1. Data Mining Goal

The dataset we have used for the purpose of this project was collected from 1971 to 1982. It is a national health and nutrition examination survey. It consists of various behavioural and clinical attributes for 1629 individuals. Using this dataset, we aim to predict whether an individual is likely to die within the next ten years (yes/no) using data mining techniques.

2. Dataset description

a. Name: National Health and Nutrition Examination Survey (NHEFS)
b. Source: https://wwwn.cdc.gov/nchs/nhanes/nhefs/
c. Dimensions: 1629 rows/ samples and 50 columns/features
d. Attribute description

Sr No.	Variable name	Description
1	active	How active are you? 0: very 1: moderate 2:inactive
2	age	Age of individual
3	alcoholfreq	How often do you drink? 0: almost everyday 1: 2 to 3 times/month 2: 1 to 4 times/month 3: less than 12 times a year 4: no alcohol
4	alcoholhowmuch	How much alcohol do you drink?
5	alcoholpy	Have you had a drink in the past year?
6	alcoholtype	Which type of alcohol do you drink?
7	allergies	Do you have any allergies? 0:No 1:Yes
8	asthma	Do you have asthma? 0:No 1:Yes
9	birthcontrol	Have you take birth control pills in the last 6 months? 0:No 1:Yes
10	birthplace	Place of birth (state codes)
11	boweltrouble	Do you use bowel trouble medication?
12	bronchitis	Do you have chronic

		bronchitis? 0:No 1:Yes
13	cholesterol	Serum cholesterol level (mg/100ml)
14	chroniccough	Do you have Chronic cough?
15	colitis	Indicates whether the individual has colitis
16	dadth	Day of death
17	dbp	Diastolic blood pressure (mg/100ml)
18	diabetes	Diabetes
19	education	Education status
20	exercise	Amount of exercise done 0:A lot 1:Moderate 2:None
21	hayfever	Hayfever 0:No 1:Yes
22	hbp	High blood pressure
23	hbpmed	High blood pressure medication 0:No 1:Yes
24	headache	Do you have headaches?
25	hepatitis	Hepatitis 0:No 1:Yes
26	hf	Heart failure
27	hightax	Are you living in a high taxed state?
28	nervousbreakdown	Mentall illness
29	otherpain	Any other body pain
30	pepticulcer	Peptic Ulcer 0:No 1:Yes
31	pica	Do you eat non standard/toxic food?
32	polio	Polio 0:No 1:Yes
33	Pregnancies	Number of pregnancies
34	price	Average tobacco price in

		state of residence
35	qsmk	Did you quit smoking?
36	race	Ethnicity
37	sbp	Systolic blood pressure
38	school	Highest school grade attended
39	seqn	Patient id
40	sex	0:male 1:female
41	smokeintensity	Number of cigarettes/per day
42	smokeyrs	Years of smoking
43	tax	Tobacco tax in state of residence
44	tb	Tuberculosis 0:No 1:Yes
45	tumor	Malignant tumor 0:No 1:Yes
46	weakheart	Do you use weak heart medication?
47	wt	Weight (in kg)
48	wtloss	Are you experiencing weight loss?
49	yrdth	Year of death
50	**Death (Class attribute)**	**Did the individual die in next 10 years? 0:No 1:Yes**

3. Data Mining tool used

For the purpose of this project, Jupyter notebook has been used for implementation and Python programming language has been used for coding. Python is a high level, object oriented programming language released in 1991. It is very interactive, intuitive, user friendly, has a simple structure and syntax, and is easy to interpret. We have used numpy library to perform arithmetic and logical operations, pandas for data manipulation and analysis, matplotlib and seaborn for data visualizations, sklearn for feature engineering and classification as well as feature selection algorithms.

4. Classification algorithms used

A Logistic Regression: It is a commonly used supervised learning classification algorithm. It is a generalised linear model used when the class or dependent attribute is categorical or nominal. It computes the weighted sum of input features, y = B.X = sum(B0+B1.x1+…+Bn.xn) where x denotes every independent or explanatory variable and B denotes the corresponding coefficient estimated using minimum squared error.

B0 denotes the bias/ intercept term and B1,..Bn denote the slopes. It then uses logistic/sigmoid function to compute the probability of an instance belonging to each class.

p = 1/(1+e^-y)

The instance is then assigned a class label based on these computed probability p. Usually a threshold of 0.5 is used. If p>0.5 the instance belongs to class 1 and if p<0.5 the instance belongs to class 0.

It uses the concept of odds which is equal to p/(1-p) where p is the probability of instance belonging to a particular class. Taking log(odds), also called logit function, we get:

log(odds)= B0+B1.x1+…Bnxn

Thus p/(1-p)= e^(B0+B1.x1+…Bnxn)

p=e^y/(1+e^y)=1/(1+e^-y)

B. KNearest Neighbor

It is another supervised learning classification algorithm. It is an example of instance based learning. There is no training stage as such and predictions are based on all the stored data. The first step is to calculate the distance from the point which needs to be labelled to all other data points. Distance metrics such as Euclidean distance or L2 norm, Manhattan distance or L1 norm, etc. can be used for this purpose. Then we must decide the number of neighbors, k. k must be odd so that a majority label can be obtained. The k closest data points (also called “neighbors”) are selected. The majority class of these neighbors is then assigned to the new instance. Since it is a distance based method, scaling the features is important in KNN to reduce effect of different units/scales/ magnitudes of the features. Note that different values for k might give different results. Optimal number of neighbors can be selected by using a learning curve of k vs error rate. In our project, we found optimal number of neighbors to be 3.

	K	Accuracy_Score
5	11	0.791381
4	9	0.802822
3	7	0.808924
2	5	0.825706
1	3	0.853928
0	1	0.893593

C. Decision Tree

It is represented in the form of a flow chart, consisting of a root node, multiple internal nodes for testing conditions and leaf nodes for final classifications. The edges connecting the nodes represent results of test conditions. It can be binary where a node branches to exactly 2 other nodes or non-binary. The aim is to have a simple/shallow or pure tree, i.e.,samples belonging to only one class after partitioning. Attribute selection measures such as information gain, gain ratio, gain index are used to select attributes at each node. We can stop partitioning when there are no tuples left to classify, no attributes left for further partitioning or all samples at a node belong to the same class. The process of selecting test attribute and partitioning based on test outcomes continues recursively until any of the stopping criteria are

met. In case of numeric features at the internal nodes, discretization is used to classify the samples after finding the appropriate split point.

