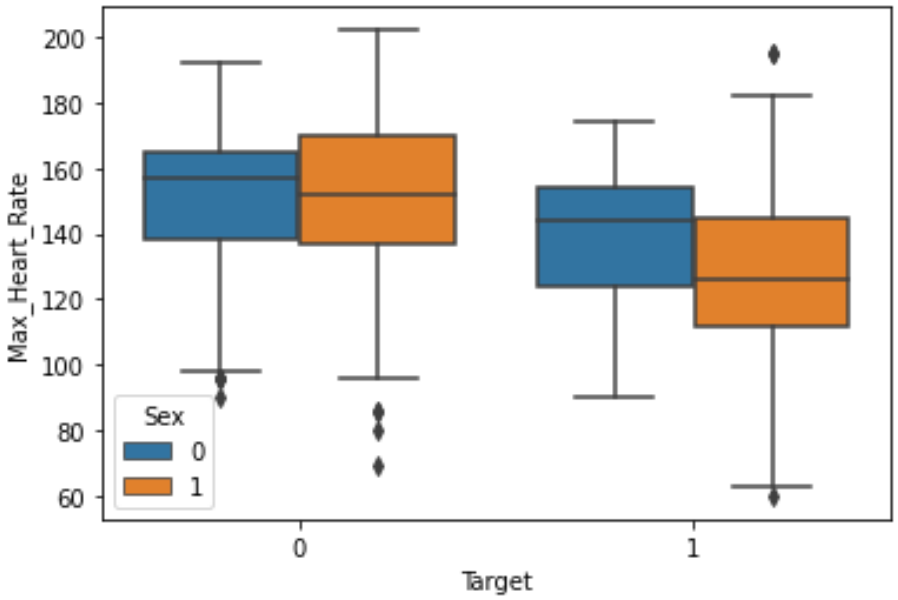

Figure 7: Trends of heart rate of people

Diabetes

Diabetes is a common chronic disease. Prediction of diabetes at an early stage can lead to improved treatment. The dataset is originally from the National Institute of Diabetes and Digestive and Kidney Diseases. The objective is to predict based on diagnostic measurements whether a patient has diabetes. Several constraints were placed on the selection of these instances from a larger database. In particular, all patients here are females at least 21 years old of Pima Indian heritage. This dataset has 9 attributes which are listed below:

- Pregnancies : Number of times pregnant
- Glucose : Plasma glucose concentration a 2 hours in an oral glucose tolerance test
- BloodPressure : Diastolic blood pressure (mm Hg)
- SkinThickness : Triceps skin fold thickness (mm)
- Insulin : 2-Hour serum insulin (mu U/ml)
- BMI : Body mass index (weight in kg/(height in m)ˆ2)
- DiabetesPedigreeFunction : Diabetes pedigree function
- Age Age (years)
- Outcome : Class variable (0: Tested Negative or 1: Tested Positive)

There are only numerical variables in this dataset. 768 observations, and 9 variables(1 dependent) are available

Imputation: Under normal circumstances, it seems that there are no missing values in the data set, but there may be missing values hidden in the data of the variables here. Then we examined the missing values of each variable according to the target variable. So we decided to apply different methods in order to fill na values according to the state of each variable because of the range differences of flag counts. For the variables {Glucose, Blood Pressure and BMI} we filled the missing values with median and for the remaining two variables {Insulin and SkinThickness} we filled them with the K Nearest Neighbours.

New Feature Interaction:

By converting these numerical variables into categorical we get a clear picture of the analysis of data. - Glucose - Women with hyperglycemia will have a higher incidence of diabetes on average the "Outcome". - Age - Middle-aged women will have a higher incidence of diabetes on average the "Outcome". - BMI - Morbidly obese women will have a higher incidence of diabetes on average the "Outcome". - Blood Pressure - Women with high blood pressure will have a higher incidence of diabetes on average the "Outcome". - Insulin - Women with abnormal insulin will have a higher incidence of diabetes on average the "Outcome." - Pregnancies - Women with a very high pregnancy rate will have a higher incidence of diabetes on average.

All the attributes have a positive correlation with respect to the target variable Outcome which is depicted in fig-8 below with Glucose having the highest correlation and BloodPressure the lowest.

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
Pregnancies	1.000000	0.129459	0.141282	-0.081672	-0.073535	0.017683	-0.033523	0.544341	0.221898
Glucose	0.129459	1.000000	0.152590	0.057328	0.331357	0.221071	0.137337	0.263514	0.466581
BloodPressure	0.141282	0.152590	1.000000	0.207371	0.088933	0.281805	0.041265	0.239528	0.065068
SkinThickness	-0.081672	0.057328	0.207371	1.000000	0.436783	0.392573	0.183928	-0.113970	0.074752
Insulin	-0.073535	0.331357	0.088933	0.436783	1.000000	0.197859	0.185071	-0.042163	0.130548
BMI	0.017683	0.221071	0.281805	0.392573	0.197859	1.000000	0.140647	0.036242	0.292695
DiabetesPedigreeFunction	-0.033523	0.137337	0.041265	0.183928	0.185071	0.140647	1.000000	0.033561	0.173844
Age	0.544341	0.263514	0.239528	-0.113970	-0.042163	0.036242	0.033561	1.000000	0.238356
Outcome	0.221898	0.466581	0.065068	0.074752	0.130548	0.292695	0.173844	0.238356	1.000000

Figure 8: Diabetes dataset correlation

In the case of diabetes, especially the Glucose, BMI and Age variables of women are an important factor. The rate of diabetes may be higher in middle-aged women aged 45-65 years.

The following is the heat map which shows the correlation of the features with each other. This image shows graphically how each feature correlates with each other feature. This analysis shows that the following features are positively correlated

with the target variable [Pregnancies, Glucose, BloodPressure, SkinThickness, Insulin, BMI, DiabetesPedigreeFunction and Age].

