BITS Pilani

Pilani | Dubai | Goa | Hyderabad

A REPORT

ON

EXPLAINING THE PREDICTION OF MACHINE LEARNING CLASSIFIERS AND ALGORITHMS USING EXPLAINABLE AI

- NIMISH BANSAL (2021B5A71179P)

AT

DYSL - CT DRDO, CHENNAI

IN FULFILLMENT OF

PRACTICE SCHOOL – I

BIRLA INSTITUTE OF TECHNOLOGY AND SCIENCE

JULY 2023

ACKNOWLEDGEMENT

I would like to take this opportunity to thank the PS Division, BITS Pilani for giving me the opportunity to undertake my Practice School-I at DYSL - CT DRDO, Chennai. I would like to express my gratitude to **Mr. Glynn John** for his continuous guidance and encouragement towards the successful completion of the project. I would like to thank my parents for constantly encouraging me to work hard and complete the project. I am extremely grateful to **Mr. Manish Pratap Singh** (Director of DYSL - CT DRDO) for giving me the opportunity to work under his supervision and for his valuable insights on various machine learning algorithms and explainable AI techniques. The support provided by the director and all the instructors motivated me throughout the course and helped me complete it on time and gain some valuable insights.

- Nimish Bansal (2021B5A71179P)

CONTENTS

PS DETAILS

Name : Nimish Bansal

ID Number : 2021B5A71179P

Station : DYSL - CT DRDO, Chennai

Duration : 30 May 2023 to 21 July 2023

PS Mentor : Mr. Glynn John

Project Mentor : Mr. Manish Pratap Singh (Director of DYSL - CT DRDO)

Project: Heart Attack prediction using Explainable AI

ABOUT DYSL - CT DRDO

DRDO Young Scientist Laboratories (DYSLs) are five specialized research laboratories located in five different cities of India, (Bengaluru, Mumbai, Chennai, Kolkata and Hyderabad). It was inaugurated by the Prime Minister of India on 2nd January 2020. Each laboratory deals with a specific area of science like artificial intelligence, quantum technologies, cognitive technologies, asymmetric technologies, and smart materials.

The focus of the DYSL-CT is cognitive technologies. Cognitive radar, cognitive radio and cognitive surveillance systems are the main areas of research. The lab concentrates on the design and initial testing of such systems, which are then passed onto higher laboratories for further testing and mass production.

Cognitive radar is an advanced radar technology that improves the performance and capabilities of conventional radar systems by employing cognitive principles. It incorporates cognitive capabilities, such as adaptability, learning and decision-making into radar systems to enhance their detection, tracking and identification of targets in complex and dynamic environments.

Cognitive radio is an advanced wireless communication technology that intelligently employs radio frequency (RF) spectrum to increase spectrum utilization and efficiency. It is intended to address the rising problem of spectrum scarcity in which the demand for wireless communication services exceeds the frequency bands that are available.

ABOUT THE PROJECT

Project Title - Implementation of various machine learning algorithms and explaining them with the help of explainable AI algorithms.

The project is based on understanding the importance of trust in machine learning algorithms and models in order to deploy them into use in the real world. In order to use a new model or new technology, the user must be able to trust the technology and its predictions.

Explainable AI is a new and emerging field of AI that helps in developing trust in the model and explains the predictions made by the model. In the project, various machine learning algorithms such as **Decision Tree, Random Forest, Support Vector Machine, K - Nearest Neighbors, Naive Bayes** were implemented on a dataset which was obtained from **kaggle** and then the predictions made by them were explained using various explainable AI models such as **LIME and SHAP.**

The test and training accuracy of various models were calculated and were optimized to achieve best possible results. In the end, the models were compared in their functioning and their accuracy in making predictions.

ABOUT MACHINE LEARNING

Machine learning is a subfield of artificial intelligence (AI) that concentrates on the development of algorithms and statistical models and makes it possible for computer systems to learn and make predictions using various techniques. The primary objective of machine learning is to enable computers to learn and analyze data, recognise patterns and make accurate predictions based on this understanding.

Some of the types of machine learning approaches are:

Supervised learning:
In Supervised learning, the machine learning model is trained using labeled data, in which the input data is paired with output labels. The model learns to predict unlabeled data by generalizing from labeled training examples.

Unsupervised learning:
Unsupervised learning involves training models on unlabeled data. The model's goal is to identify patterns, structures or relationships within the data without any predefined labels or outputs.