Initial Entropy/ Information is computed (information needed to classify a sample):

H= -(p1log2(p1)+….+pnlog2(pn)) where p1,..pn denote possible probabilities

Then entropy is computed for each attribute (information needed to classify a sample after splitting on a particular attribute) and information gain is calculated by subtracting it from initial entropy. Attribute with the maximum information gain is chosen as the test attribute. Decision trees are difficult to scale and are prone to overfitting. This can be dealt with by using pre-pruning, i.e., halting decision tree early based on a certain threshold, or post pruning, i.e., removing branches from an already constructed tree.

D. Random Forest

It is a type of ensemble method which combines multiple base learners or decision trees. Therefore it has higher accuracy than a single decision tree. It is a bagging technique where base classifiers are generated in parallel. Random samples with replacement are extracted from the data (bootstrap) and a separate Decision Tree model is built on each of these sample sets. A subset of features is also randomly chosen for each tree. Attribute selection measures such as information gain, gain ratio are used to decide test attributes at each node of the decision trees from the randomly selected subset of features. Each tree will predict a class label for the new/unseen instance. A majority vote is taken and assigned to the instance. Each base model has an equal vote here unlike boosting methods. Random forest models give good performance and are robust to outliers. However they are computationally more expensive and not suitable for smaller datasets. Parameters such as number of weak learners, maximum features per base model, maximum depth of each tree need to be additionally specified.

E. Support Vector Classifier

SVC uses a linear decision boundary or a hyperplane to separate instances of the two classes. Two parallel planes are drawn, which pass through the closest data points on either side of the hyperplane. These points are called support vectors. The aim is to maximize margin between these two lines, i.e. maximizing distance between the nearest points of either class and the hyperplane. Doing so reduces chances of misclassification. These margins can be soft or hard. Soft margins allow room for some error/ignore certain constraints to increase generalization of the model. On the other hand, hard margins enforce all constraints and are more rigid. In case data is not linearly separable, kernel function is used to transform it into higher dimensional space. Examples include polynomial, sigmoid and RBF (radial basis function) kernels.

6.Attribute selection methods

A. Fisher score or chi square test

It is a type of filtering method. A Chi square statistic is calculated for each categorical feature and class attribute.

Steps of a chi square test:
Set hypothesis: Here null hypothesis H0 states that the 2 features are not correlated. Alternate hypothesis H1 states that features are correlated.
Decide test statistic: Chi square test statistic is used.
Decision rule: If X^2> critical X^2 value, reject H0. Otherwise do not reject H0.
Critical X^2= X^2(alpha,df) where degree of freedom df= (number of rows-1)*(number of columns -1) from the contingency table. This can be got from the distribution table. Alpha indicates significance level.
Compute test statistic: sum((observed-expected)^2/expected)
Conclusion: Compare test statistic and critical value and conclude whether features are correlated or not.
Features highly correlated with the target variable are then selected. This means features with higher chi square value (null hypothesis will be rejected) are more dependent on class attribute and are selected.

B. Recursive Feature Selection
It uses a machine learning model at its core and acts as a wrapper function for this model. For this project, a logistic regression model has been used. It fits the model on the data recursively until the required number of features are left. It starts by fitting the model on all the features, ranks them based on importance and removes the least important feature. Then it again fits the model on the remaining subset of features. This process of removing least important feature continues till we are left with desired number of features. Feature importance can be gathered either from the model itself (using feature_importances_ attribute or coeff_property in case of logistic regression) or using additional techniques.

C. Mutual information
It is also a filtering method. If one variable is known, mutual information gives an estimate of how much information we can obtain about another variable. If variables are not correlated, knowing one of them will not give any information about the other. Thus mutual information = 0. It is a measure of dependency between the variables. If features have high correlation, knowing one of the variables would help determine value of the other, thus mutual information value will be high.
$I(X ; Y) = H(X) – H(X \mid Y)$
I(X;Y) denotes mutual information of X and Y
H(X) is entropy/ information for X.
H(X|Y) is entropy for X given Y.

D. Anova f value
It is a type of filtering method. A similar 5 step hypothesis test procedure is followed as in chi square test where null hypothesis states that there is no relationship between explanatory and response variable. Alternate hypothesis states that there is a relation between the two variables. F statistic is used.
F= MS Reg/MS Res
Mean Square Regression= Regression Sum of Squares/k, where k is the number of predictors. (Regression degree of freedom=k)

Mean Square Residual= Residual Sum of Squares/n-k-1, where n is the number of samples. (Residual degree of freedom=n-k-1)

$$\text{Reg SS}=\sum_{i=1}^{n}(\hat{y}_i-\bar{y})^2 \text{ , Res SS}=\sum_{i=1}^{n}(y_i-\hat{y}_i)^2$$

A critical F value is obtained from distribution table=F (Reg df,Res df, alpha). If test statistic > critical value, reject H0 and accept that there is a relationship between explanatory and target variable. In this way, anova f value is computed for each feature and target variable. Features correlated with target variable(higher f value) are selected.

E. Forward Selection

It is a wrapper method. It starts with no features and at every iteration, a single feature is added at a time as long as it helps improve model performance. The feature which improves model performance the most is added first. The criteria used to determine performance varies (highest accuracy, lowest p value, etc.). In this case, we have used R2 score for this purpose. It adds one feature at a time and tests at every step in order to achieve a reduced set of features.

7. Features selected using each of the attribute selection methods:

Method	Selected features
Fisher score	['sbp', 'smokeyrs', 'asthma', 'polio', 'nerves', 'hbpmed', 'active', 'pregnancies', 'cholesterol', 'tax']
Recursive feature elimination	['race', 'smokeyrs', 'asthma', 'bronch', 'hf', 'hepatitis', 'tumor','alcoholfreq', 'otherpain', 'cholesterol']
Information gain	['cholesterol', 'tax', 'pregnancies', 'active', 'hbpmed', 'smokeyrs','qsmk', 'polio', 'hf', 'pica']
Anova f value	['pregnancies', 'tax', 'active', 'hbpmed', 'polio', 'cholesterol', 'nerves', 'asthma', 'hf', 'sbp']
Forward selection	['income', 'wt', 'bronch', 'tb', 'hf', 'colitis', 'hayfever', 'polio','alcoholpy', 'pregnancies']

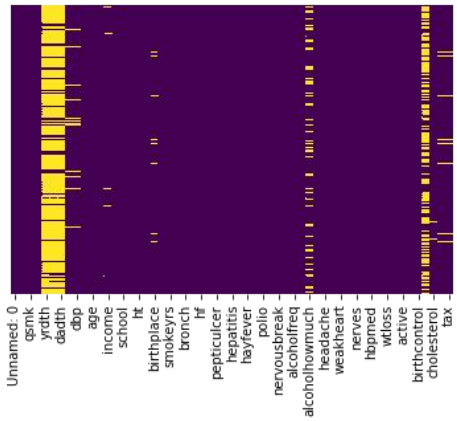
Observation: Features selected using the 5 methods were not the same.