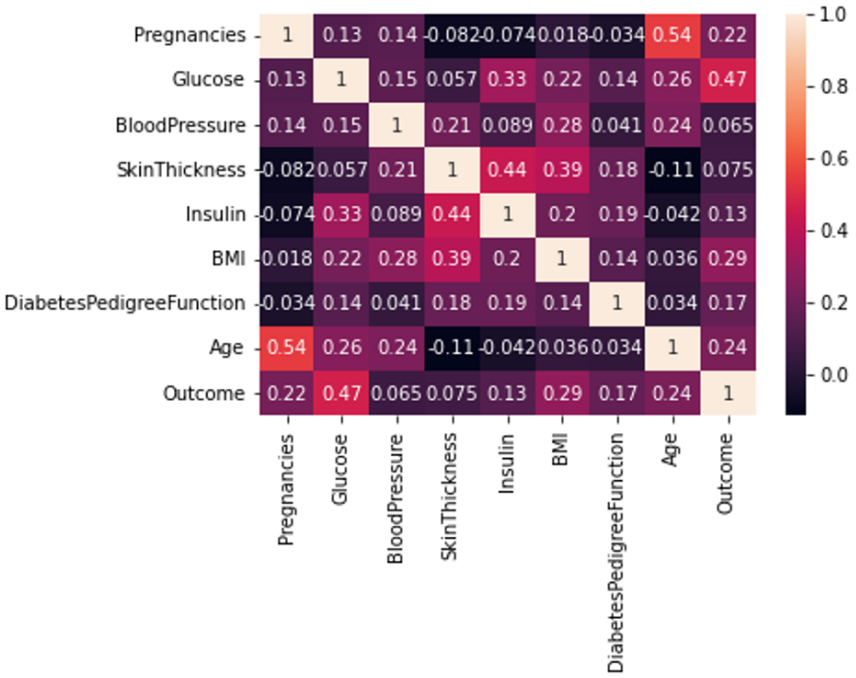

Figure 9: Heatmap correlation

The following pie chart shows the number of people diagnosed and non diagnosed with diabetes.

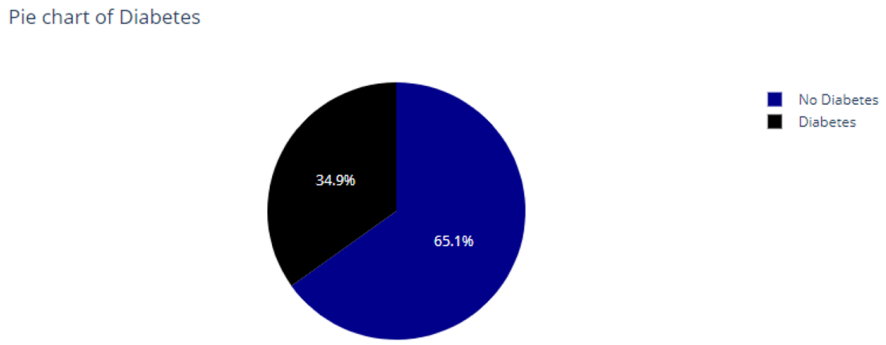

Figure 10: Distribution w.r.t target

Below are the distributions plots of each feature

- Glucose Distribution Plot:

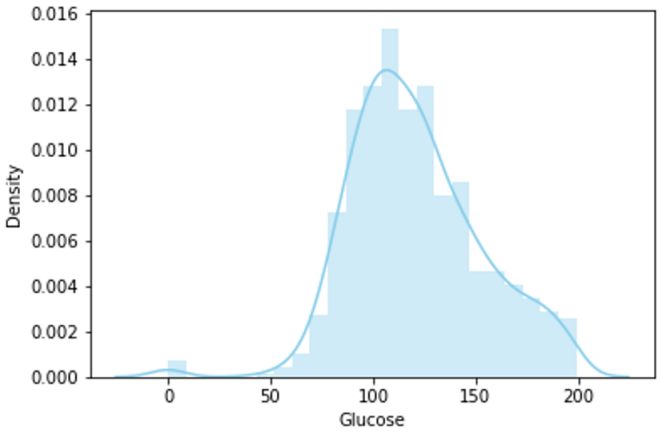

Figure 11: Glucose Distribution Plot

- BloodPressure Distribution Plot:

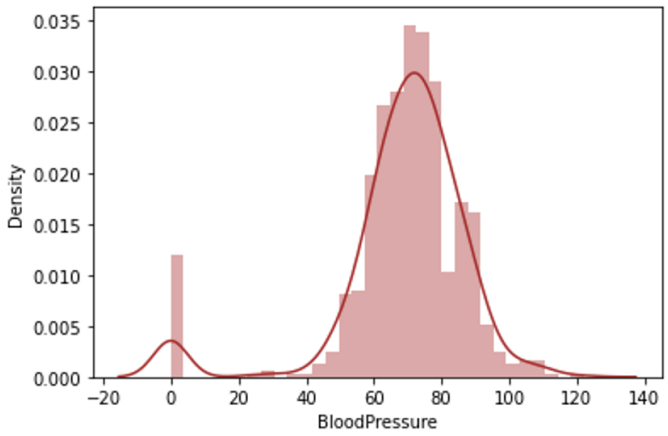

Figure 12: BloodPressure Distribution Plot

- Insulin Distribution Plot:

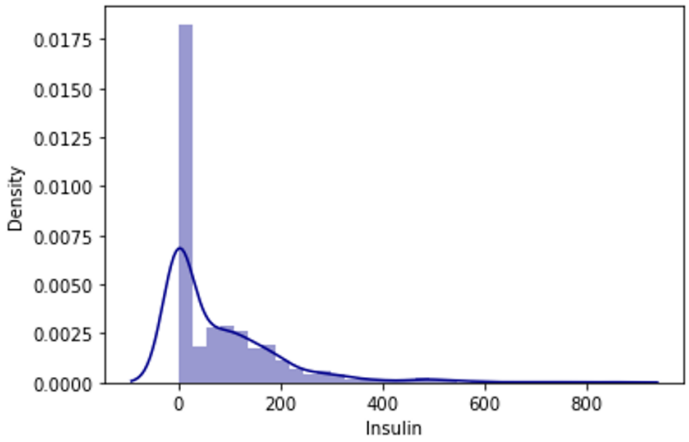

Figure 13: Insulin Distribution Plot

- Pregnancies Histogram Plot:

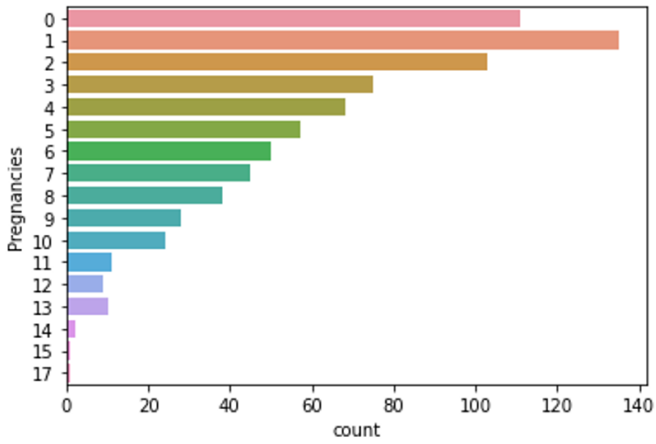

Figure 14: Pregnancies Histogram Plot

- Age Histogram Plot:

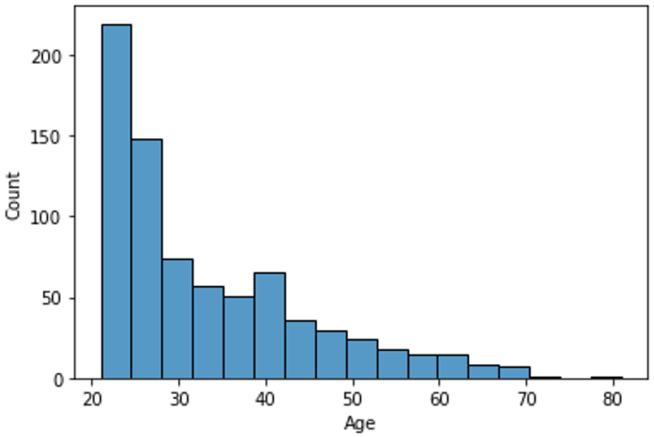

Figure 15: Age Histogram Plot

- Pair plot:

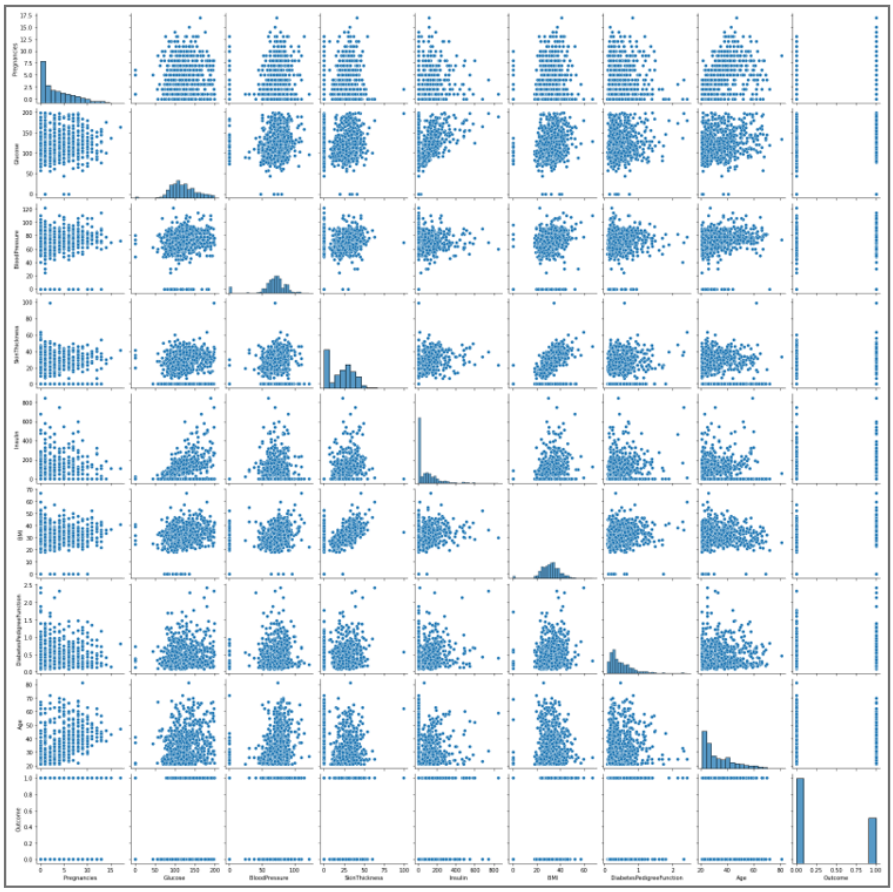

Figure 16: Pair plot of data

Methods

Each dataset used, presents itself essentially as a classification task. For this reason we employed different classification algorithms. The design methodology and Algorithm used for each dataset is explained further in the following sections.

CVD (Cardiovascular Disease Classification using ANNs)

For this task a Artificial Neural Network model was trained, to classify patient cases as having the Target (CVD) diseases or not. The methodology is summarized as:

- 11 data columns were selected as features and the 'Target' column was selected as our label.
- The set was distributed into training and testing data and then scaled using a Minmax scaler.
- A sequential model was created using the Tensorflow Keras API which comprised 3 hidden layers and 1 output layer.
- The three hidden layers used the SELU ( ) activation function, this function was used for normalization. Each layer had fewer neurons than the preceding layer.
- A sigmoid function was used in the last layer to output the probabilities for the predicted target.
- A Dropout layer was used after every Dense layer to avoid overfitting.
- A callback EarlyStopping which monitored the validation loss metric was also used to prevent overfitting.
- The neural network was trained for 600 epochs , but the training was stopped at 85 epochs due to the early stop callback.
- The model's metrics showing the model's Loss, Validation-Loss, Accuracy and Validation-Accuracy are shown in the fig-17 below.

The last recorded metrics are shown in table below.

loss	accuracy	val loss	val accuracy
0.4256	0.8271	0.3684	0.8431

CVD (Cardiovascular Disease) Classification using SVM

For this task a SVD model was trained, to classify patient cases as having the Target (CVD) diseases or not. The methodology is summarized as:

- Exploratory data analysis was conducted using the Seaborn library the results of which are summarized and displayed graphically in the introduction section.
- 11 data columns were selected as features and the 'Target' column was selected as our label.