Semi-supervised learning:
This methodology combines supervised and unsupervised learning. The model is trained using a small quantity of labeled data and a large amount of unlabeled data. The labeled data guides the learning process, whereas the unlabeled data facilitates the discovery of new patterns.

Reinforcement learning:
In reinforcement learning, an agent learns to interact with an environment and make decisions based on trial and error. The agent receives a reward based on its actions, which allows it to learn optimal behavior on its own in an unknown environment.

MACHINE LEARNING ALGORITHMS (CLASSIFIERS)

The following machine learning algorithms were used to make the predictions of the project:

1. Decision Tree
2. Random Forest
3. Support Vector Machine (SVM)
4. K-Nearest Neighbors (KNN)
5. Naive Bayes

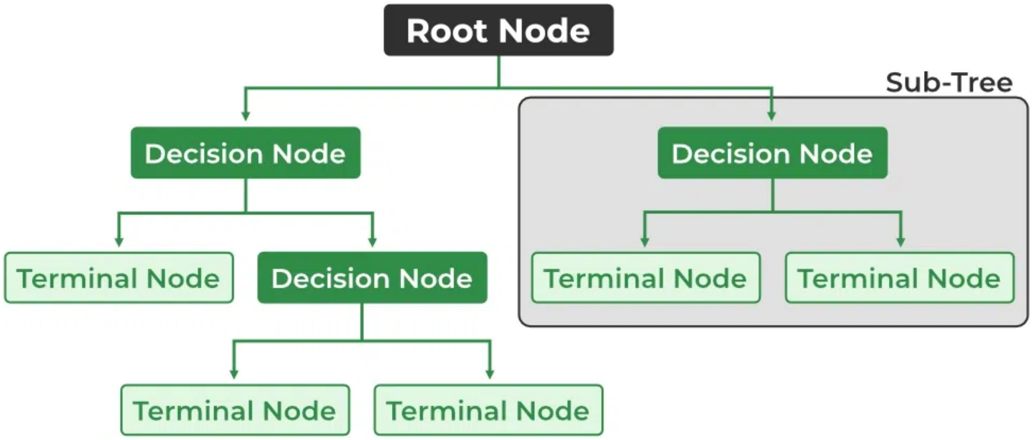
DECISION TREE

A decision tree is an algorithm for supervised machine learning that is used for both classification and regression models. A decision tree is a flowchart-like tree structure in which each internal or decision node represents a classification feature, branches represent decisions, and leaf nodes represent the classifier's output.

ROOT NODE: The root node is the topmost decision node or the parent node, where all the data is stored.
DECISION NODE: A node that has two or more branches.
LEAF NODE: A leaf node is a node that carries the classification or the decision and has homogeneous data.
ENTROPY: It is a measure of randomness or variability in the data set.
INFORMATION GAIN: It is the measure of the decrease in the randomness after the data set is split into different nodes based on the threshold.

The entropy for a subset of the original dataset which is having K number of classes and for the ith node is given by:

$$H_i = - \Sigma^{n}_{k\in K} p(i,k) \log_2 p(i,k)$$



GINI IMPURITY: Gini Impurity is a score that evaluates the accuracy of a classification division. The Gini Impurity score ranges from 0 to 1, where 0 indicates that all observations belong to the same class and 1 represents an arbitrary distribution of the elements.

The complete decision tree algorithm can be explained through the following steps:

Step 1: The complete dataset is available at the root node which is put into the decision tree for classification.
Step 2: The best attribute feature from the dataset is found using Attribute Selection Measure (ASM) technique.
Step 3: The root node is divided into subsets that contain the best possible values for the best features.
Step 4: A decision node is generated based on the classification, which contains the best attribute.
Step 5: Repeatedly new decision trees are made using the subsets of the dataset created in step 3 and the process is continued until the data is classified into separate classes and achieves homogeneity.

ADVANTAGES:

1. Less data modification and cleaning is required as compared to others.
2. It helps to consider all possible features and predict the best possible results.

DISADVANTAGES:

1. The decision tree contains many layers, which leads to complexity.
2. Decision tree algorithm has an issue of overfitting of the model which leads to a drop in the testing set accuracy.

A decision tree can be used to help build automatic prediction models, which have applications in machine learning, data science, data mining, and statistics.