8. Data Mining Procedure

A.Data Acquisition: We sourced our dataset from the Centre for Disease Control and Prevention website.

B. Exploratory Data Analysis and Data Pre-processing-

I. Missing data- We found missing values in 12 columns. These are shown in the missingness map below. Columns with majority values as missing were dropped. For the remaining features, we replaced null values by the median of the column with respect to the class label. In case of categorical columns with missing data, the null values were replaced by mode of the column.



II. Noisy data-We checked for outliers using the inter quartile range criteria.

IQR= Q3 - Q1

Lower bound=Q1 - 1.5*IQR

Upper bound=Q3 + 1.5*IQR

Values lesser than lower bound or greater than upper bound denote outliers.

Outliers were found in some columns but we decided to keep them due to the sensitive nature of medical data. They show vital variability in the data. It is possible that individuals had extreme values for these attributes, they do not seem like data entry errors. Getting rid of the outliers may increase our model's accuracy but they are essential to show the natural variation in medical data.

III. Feature scaling/data transformation- We also performed feature scaling to normalize the range of the data and prevent incorrect predictions due to the tremendous difference between the magnitudes of the features. For this, we used the standard scalar library.

Standardization (also called z-score normalization) replaces values of the features with their z scores.

$$x' = (x - \bar{x}) / \sigma$$

where x is the data point, $\bar{x}$ is the mean of the entire column for a particular feature, and σ is the standard deviation. After applying this technique, distribution is converted in such a way that mean=0 and std deviation=1.

IV. Renaming columns: We renamed the columns as needed. For example: “death” was renamed to “Class” indicating the class attribute.

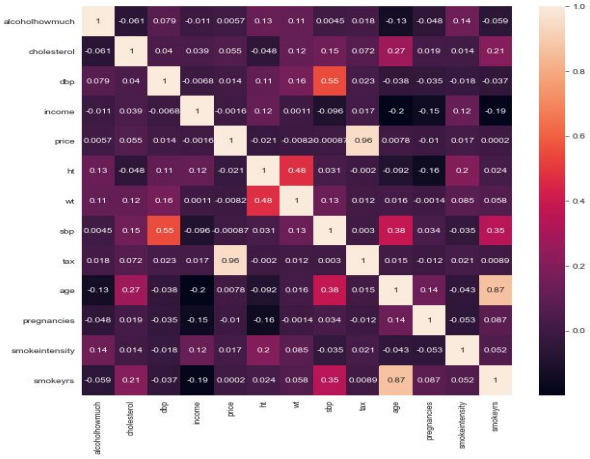
V. Encoding: We checked data types of the features and converted them to the required data types: numeric (integer, float) or categorical (object). Label encoding was performed on the categorical/nominal features. Ordinal encoding was used for ordinal columns like exercise, active where the class labels had a meaningful ranking.

VI. We checked for correlation between features using a correlation matrix which gives the Pearson correlation coefficient between each pair of features.

r=(sum((ai-mean(a))*(bi-mean(b)))/n*sd(a)*sd(b)

If r>0, it indicates positive correlation.

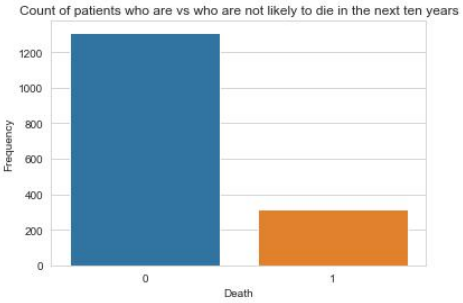
If r<0, it indicates negative correlation.
If r=0, it indicates no correlation.

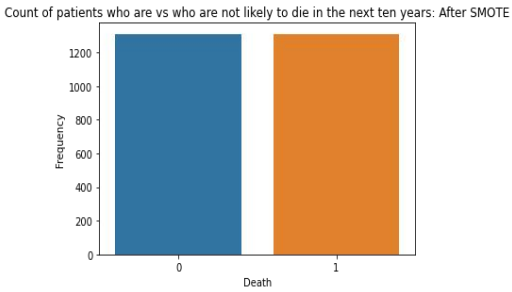


Highest correlations were found between price and tax, age and smoke years, sbp and dbp.

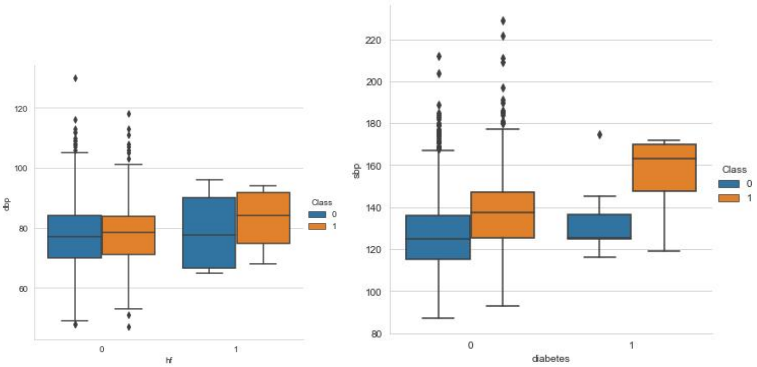
VII. Oversampling

Our dataset was imbalanced and had higher number of class 0 labels (patients who did not die). We performed SMOTE to randomly resample from the minority class (class=1) in order to deal with class imbalance problem. After SMOTE, we got equal number of samples in both classes.