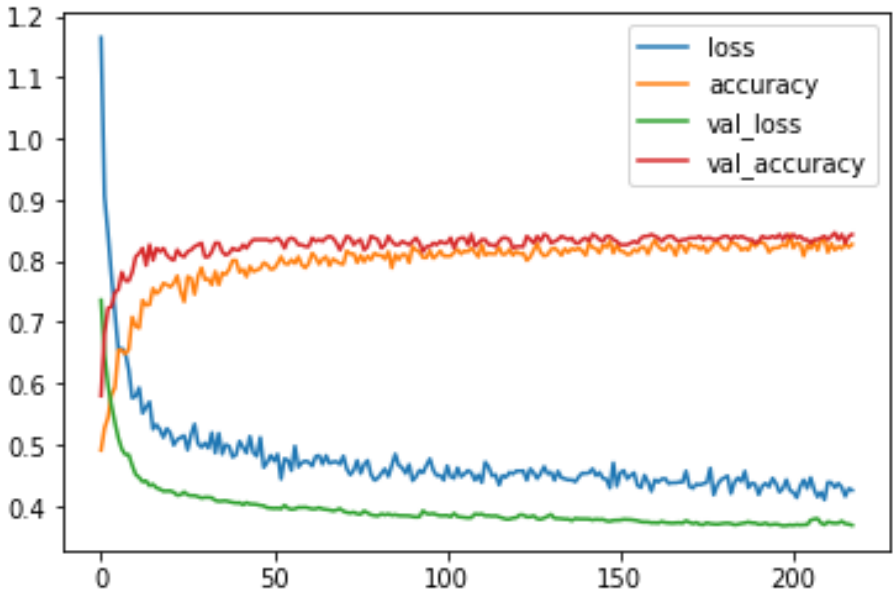

Figure 17: Performance Metrics of ANN

- The set was distributed into training and testing data and then scaled using a Minmax scaler.
- The model was evaluated using the Confusion matrix and Classification Report metrics. These results are summarized and compared in the Comparison section along with other algorithms.

CVD (Cardiovascular Disease Classification) using Decision Trees

For this task a Decision Tree model was trained, to classify patient cases as having the Target (CVD) diseases or not. The methodology is summarized as:

- Exploratory data analysis was conducted using the Seaborn library the results of which are summarized and displayed graphically in the introduction section.
- 11 data columns were selected as features and the 'Target' column was selected as our label.
- The set was distributed into training and testing data and then scaled using a Minmax scaler.
- The model was trained using a Decision Tree classifier with the entropy criterion with a max depth of 4. The image of the Tree is shown in the notebook.
- The model was evaluated using the Confusion matrix and Classification

Report metrics. These results are summarized and compared in the Comparison section along with other algorithms.

Diabetes Using Logistic Regression:

For this task a Logistic regression model was trained, to classify patient cases as having the Target (Diabetes) diseases or not. The methodology is summarized as:

- 8 data columns are selected as features and the target column (Outcome) is selected as the label.
- The dataset was distributed into training and testing data and then scaled using Standard scaler.
- The model was trained using a Logistic Regression method and the accuracy was predicted.
- The model was evaluated using the Confusion matrix and Classification Report metrics. These results are summarized and compared in the Comparison section along with other algorithms.

Diabetes Using Decision Tree Classification:

For this task a Decision Tree model was trained, to classify patient cases as having the Target (Diabetes) diseases or not. The methodology is summarized as:

- 8 data columns are selected as features and the target column (Outcome) is selected as the label.
- The model was trained using a Decision Tree classifier with the entropy criterion with a max depth of 3. The image of the Tree is shown in the notebook.
- The model was evaluated using the Accuracy, Confusion matrix and Classification Report metrics of training and testing data. These results are summarized and compared in the Comparison section along with other algorithms.

Diabetes Using SVM:

For this task a SVC model was trained, to classify patient cases as having the Target (Diabetes) diseases or not. The methodology is summarized as:

- 8 data columns are selected as features and the target column (Outcome) is selected as the label.
- Exploratory data analysis was conducted using the Seaborn library the results of which are summarized and displayed graphically in the introduction section.
- The model was trained using a Support Vector Classifier and then Accuracy, Confusion Matrix and the Classification Report metrics were evaluated of both training and testing data.

Example Analysis

Here we use the **Breast Cancer Dataset** as an example to illustrate the process.

Define the question

Breast cancer is the most common cancer amongst women in the world. It accounts for 25% of all cancer cases, and affects millions of people each and every year. It starts when cells in the breast begin to grow out of control. These cells usually form tumors that can be seen via X-ray or felt as lumps in the breast area. The key challenge against its detection is how to classify tumors into malignant (cancerous) or benign(non cancerous).

Here we use Breast Cancer Wisconsin (Diagnostic) Dataset. It provides rich features describing tumors (radius, texture, perimeter, area, smoothness, compactness, concavity, concave points, symmetry, fractal dimension) from different angles(mean, standard deviation, and worst). Additionally it has preliminary diagnosis as M(alignant) or B(enign). So it's adequate for modeling.

Based on the dataset, we think it's perfect to complete the analysis of these tumors using machine learning various classification algorithms.

Tidy the data

The dataset contains 569 records with 32 attributes. These attributes' name, non-null count, data type represented in the dataset are shown in fig-18.

```
        ***** RAW DATASET *****


<class 'pandas.core.frame.DataFrame'>
RangeIndex: 569 entries, 0 to 568
Data columns (total 32 columns):
 #   Column                   Non-Null Count  Dtype
---  ------                   --------------  -----
 0   id                       569 non-null    int64
 1   diagnosis                569 non-null    object
 2   radius_mean              569 non-null    float64
 3   texture_mean             569 non-null    float64
 4   perimeter_mean           569 non-null    float64
 5   area_mean                569 non-null    float64
 6   smoothness_mean          569 non-null    float64
 7   compactness_mean         569 non-null    float64
 8   concavity_mean           569 non-null    float64
 9   concave points_mean      569 non-null    float64
 10  symmetry_mean            569 non-null    float64
 11  fractal_dimension_mean   569 non-null    float64
 12  radius_se                569 non-null    float64
 13  texture_se               569 non-null    float64
 14  perimeter_se             569 non-null    float64
 15  area_se                  569 non-null    float64
 16  smoothness_se            569 non-null    float64
 17  compactness_se           569 non-null    float64
 18  concavity_se             569 non-null    float64
 19  concave points_se        569 non-null    float64
 20  symmetry_se              569 non-null    float64
 21  fractal_dimension_se     569 non-null    float64
 22  radius_worst             569 non-null    float64
 23  texture_worst            569 non-null    float64
 24  perimeter_worst          569 non-null    float64
 25  area_worst               569 non-null    float64
 26  smoothness_worst         569 non-null    float64
 27  compactness_worst        569 non-null    float64
 28  concavity_worst          569 non-null    float64
 29  concave points_worst     569 non-null    float64
 30  symmetry_worst           569 non-null    float64
 31  fractal_dimension_worst  569 non-null    float64
dtypes: float64(30), int64(1), object(1)
memory usage: 142.4+ KB
```