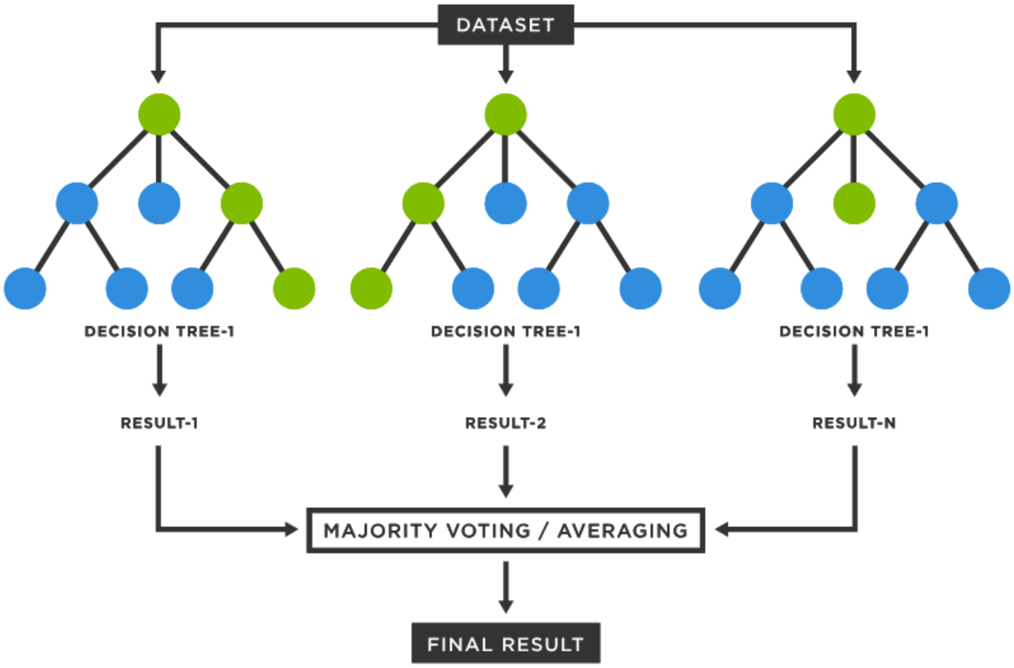
RANDOM FOREST

A random forest is an algorithm for supervised machine learning that combines several decision tree models. It can be used to solve problems involving classification and regression. It is based on ensemble learning, which evaluates the predictions of multiple decision trees and selects the output based on majority. A random forest algorithm comprises multiple parallel decision trees.
The 'forest' of numerous trees generated by the random forest algorithm is trained using the bagging technique.

BAGGING (BOOTSTRAP AGGREGATION):
Bagging is an ensemble algorithm that improves the accuracy of machine learning algorithms. It predicts by taking the average of the output from many decision trees. Increasing the number of trees increases the precision of the outcome and reduces the issue of overfitting.

BOOSTING:

It creates sequential models and combines predictions from all the models such that the final model has the highest accuracy.



FEATURES:

1. Accuracy is greater than Decision tree as it is a combination of many decision trees.
2. Allows to handle the missing data instead of modifying it.
3. Helps in solving the issue of overfitting of the model.

SUPPORT VECTOR MACHINE

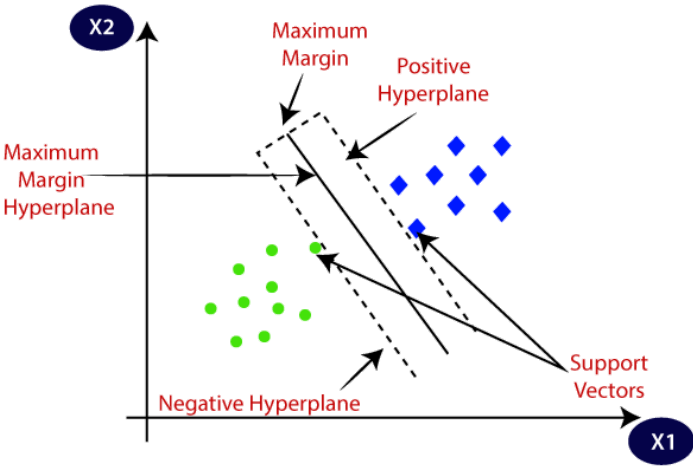
❖ KEY TERMS:

SUPPORT VECTOR:

They are the vectors that are in close proximity to the hyperplane and influence its position. These vectors are called sustain vectors because they sustain the hyperplane.