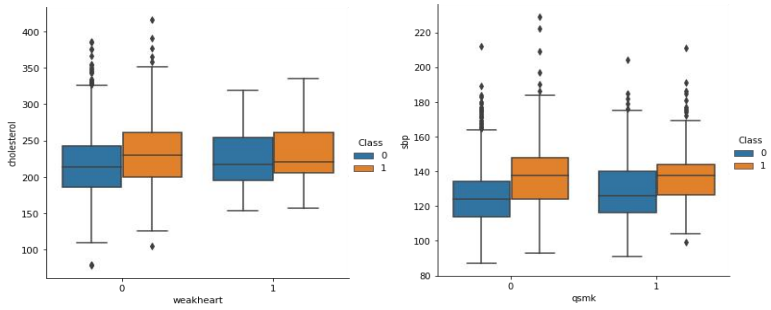




C. Once we were sure of the quality of data, we performed certain visualizations to get more insights into the dataset.

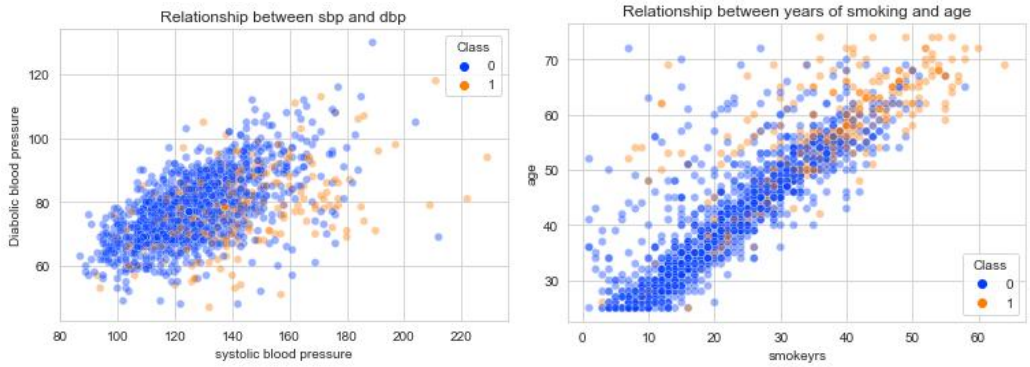


A boxplot was used to see relation between diabolic blood pressure in individuals who had and did not have heart failure, across both class labels. It was observed that individuals with class label 1(who did die in next 10 years) had higher levels of dbp. Similarly, patients who did die in next 10 years had higher level of systolic blood pressure (sbp). Those with class label 1 and diabetes had highest sbp levels. Some other visualizations we made were:



Patients who did die in next 10 years had higher cholesterol. Those with class label=1 who did not quit smoking had highest systolic blood pressure.

Scatterplots were made for pairs of features with highest correlation. They corroborated a strong positive correlation between them,i.e., as level of one feature increases so does the other.



D. Splitting the data- We split the data into train and test sets. 66% of data was used for training and 34% for testing. We used stratify=y to ensure that class distribution is preserved in train and test sets.

```
1 np.unique(y_train,return_counts=True)
```

```
(array([0, 1]), array([865, 865], dtype=int64))
```

```
1 np.unique(y_test,return_counts=True)
```

```
(array([0, 1]), array([446, 446], dtype=int64))
```

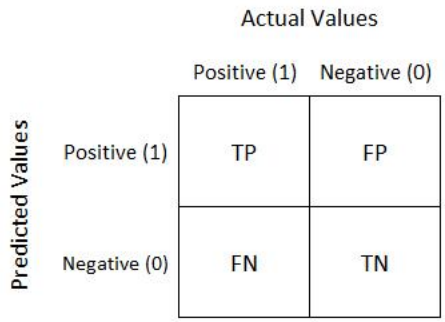
E. 5 classification algorithms were performed, including:

- Logistic Regression
- Decision Tree
- Random Forest
- Support Vector Classifier
- KNearest Neigbor

For each of these, we computed the accuracy, confusion matrix, true positive rate or recall, true negative rate, false positive rate, false negative rate, precision, Fmeasure,MCC, Roc area. We computed overall metrics, class wise metrics and weighted average of metrics.

- Accuracy- It is equal to the number of correct predictions / total predictions. It has a range from 0 to 1. The Accuracy measure merely tells how good or bad our model is but gives no information regarding what is wrong with it or where it is making errors. Hence we evaluated our models based on other metrics as well.

- Confusion matrix is a table that gives insights into how many correct predictions our model has made. It correlates actual and predicted output values. True Positive (TP) values are those which are predicted correctly as positive. False Positive (FP) values are incorrectly predicted as positive. False Negative (FN) values should have been predicted as positive but were not. Lastly, True Negative (TN) values were correctly predicted as negative.



Type 1 Error=FP Rate= FP/FP+TN, Type 2 Error= FN Rate=FN/FN+TP. In these terms we can say Accuracy= TN+TP / (TP+TN+FP+FN).

- Precision or positive prediction value- It is the ratio of true positive to the total positives, i.e., ratio of patients who have been correctly identified with cancer to the total number of patients who have been detected with cancer.

Precision= TP/TP+FP

- Recall or Sensitivity- It is the ratio of true positives to the sum of true positives and false negatives. In simple terms, it is the ratio of correctly predicted values to the total number of actual positive values.

Recall= TP/TP+FN

- F Beta- It is the harmonic mean of precision and recall. If Beta value= 1, it is called F1 score.

$$F_\beta = \frac{(1+\beta^2)\cdot(\text{precision}\cdot\text{recall})}{(\beta^2\cdot\text{precision}+\text{recall})}$$

F1 Score-It takes into account true positive, false positive and false negative but not true negative. A good F1 Score indicates a good precision as well as recall and this way, we do not need to individually focus on any one of them.

F1 Score= 2 * (Precision*Recall / Precision+Recall)

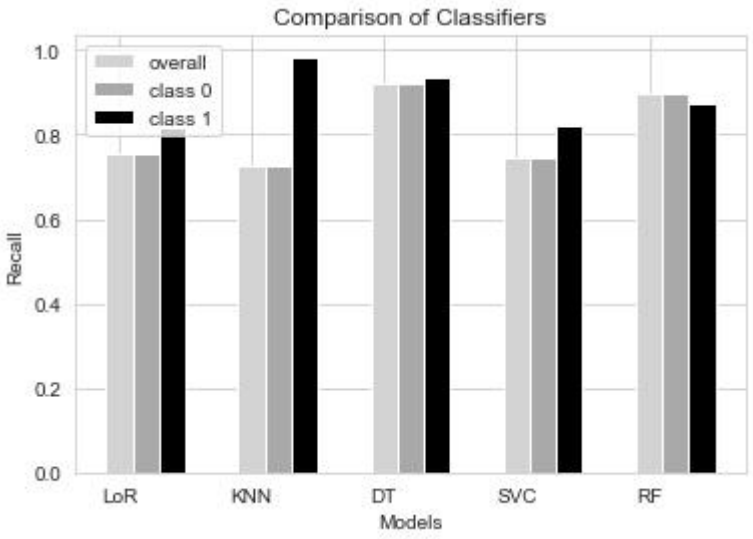
=2*TP/2* TP+FP+FN

- Matthews correlation coefficient- This uses all four categories of the confusion matrix.