Figure 18: Breast Cancer Dataset attributes

We eliminate the _worst features since these extreme features don't represent a fair distribution of records. Also we also eliminate the _se features because they are similar to the _mean features while the _mean features are better to represent the records considering the size of the dataset. Additionally, we change the diagnosis feature from categorical[M, B] to numerical[1, 0] for easy processing later. Now data showing in fig-19 are neat and ready for further analysis.

```
                    ***** DATA CLEANED and are READY for FURTHER ANALYSIS *****
```

```
/var/folders/c6/p6pvw6611071bx94m2j2j1g80000gn/T/ipykernel_36048/4091969521.py:2: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning
-a-view-versus-a-copy
  ds['diagnosis'] = ds['diagnosis'].map({'M':1, 'B':0})
```

]:

	diagnosis	radius_mean	texture_mean	perimeter_mean	area_mean	smoothness_mean	compactness_mean	concavity_mean	concave points_mean	symmetry_mean
0	1	17.99	10.38	122.80	1001.0	0.11840	0.27760	0.30010	0.14710	0.2419
1	1	20.57	17.77	132.90	1326.0	0.08474	0.07864	0.08690	0.07017	0.1812
2	1	19.69	21.25	130.00	1203.0	0.10960	0.15990	0.19740	0.12790	0.2069
3	1	11.42	20.38	77.58	386.1	0.14250	0.28390	0.24140	0.10520	0.2597
4	1	20.29	14.34	135.10	1297.0	0.10030	0.13280	0.19800	0.10430	0.1809
...	...	...	...	...	...	...	...	...	...	...
564	1	21.56	22.39	142.00	1479.0	0.11100	0.11590	0.24390	0.13890	0.1726
565	1	20.13	28.25	131.20	1261.0	0.09780	0.10340	0.14400	0.09791	0.1752
566	1	16.60	28.08	108.30	858.1	0.08455	0.10230	0.09251	0.05302	0.1590
567	1	20.60	29.33	140.10	1265.0	0.11780	0.27700	0.35140	0.15200	0.2397
568	0	7.76	24.54	47.92	181.0	0.05263	0.04362	0.00000	0.00000	0.1587

569 rows × 11 columns

Figure 19: Reduced Dataset

Explore the data

Now we have the mean features together with the diagonal feature for further analysis. We can take a closer look at these features as well as from a statistical point of view, showing in fig-20.

```
                    ***** FEATURES in a STATISTICAL VIEW *****
```

	count	mean	std	min	25%	50%	75%	max
diagnosis	0.0	NaN	NaN	NaN	NaN	NaN	NaN	NaN
radius_mean	569.0	14.127292	3.524049	6.98100	11.70000	13.37000	15.78000	28.11000
texture_mean	569.0	19.289649	4.301036	9.71000	16.17000	18.84000	21.80000	39.28000
perimeter_mean	569.0	91.969033	24.298981	43.79000	75.17000	86.24000	104.10000	188.50000
area_mean	569.0	654.889104	351.914129	143.50000	420.30000	551.10000	782.70000	2501.00000
smoothness_mean	569.0	0.096360	0.014064	0.05263	0.08637	0.09587	0.10530	0.16340
compactness_mean	569.0	0.104341	0.052813	0.01938	0.06492	0.09263	0.13040	0.34540
concavity_mean	569.0	0.088799	0.079720	0.00000	0.02956	0.06154	0.13070	0.42680
concave points_mean	569.0	0.048919	0.038803	0.00000	0.02031	0.03350	0.07400	0.20120
symmetry_mean	569.0	0.181162	0.027414	0.10600	0.16190	0.17920	0.19570	0.30400
fractal_dimension_mean	569.0	0.062798	0.007060	0.04996	0.05770	0.06154	0.06612	0.09744

Figure 20: Data statistics

A heatmap showing the correlation of the features with each other is shown in fig. The image shows graphically how each feature correlates with each other feature. This analysis shows that the following features [‘radius_mean’, ‘texture_mean’, ‘perimeter_mean’, ‘area_mean’, ‘smoothness_mean’, ‘compactness_mean’, ‘concavity_mean’, ‘concave points_mean’, ‘symmetry_mean’, ’fractal_dimension_mean] are positively correlated with the target variable[diagnosis] , with[‘radius_mean’, ‘perimeter_mean’, ‘area_mean’, ‘concave points_mean’, ‘concavity_mean’] showing the highest correlation.