HYPERPLANE:

It is a linear decision boundary that separates two different classes.



MARGIN:

It refers to the distance between the hyperplane and the support vectors.

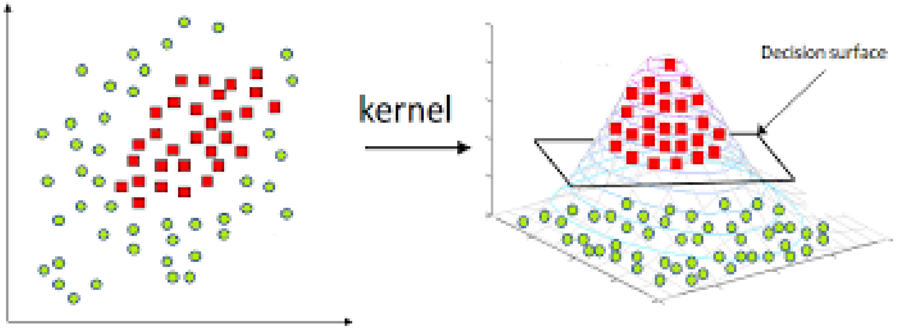
Support Vector Machines (SVM) aim to find an optimal hyperplane that maximizes the margin. It aims to do so, because then the two classes are properly classified and there wouldn't be any confusion while classifying a new data point.

HANDLING NON - LINEAR DATA USING KERNEL TRICK:

In the case of non-linearly separable data, it may be impossible to identify an optimal hyperplane in the initial feature space. In these situations, SVM employs the kernel trick.

The kernel trick permits SVM to map the data implicitly to a higher-dimensional space where a linear decision boundary may be feasible. SVM operates as if it is in a higher-dimensional space without explicitly computing the transformation by computing the inner product between pairs of data points in the original feature space using a kernel function.

Thus, by implementing the kernel trick, SVM can find an optimal hyperplane in the higher-dimensional feature space, which corresponds to a non-linear decision boundary in the original feature space.



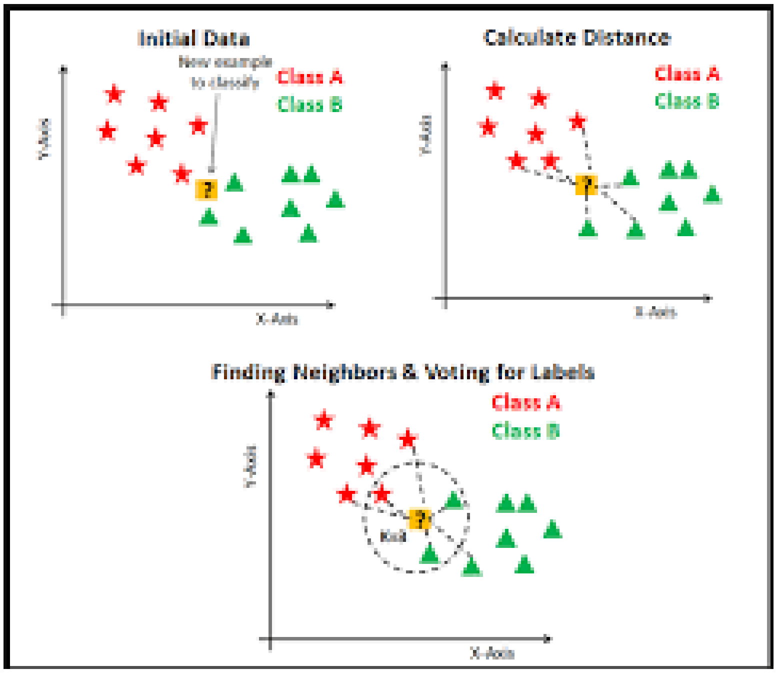
K - NEAREST NEIGHBORS

K-Nearest Neighbors (KNN) operates based on the idea that similar data points belong to the same class. When a new data point needs to be classified, KNN finds its k nearest neighbors in the training data and makes predictions based on their labels.

CHOOSING k:

In the KNN algorithm, the k value specifies the number of neighbors that will be examined to determine the classification of a new data point. For instance, if k = 1, the data point will be designated the same class as its nearest neighbor. Defining k can be a delicate balancing act, as different values can result in either overfitting or underfitting.