MCC=TP*TN-FP*FN/sqrt((TP+FP)(TP+FN)(TN+FP)(TN+FN))

Performance was tested using 10 fold cross validation. In 10-fold cross validation, our dataset will be divided into 10 blocks. Then in the first iteration, out of the 1629 records in our dataset, 1629/10= approximately 163 will be test data and the model will be trained on the remaining 1466 samples, giving us accuracy 1. In iteration 2, the next 163 samples will act as test data and the model will be trained on the remaining samples, giving us accuracy 2. A similar process will be followed for the next 8

iterations. The final accuracy of our machine learning model will be the mean of the 10 accuracies.



Comparison of Recall or TPR of classifiers using 10 fold cross validation. Here Decision Tree had highest overall Recall.

F. Feature selection methods were used to prepare reduced datasets. These techniques were applied on training data and the selected features were extracted from test data. We set the number of features to be selected as 10 for all methods.

- Fisher score
- Recursive Feature Elimination
- Information Gain
- Anova f value
- Forward Selection

The same classification models were then built on reduced training data and tested on reduced test data and evaluated using the same performance metrics.

9. Performance metrics

A. Overall:

Model	Confusion matrix	TPR	TNR	FPR	FNR	Precision	MCC	F measure	ROC area	Recall
Logistic Regression	[978 333 265 1046]	0.75	0.8	0.2	0.25	0.7868	0.5445	0.7658	0.7719	0.75
Decision Tree	[[1202 109] [93 1218]]	0.92	0.93	0.07	0.08	0.9281	0.8459	0.9224	0.9229	0.92
Random Forest	[[1156 155] [162 1149]]	0.88	0.88	0.12	0.12	0.8770	0.7582	0.8794	0.8790	0.88
Support Vector Classifier	[[967 344] [257 1054]]	73.7	80.4	0.2	0.26	0.7900	0.542	0.7629	0.7707	0.74
KNN	[[897 414] [114 1197]]	0.68	0.91	0.09	0.32	0.8872	0.6135	0.7726	0.7986	0.68

Logistic Regression- Reduced features (chi sq)	[[434 25] [55 40]]	0.95	0.42	0.58	0.05	0.8875	0.429	0.9156	0.6832	0.95
Decision Tree Reduced features (chi sq)	[[415 44] [29 66]]	0.9	0.69	0.31	0.1	0.9346	0.5658	0.9191	0.7994	0.9
Random Forest Reduced features (chi sq)	[[439 20] [37 58]]	0.96	0.61	0.39	0.04	0.9222	0.6144	0.9390	0.7834	0.96
SVC reduced features (chi sq)	[[442 17] [59 36]]	0.96	0.38	0.62	0.04	0.8822	0.4381	0.9208	0.6709	0.96
KNN reduced features (chi sq)	[[444 15] [57 38]]	0.97	0.4	0.6	0.03	0.8862	0.4707	0.9249	0.6836	0.97
LoR(anova f)	[[433 26] [47 48]]	0.94	0.51	0.49	0.06	0.9020	0.4970	0.9222	0.7243	0.94
DT(anova f)	[[429 30] [25 70]]	0.93	0.74	0.26	0.07	0.9449	0.6580	0.93975	0.8357	0.93
RF(anova f)	[[452 7] [31 64]]	0.98	0.67	0.33	0.02	0.9358	0.7424	0.9596	0.8292	0.98
SVC(anova f)	[[434 25] [52 43]]	0.95	0.45	0.55	0.05	0.8930	0.4573	0.91851	0.6990	0.95
KNN(anova f)	[[442 17] [53 42]]	0.96	0.44	0.56	0.04	0.8929	0.4949	0.9266	0.7025	0.96
LoR(mutual info)	[[436 23] [51 44]]	0.95	0.46	0.54	0.05	0.8952	0.4774	0.9217	0.7065	0.95
DT(mutual info)	[[428 31] [26 69]]	0.93	0.73	0.27	0.07	0.9427	0.6456	0.9375	0.8293	0.93
RF(mutual info)	[[449 10] [31 64]]	0.98	0.67	0.33	0.02	0.9354	0.7222	0.9563	0.8585	0.98
SVC(mutual info)	[[439 20] [53 42]]	0.96	0.44	0.56	0.04	0.8922	0.4764	0.9232	0.6992	0.96
KNN(mutual info)	[[448 11] [59 36]]	0.98	0.38	0.62	0.02	0.8836	0.4801	0.9275	0.7041	0.98
LoR(forward selection)	[[456 3] [93 2]]	0.99	0.02	0.98	0.01	0.8306	0.0578	0.9047	0.5072	0.99
DT(forward selection)	[[447 12] [29 66]]	0.97	0.69	0.31	0.03	0.9390	0.7245	0.9561	0.8342	0.97
RF(forward selection)	[[459 0] [38 57]]	1.0	0.6	0.4	0.0	0.9235	0.7443	0.9602	0.8	1.0

SVC(forward selection)	[[459 0] [41 54]]	1.0	0.57	0.43	0.0	0.918	0.7223	0.9572	0.7842	1.0
KNN(forward selection)	[[457 2] [48 47]]	1.0	0.49	0.51	0.0	0.9049	0.6509	0.9481	0.7451	1.0
LoR(rfe)	[[451 8] [85 10]]	0.98	0.11	0.89	0.02	0.8414	0.1867	0.9065	0.5439	0.98
DT(rfe)	[[445 14] [30 65]]	0.97	0.68	0.32	0.03	0.9368	0.7046	0.9528	0.8268	0.97
RF(rfe)	[[455 4] [36 59]]	0.99	0.62	0.38	0.01	0.9266	0.7270	0.9578	0.8061	0.99
SVC(rfe)	[[459 0] [46 49]]	1.0	0.52	0.48	0.0	0.9089	0.6846	0.9522	0.7578	1.0
KNN(rfe)	[[458 1] [56 39]]	1.0	0.41	0.59	0.0	0.8910	0.5946	0.9414	0.7041	1.0