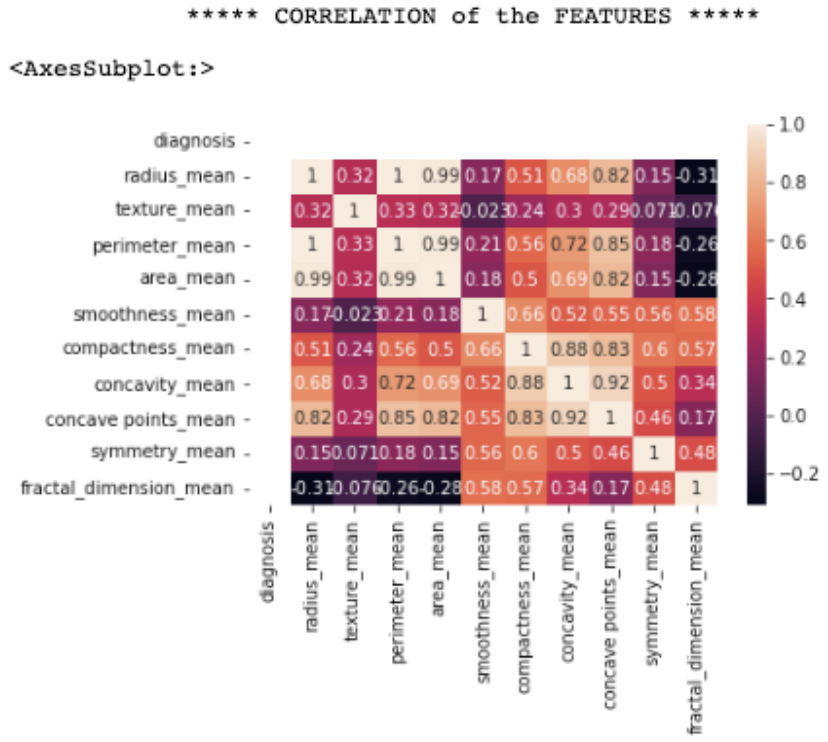

Figure 21: Heatmap

The above is proved by fig-21. showing indeed [‘radius_mean’, ‘perimeter_mean’, ‘area_mean’, ‘concave points_mean’, ‘concavity_mean’] are major contributors to diagnosis.

Furthermore we understand the distribution of Malignant vs. Benign graphed in fig-23.

```
***** MAJOR CONTRIBUTORS to the diagnosis *****
<AxesSubplot:>
```

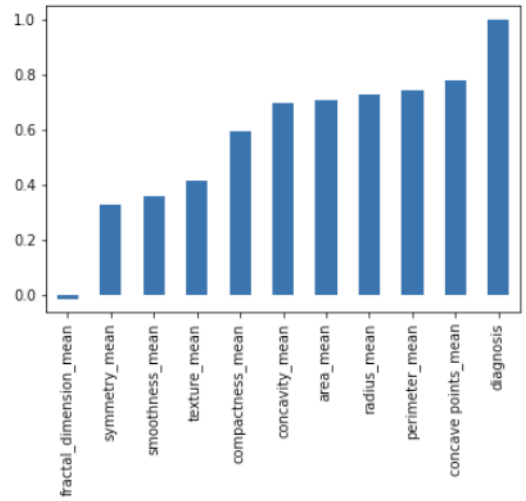

Figure 22: Major Contributions to the diagnosois

```
***** COUNTS of 1(MALIGNANT) vs. 0(BENIGN) *****
: <AxesSubplot:>
```

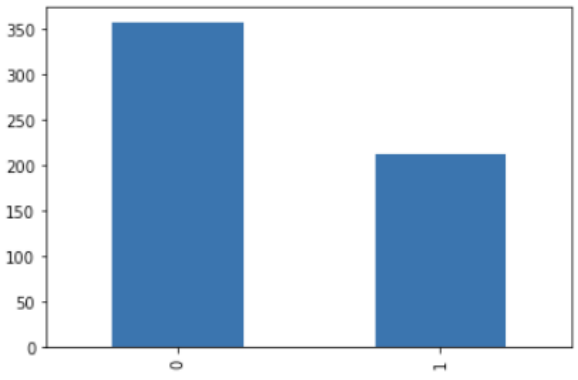

Figure 23: Distribution of Breast cancer dataset

Moreover we have an impression regarding both distribution of a single feature as well as its relationships with other features in fig-24.

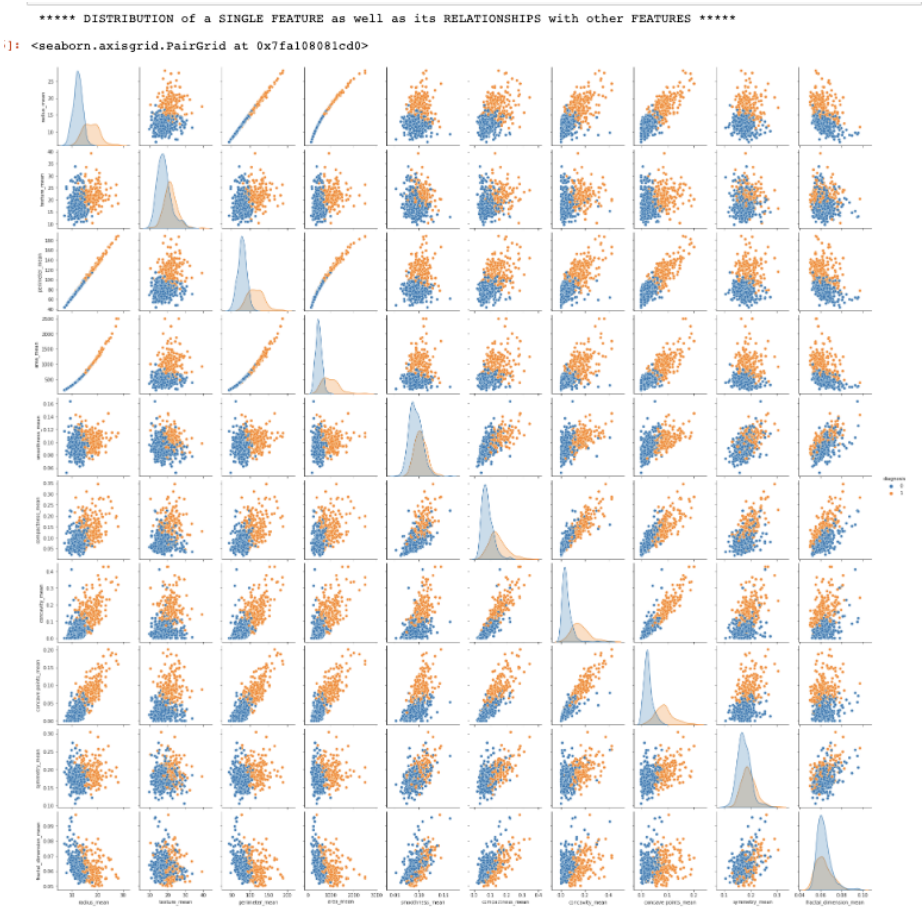

Figure 24: Pairplot of data

Use the models in one shot (RandomForestClassifier, KNeighborsClassifier, GaussianNB, SGDClassifier)

10 data columns ending with _mean are selected as features, and the diagnosis column is selected as a label.

StandardScaler is used to standardize X.