B. For class 0

Model	Confusion matrix	TPR	TNR	FPR	FNR	Precision	MCC	F measure	ROC area	Recall
Logistic Regression	[[1072 239] [324 987]]	0.75	0.8	0.2	0.25	0.7868	0.5445	0.7658	0.858	0.75
Decision Tree	[[1224 87] [105 1206]]	0.92	0.9	0.07	0.08	0.928	0.84	0.9224	0.927	0.92
Random Forest	[[1145 166] [135 1176]]	0.88	0.8	0.12	0.12	0.8770	0.7582	0.8794	0.951	0.88
Support Vector Classifier	[[1075 236] [333 978]]	0.74	0.8	0.2	0.26	0.7900	0.542	0.7629	0.855	0.790
KNN	[[1290 21] [362 949]]	0.68	0.9	0.09	0.32	0.8872	0.6135	0.7726	0.897	0.68
Logistic Regression-Reduced features (chi sq)	[[40 55] [25 434]]	0.95	0.4	0.58	0.05	0.8875	0.4293	0.9156	0.847	0.95
Decision Tree Reduced features (chi sq)	[[66 29] [47 412]]	0.9	0.6	0.31	0.1	0.9346	0.5658	0.9191	0.787	0.9
Random Forest Reduced features (chi sq)	[[63 32] [11 448]]	0.96	0.6	0.39	0.04	0.9222	0.6144	0.9390	0.887	0.96
SVC reduced features (chi sq)	[[36 59] [17 442]]	0.96	0.3	0.62	0.04	0.8822	0.4381	0.9208	0.843	0.96

KNN reduced features (chi sq)	[[38 57] [15 444]]	0.97	0.4	0.6	0.03	0.8862	0.4707	0.9249	0.811	0.97
LoR(anova f)	[[48 47] [26 433]]	0.94	0.5	0.49	0.06	0.9020	0.4970	0.9222	0.842	0.94
DT(anova f)	[[69 26] [35 424]]	0.93	0.7	0.26	0.07	0.9449	0.6580	0.9397	0.842	0.93
RF(anova f)	[[56 39] [17 442]]	0.98	0.67	0.33	0.02	0.9358	0.7424	0.9596	0.885	0.98
SVC(anova f)	[[43 52] [25 434]]	0.95	0.45	0.55	0.05	0.8930	0.4573	0.9185	0.829	0.95
KNN(anova f)	[[42 53] [17 442]]	0.96	0.44	0.56	0.04	0.89292	0.4949	0.9266	0.788	0.96
LoR(mutual info)	[[44 51] [23 436]]	0.95	0.46	0.54	0.05	0.8952	0.4774	0.9217	0.846	0.95
DT(mutual info)	[[68 27] [31 428]]	0.93	0.73	0.27	0.07	0.9427	0.6456	0.9375	0.811	0.93
RF(mutual info)	[[61 34] [12 447]]	0.98	0.67	0.33	0.02	0.9354	0.7222	0.9563	0.858	0.98
SVC(mutual info)	[[42 53] [20 439]]	0.96	0.44	0.56	0.04	0.8922	0.4764	0.9232	0.845	0.96
KNN(mutual info)	[[36 59] [11 448]]	0.98	0.38	0.62	0.02	0.8836	0.4807	0.927	0.829	0.98
LoR(forward selection)	[[4 91] [2 457]]	0.99	0.02	0.98	0.01	0.8306	0.0578	0.9047	0.670	0.99
DT(forward selection)	[[65 30] [14 445]]	0.97	0.69	0.31	0.03	0.9390	0.7245	0.9561	0.847	0.97
RF(forward selection)	[[57 38] [1 458]]	1.0	0.6	0.4	0.0	0.9235	0.7443	0.9602	0.903	1.0
SVC(forward selection)	[[51 44] [1 458]]	1.0	0.57	0.43	0.0	0.918	0.7223	0.9572	0.783	1.0
KNN(forward selection)	[[43 52] [1 458]]	1.0	0.49	0.51	0.0	0.9049	0.6509	0.9481	0.825	1.0
LoR(rfe)	[[10 85] [8 451]]	0.98	0.11	0.89	0.02	0.8414	0.1867	0.90653	0.723	0.98

DT(rfe)	[[65 30] [14 445]]	0.97	0.68	0.32	0.03	0.9368	0.7046	0.9528	0.818	0.97
RF(rfe)	[[57 38] [0 459]	0.99	0.62	0.38	0.01	0.9266	0.7270	0.9578	0.899	0.99
SVC(rfe)	[[49 46] [0 459]]	1.0	0.52	0.48	0.0	0.9089	0.6846	0.9522	0.794	1.0
KNN(rfe)	[[39 56] [1 458]]	1.0	0.41	0.59	0.0	0.8910	0.5946	0.9414	0.811	1.0

C. For class 1

Model	Confusion matrix	TPR	TNR	FPR	FNR	Precision	MCC	F measure	ROC area	Recall
Logistic Regression	[[987 324] [239 1072]]	0.8	0.75	0.25	0.2	0.7585	0.5445	3.1914	0.858	0.8
Decision Tree	[[1206 105] [87 1224]]	0.92	0.91	0.08	0.07	0.9178	0.8459	0.9234	0.925	0.929
Random Forest	[[1176 135] [166 1145]]	0.87	0.88	0.11	0.12	0.8811	0.7582	0.8787	0.935	0.876
Support Vector Classifier	[[978 333] [236 1075]]	0.80	0.73	0.26	0.19	0.7539	0.5427	0.7781	0.855	0.804
KNN	[[949 362] [21 1290]]	0.91	0.68	0.32	0.09	0.7430	0.6135	0.8193	0.897	0.91
Logistic Regression-Reduced features (chi sq)	[[434 25] [55 40]]	0.42	0.95	0.05	0.58	0.6153	0.4293	0.5	0.847	0.42
Decision Tree Reduced features (chi sq)	[[412 47] [29 66]]	0.69	0.9	0.1	0.31	0.6	0.5658	0.6439	0.769	0.69
Random Forest Reduced features (chi sq)	[[448 11] [32 63]]	0.61	0.96	0.04	0.39	0.7435	0.6144	0.6705	0.874	0.61
SVC reduced features (chi sq)	[[442 17] [59 36]]	0.38	0.96	0.04	0.62	0.6792	0.4381	0.4864	0.844	0.38
KNN reduced features (chi sq)	[[444 15] [57 38]]	0.4	0.97	0.03	0.6	0.7169	0.4707	0.5135	0.811	0. 0.4
LoR(anova f)	[[433 26] [47 48]]	0.51	0.94	0.06	0.49	0.6486	0.4970	0.5680	0.842	0.51
DT(anova f)	[[424 35] [26 69]]	0.74	0.93	0.07	0.26	0.7	0.6580	0.7179	0.844	0.74
RF(anova f)	[[442 17] [39 56]]	0.67	0.98	0.02	0.33	0.9014	0.7424	0.7710	0.897	0.67
SVC(anova f)	[[434 25]	0.45	0.95	0.05	0.55	0.6323	0.4573	0.5276	0.831	0.45

	[52 43]]									
KNN(anova f)	[[442 17] [53 42]]	0.44	0.96	0.04	0.56	0.7118	0.4949	0.5454	0.788	0.44
LoR(mutual info)	[[436 23] [51 44]]	0.46	0.95	0.05	0.54	0.6567	0.4774	0.5432	0.846	0.46
DT(mutual info)	[[428 31] [27 68]]	0.73	0.93	0.07	0.27	0.69	0.6456	0.7076	0.836	0.73
RF(mutual info)	[[447 12] [34 61]]	0.67	0.98	0.02	0.33	0.8648	0.7222	0.7573	0.882	0.67
SVC(mutual info)	[[439 20] [53 42]]	0.44	0.96	0.04	0.56	0.6774	0.4764	0.5350	0.845	0.44
KNN(mutual info)	[[448 11] [59 36]]	0.38	0.98	0.02	0.62	0.7659	0.4801	0.5070	0.829	0.38
LoR(forward selection)	[[457 2] [91 4]]	0.02	0.99	0.01	0.98	0.4	0.0578	0.0399	0.670	0.02
DT(forward selection)	[[445 14] [30 65]]	0.69	0.97	0.03	0.31	0.8461	0.7245	0.7630	0.848	0.69
RF(forward selection)	[[458 1] [38 57]]	0.6	1.0	0.0	0.4	1.0	0.7443	0.7499	0.874	0.6
SVC(forward selection)	[[458 1] [44 51]]	0.57	1.0	0.0	0.43	1.0	0.7223	0.7248	0.787	0.57
KNN(forward selection)	[[458 1] [52 43]]	0.49	1.0	0.0	0.51	0.959	0.6509	0.6527	0.825	0.49
LoR(rfe)	[[451 8] [85 10]]	0.98	0.98	0.02	0.89	0.555	0.1867	0.1769	0.723	0.98
DT(rfe)	[[445 14] [30 65]]	0.68	0.97	0.03	0.32	0.8227	0.7046	0.7471	0.816	0.68
RF(rfe)	[[459 0] [38 57]]	0.62	0.99	0.01	0.38	0.9365	0.7270	0.7468	0.893	0.62
SVC(rfe)	[[459 0] [46 49]]	0.52	1.0	0.0	0.48	1.0	0.6846	0.0	0.793	0.52

KNN(rfe)	[[458, 1] [56 39]]	1.0	1.0	0.0	0.59	0.975	0.5946	0.5777	0.853	1.0

D. Weighted Average

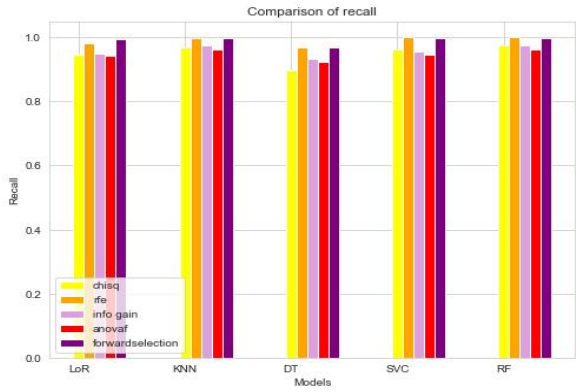
Model	TPR	TNR	FPR	FNR	Precision	MCC	F measure	ROC area	Recall
Logistic Regression	0.7719	0.77	0.22	0.214	0.7726	0.5445	0.978	0.8588	0.771
Decision Tree	0.9229	0.92	0.0770	0.073	0.9230	0.8459	0.922	0.9263	0.922
Random Forest	0.8790	0.87	0.120	0.114	0.8791	0.7582	0.8790	0.9432	0.879
Support Vector Classifier	0.7707	0.77	0.2292	0.217	0.7719	0.5427	0.7705	0.8558	0.770
KNN	0.798	0.79	0.2013	0.146	0.8151	0.6135	0.7959	0.8971	0.798
Logistic Regression- Reduced features (chi sq)	0.8555	0.51	0.4890	0.144	0.8408	0.4293	0.8443	0.8470	0.855
Decision Tree Reduced features (chi sq)	0.8682	0.73	0.2693	0.137	0.8772	0.5658	0.8719	0.7842	0.868
Random Forest Reduced features (chi sq)	0.897	0.66	0.3301	0.077	0.8916	0.6144	0.8929	0.8854	0.897
SVC reduced features (chi sq)	0.8628	0.47	0.5209	0.137	0.84742	0.4381	0.8463	0.8439	0.862
KNN reduced features (chi sq)	0.8700	0.49	0.5027	0.129	0.8572	0.4707	0.8544	0.8110	0.8700
LoR(anova f)	0.8682	0.58	0.4196	0.131	0.8586	0.4970	0.8615	0.8429	0.8682
DT(anova f)	0.9007	0.77	0.2292	0.11	0.90293	0.6580	0.90172	0.8424	0.9007
RF(anova f)	0.9314s	0.72	0.2729	0.101	0.9299	0.7424	0.9273	0.8879	0.931
SVC(anova f)	0.86101	0.53	0.4628	0.138	0.8483	0.4573	0.8514	0.8301	0.8610
KNN(anova f)	0.8736	0.53	0.4685	0.126	0.8618	0.4949	0.8612	0.7880	0.8736
LoR(mutual info)	0.8664	0.54	0.4533	0.133	0.8543	0.4774	0.8568	0.8468	0.8664
DT(mutual info)	0.897	0.76	0.2383	0.104	0.8993	0.6456	0.8981	0.8155	0.8971
RF(mutual info)	0.9259	0.72	0.2740	0.083	0.92331	0.7222	0.9222	0.8626	0.9259
SVC(mutual info)	0.8682	0.53	0.4696	0.131	0.8554	0.4764	0.8566	0.8453	0.8682
KNN(mutual info)	0.8736	0.48	0.5186	0.126	0.8634	0.4801	0.8554	0.8292	0.8736
LoR(forward selection)	0.82671	0.18	0.8121	0.167	0.7567	0.578	0.7564	0.6703	0.8267