The dataset is split into training and testing data as 80% vs. 20%.

In one shot, multiple models (RandomForestClassifier, KNeighborsClassifier, GaussianNB, SGDClassifier) are used.

Perform the analysis

In one shot again, all models are evaluated using accuracy_score, precision_score, recall_score, f1_score, classification_report, confusion_matrix. Please see fig-25 & fig-26.

```
Model: RandomForestClassifier()
Accuracy: 0.9473684210526315
Precision: 0.926829268292683
Recall: 0.926829268292683
F1: 0.926829268292683
              precision    recall  f1-score   support

           0       0.96      0.96      0.96        73
           1       0.93      0.93      0.93        41

    accuracy                           0.95       114
   macro avg       0.94      0.94      0.94       114
weighted avg       0.95      0.95      0.95       114

AxesSubplot(0.125,0.125;0.62x0.755)


Model: KNeighborsClassifier()
Accuracy: 0.9385964912280702
Precision: 0.925
Recall: 0.9024390243902439
F1: 0.9135802469135802
              precision    recall  f1-score   support

           0       0.95      0.96      0.95        73
           1       0.93      0.90      0.91        41

    accuracy                           0.94       114
   macro avg       0.94      0.93      0.93       114
weighted avg       0.94      0.94      0.94       114

AxesSubplot(0.125,0.125;0.496x0.755)
```

Figure 25: Performance Metrics Breast Cancer Dataset

```
Model: GaussianNB()
Accuracy: 0.9298245614035088
Precision: 0.9714285714285714
Recall: 0.8292682926829268
F1: 0.8947368421052632
              precision    recall  f1-score   support

           0       0.91      0.99      0.95        73
           1       0.97      0.83      0.89        41

    accuracy                           0.93       114
   macro avg       0.94      0.91      0.92       114
weighted avg       0.93      0.93      0.93       114

AxesSubplot(0.125,0.125;0.3968x0.755)


Model: SGDClassifier()
Accuracy: 0.956140350877193
Precision: 1.0
Recall: 0.8780487804878049
F1: 0.9350649350649352
              precision    recall  f1-score   support

           0       0.94      1.00      0.97        73
           1       1.00      0.88      0.94        41

    accuracy                           0.96       114
   macro avg       0.97      0.94      0.95       114
weighted avg       0.96      0.96      0.96       114

AxesSubplot(0.125,0.125;0.31744x0.755)
```

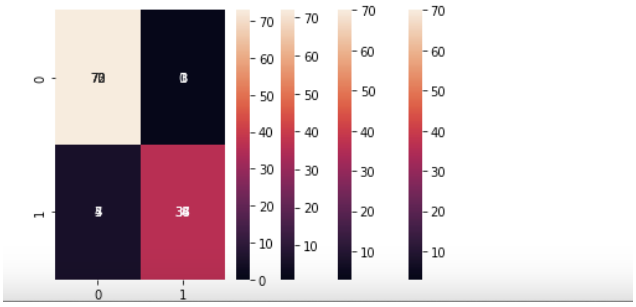

Figure 26: Performance Metrics Breast Cancer Dataset

Check results

All models' overall scores are pretty high: 93%–96% while there's no champion forever since data are shuffled for each and every run.

Btw:

I test RandomForestClassifier, KNeighborsClassifier, GaussianNB and SGDClassifieruse, together with LogisticRegression and SupportVectorMachine, all of them achieve 93% - 96%. I keep the former four while drop the latter two since others use them.

I use the former four models in one shot so that it's code-efficient and it's easy for model comparison.

Comparisons

In this section we compile and compare the various Evaluation metrics of each and every model.

CVD (Cardiovascular Disease) Classification

Three models were created for the CVD dataset, as listed below: - CVD (Cardiovascular Disease) Classification using ANNs - CVD (Cardiovascular Disease) Classification using SVM - CVD (Cardiovascular Disease) Classification using Decision Trees

The image of Confusion Matrix for each model is shown as below in fig-27-29 :