DT(forward selection)	0.9259	0.74	0.2573	0.079	0.9231	0.7245	0.9230	0.8477	0.9259
RF(forward selection)	0.9314	0.66	0.3314	0.07	0.9366	0.7443	0.9241	0.8987	0.9314
SVC(forward selection)	0.9259	0.64	0.3575	0.081	0.9320	0.7223	0.9173	0.7840	0.9259
KNN(forward selection)	0.9097	0.58	0.4193	0.095	0.9142	0.6509	0.8974	0.8256	0.9097
LoR(rfe)	0.8321	0.25	0.7442	0.167	0.7923	0.1867	0.7814	0.7238	0.8321
DT(rfe)	0.9205	0.73	0.2668	0.0794	0.9172	0.7046	0.9176	0.8180	0.9205
RF(rfe)	0.9277	0.68	0.3154	0.068	0.9283	0.7270	0.9217	0.8986	0.9277
SVC(rfe)	0.9169	0.59	0.4011	0.083	0.7530	0.6846	0.78898	0.7940	0.9169
KNN(rfe)	0.89711	0.51	0.4887	0.102	0.9054	0.5946	0.8790	0.8183	0.8971

Metrics used to choose model: Class 1 True Positive Rate or Recall. If a person is going to die in the next 10 years but has been predicted to not (false negative) would be the worst case scenario and could have disastrous consequences. Thus, we focused on Recall as we want to reduce the number of false negatives. We picked the best model which gives maximum recall score. Since we want the least number of patients who are likely to die in the next 10 years to be predicted as not likely, we used class 1 Recall or True Positive Rate. We also looked at overall recall and weighted average of recall.

KNN after Recursive Feature Selection metrics	
Overall Recall	1.0
Class 1 Recall	1.0
Class 0 Recall	1.0
Weighted average for Recall	0.89

E. **Best model**: K Nearest Neighbors with features selected using Recursive Feature Elimination method. True positive rate or Recall for class 1 was highest for this model. It also had highest recall for class 0, overall recall score and a decently high weighted average for recall score. We also observed that the classifier's performance improved after reducing the feature set using RFE as compared to training the KNN model on the entire dataset.



The above figure shows comparison of overall Recall for all 5 classifiers for 5 feature selection methods.

F. Randomized Search

On our best model so far, we performed randomized search to further improve our model's performance. Randomized search takes a range of parameters for different attributes of the classifier and tries random combinations of these to get optimal res ults. We got best parameters as : {'weights': 'distance', 'p': 1, 'n_neighbors': 23}.
Based on these parameters, L1 norm/ Manhattan distance will be used to find 23 ne arest neighbors and get their majority vote. Neighbors will not be weighted equally, r ather their weight will be inversely proportional to their distance from the point to b e labelled. The closer the neighbor (lesser is the distance), higher is its influence.
We created a final K Nearest Neighbors Classifier model using these parameters on the reduced dataset, with features selected using Recursive Feature Elimination. Recall metrics after Randomized search:

True positive rate is 1.0
True positive rate for class 0 is 1.0
True positive rate for class 1 is 0.48
Weighted average of True positive rate/Recall is 0.91.
We observed that accuracy increased from 89.71% to 91.16% and weighted average of recall increased from 0.89 to 0.91. However, recall for class 1 decreased substantially.

Comparison of KNN before and after feature selection using Recursive Feature Elimination:

Model	TPR	TNR	FPR	FNR	Precisi on	MCC	F measure	ROC area	Recall
KNN before feature selection (Overall)	0.68	0.91	0.09	0.32	0.8872	0.6135	0.7726	0.7986	0.68
KNN after feature selection	1.0	0.41	0.59	0.0	0.8910	0.5946	0.9414	0.7041	1.0

using rfe (Overall)									
KNN before feature selection using rfe (Class 0)	0.68	0.9	0.09	0.32	0.8872	0.6135	0.7726	0.897	0.68
KNN after feature selection using rfe (Class 0)	1.0	0.41	0.59	0.0	0.8910	0.5946	0.9414	0.811	1.0
KNN before feature selection using rfe (Class 1)	0.91	0.68	0.32	0.09	0.7430	0.6135	0.8193	0.897	0.91
KNN after feature selection using rfe (Class 1)	1.0	1.0	0.0	0.59	0.975	0.5946	0.5777	0.853	1.0
KNN before feature selection using rfe (Weighted Average)	0.798	0.79	0.201	0.30	0.8151	0.6135	0.7959	0.8971	0.798
KNN after feature selection using rfe (Weighted Average)	0.8971	0.51	0.488	0.42	0.9054	0.5946	0.8790	0.8183	0.8971

10. Conclusion

We researched the performance of various classifiers using a myriad of performance metrics. We applied feature selection techniques and again evaluated model performance by applying the same classification algorithms on the reduced data. We chose best model as the one with highest recall. We learnt about the importance of different metrics based on the use case and that there is no single best metric which can always be used to give optimal results. We also learnt the importance of data reduction and observed that it can help improve model performance in many cases. Further, this work can be scaled to warn individuals to take precautionary health measures and eventually improve life expectancy.

11. Work Distribution:

1. Heena Rijhwani: Dataset Research, Data Pre-processing and Visualizations, Classification algorithms, three feature selection methods, all Performance metrics, Report.

Shalu Shalu: Two feature selection methods, Report tables.

12. References:

1. Class notes

2. https://scikit-learn.org/stable/supervised_learning.html#supervised-learning

3. https://www.kaggle.com/prashant111/comprehensive-guide-on-feature-selection