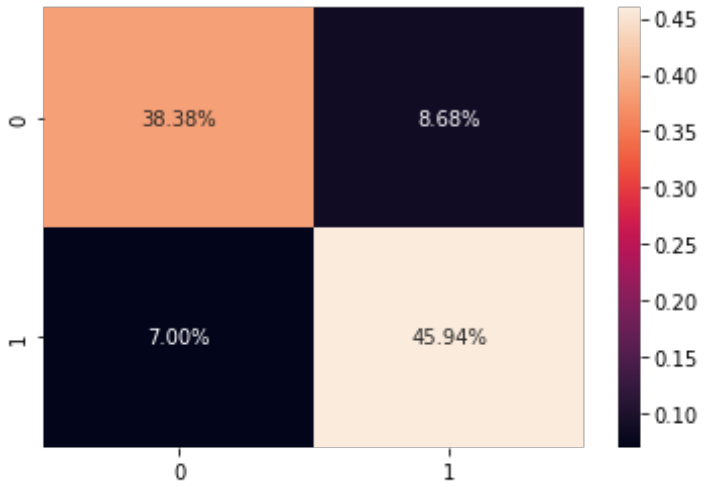

Figure 27: Confusion Matrix for ANN model

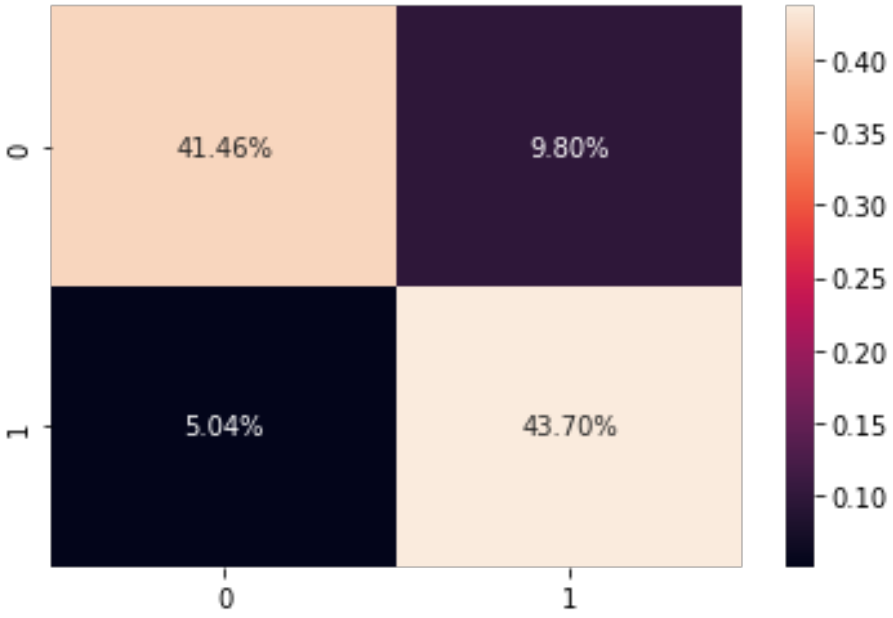

Figure 28: Confusion Matrix for SVM model

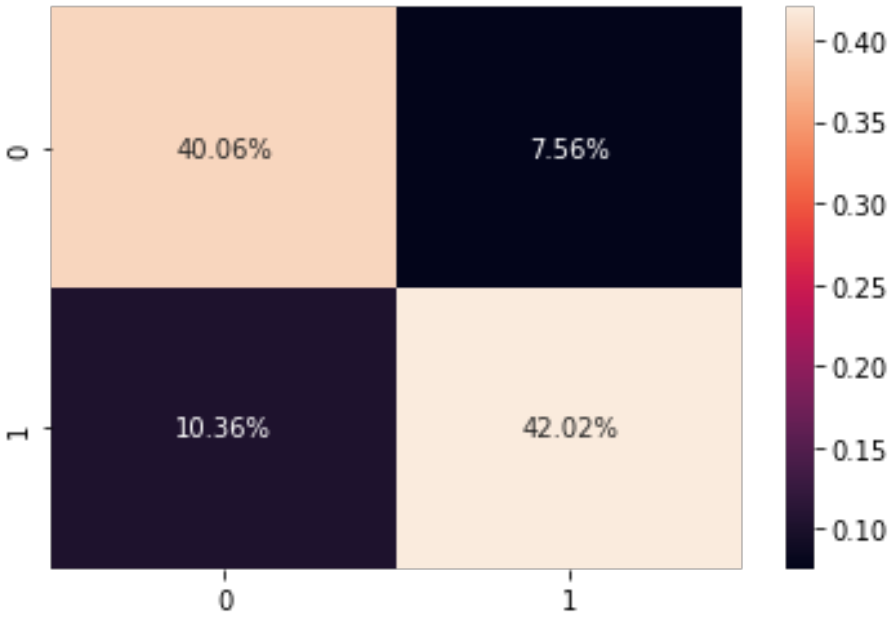

Figure 29: Confusion Matrix for Decision Tree Classifier

For easy comparison a table is shown below that compiles all the True Positive, False Positive, True Negative and False Negative values gathered from the confusion matrix for each model trained on the CVD dataset.

Model name	TP	FP	FN	TN
SVM	41.46%	9.80%	5.04%	43.70%
ANN	38.38%	8.68%	7%	45.94%
Decision Tree	40.06%	7.56%	10.36%	42.02%

The classification report for each model trained on the CVD dataset is shown in the table with its respective label below.

Table 4: Classification Report for ANN model

	Precision	Recall	F-1
0	0.85	0.82	0.83
1	0.84	0.87	0.85
accuracy			0.84

Table 5: Classification Report Decision Tree model

	Precision	Recall	F-1
0	0.85	0.82	0.83
1	0.84	0.87	0.86
accuracy	0.85		

Table 6: Classification Report for SVM model

	Precision	Recall	F-1
0	0.89	0.81	0.85
1	0.82	0.90	0.85
accuracy			0.85

Comparing all the models given their metrics, all the models performed somewhat similarly but out of the three models the SVM model performed the best.

Diabetes Classification

Models used for the Diabetes data set are listed below: - Diabetes Classification Using Logistic Regression - Diabetes Classification Using Decision Tree Classification - Diabetes Classification Using SVM

The image of Confusion Matrix for each model is shown as below fig:

Model name	TP	FP	FN	TN
Logistic Regression	132	36	14	49
Decision Tree	124	31	22	54
SVM	132	35	14	50

The classification report for each model trained on the Diabetes dataset is shown in the table with its respective label below.

Table 8: Classification Report for Logistic Regression

	Precision	Recall	F-1
0	0.79	0.90	0.84
1	0.78	0.58	0.66
accuracy			0.78

Table 9: Classification Report for Decision Tree Classifier

	Precision	Recall	F-1
0	0.80	0.85	0.82
1	0.71	0.64	0.67
accuracy			0.77

Table 10: Classification Report for SVM

	Precision	Recall	F-1
0	0.79	0.90	0.84
1	0.78	0.59	0.67
accuracy			0.79

Conclusion

Learning is possible considering the two step process, for at least finite hypothesis sets. Our Datasets and hypothesis sets are fixed, the data is assumed to be IID (Independent Identically Distributed) generated from a target distribution P[y|x], from this same distribution we obtain Independent test points for each dataset with respect to which we calculate Eout. This means that when we are considering the first step of the Two Step Learning approach i.e Eout(g) nearly

equal to Ein, the testing and training points come from the same distribution. Next we ensure that the same error measure is used by the algorithm for selecting the best hypothesis as well as for when comparing "g" nearly equal to "f" that is Eout.

We understand that its not the question of choosing the most complicated hypothesis set, because a much smaller hypothesis set may very well be able to adapt to the problem (Optimal Tradeoff).

The fundamental Learning approach dictates that for a complex Target function we would need a larger set of Hypothesis and in order to maintain Eout nearly equal to Ein we would need a larger set of data.

Our experiments are made up of complex datasets and incidentally their respective target functions are also complex, so the more the data our models has to train on would be beneficial. One major attribute of learning is the "Error", which is user defined and to measure error in our experiments we have used metrics like Classification report and Confusion matrices to quantify the error of our trained models. This error effects how we select "g", our hypothesis.

Additionally we have implemented multiple models and compiled their Error metrics so that selection of a suitable model specific to the dataset, that outperforms others can be selected.

References

- Cardiovascular Disease Prediction Dataset - OpenML
- Breast Cancer Prediction Dataset
- Diabetes Dataset - OpenML