In order to avoid ties during classification, it is recommended that k be an odd number, and cross-validation techniques can aid in selecting the optimal k for the dataset.



METRIC FOR MEASURING THE PROXIMITY:

In order to determine which data points are closest to a given data point, the distance between the new data point and the other data points need to be calculated.

The following metrics are used to calculate the proximity:

Euclidean distance:

$$d(x, y) = \sqrt{\sum_{i=1}^{n} (y_i - x_i)^2}$$

Manhattan distance:

$$d(x, y) = \sum_{i=1}^{m} \left| x_i - y_i \right|$$

While classifying, the class label is assigned to the new data point based on the majority between the classes among the k neighbors.

NAIVE BAYES

The Naive Bayes classifier is based on Bayes Theorem. Naive Bayes classifier has the following assumptions :

1. Every feature is independent of other features.
2. Every feature has an equal contribution to the prediction.

Bayes Theorem : Bayes Theorem finds the probability of an event occurring given that another event has already occurred.

$$P(A|B) = \frac{P(B|A)\, P(A)}{P(B)}$$

1. P(A) is termed as the prior probability.
2. P(A|B) is termed as the posterior probability.
3. P(B|A) is termed as the likelihood probability.

Bayes' theorem can be applied in the following manner:

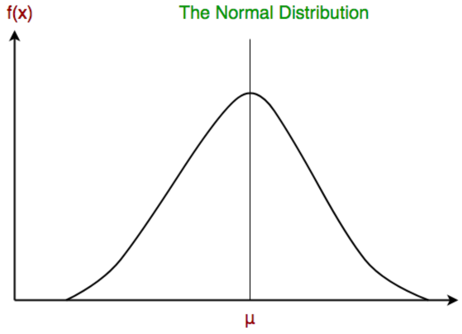
$$P(y|X) = \frac{P(X|y)\,P(y)}{P(X)}$$

where, y is a class variable and X is a feature vector (of size *n*).

So the final equation becomes :

$$P(y|x1, x2,\dots\dots, xn) = \frac{P(x1|y)P(x2|y).....P(xn|y)P(y)}{P(x1)P(x2)......P(xn)}$$

Gaussian Naive Bayes Classifier: In Gaussian Naive Bayes, it is presumed that continuous values associated with each feature follow a normal distribution. When plotted, a bell-shaped curve symmetric about the mean of the feature values is obtained.



WORKING OF NAIVE BAYES ALGORITHM:

1) The data set is converted into a frequency table.
2) This is in turn converted to a likelihood table by finding the probabilities.
3) Then the Bayes theorem is used to calculate the posterior probability of each class.
4) The class with the highest posterior probability is said to be the outcome of the prediction.

Advantages:

1. It quickly predicts the class of a test data set. So it performs well in multi-class prediction.
2. When the assumption of independence of every feature holds, the classifier performs better compared to other machine learning models.

Disadvantages:

1. If a categorical variable contains a category that was not observed during training, the model will allocate zero probability and will be incapable of making a prediction. This is known as the "Zero Frequency"
2. Another disadvantage of this algorithm is that it assumes that every feature is independent of the other. In real life, it is almost impossible for this to happen.

ABOUT EXPLAINABLE AI

Explainable AI is a collection of tools and frameworks that enable users to comprehend and interpret the predictions made by various machine learning models, thereby enhancing their trust in them.

The aim of Explainable AI (XAI) is to create machine learning techniques that develop more explainable models that maintain a high level of learning performance. It must also enable users to understand, trust, and manage new emerging machine learning and artificial intelligence models.

To achieve this objective, machine learning techniques are modified such that the predictions will be explained, the strengths and limitations of the model will be identified, and their future behavior will be predicted.

Benefits of XAI :

1. Helps in developing Trust in AI models
2. Helps to reduce biasing in AI models
3. Provides insights against Adversarial Attacks

EXPLAINABLE AI ALGORITHMS

The following explainable AI algorithms were used to explain the predictions made by the machine learning algorithms:

1. LIME
2. SHAP

LIME

LIME stands for **Local Interpretable Model-agnostic Explanations.** It is a visualization technique that helps in explaining individual predictions.

Local explanations:
It means that LIME explains the predictions of a single data point such that it is locally faithful in the environment it is being explained.

Interpretable:
It refers to the ability to understand and explain the predictions of a complex model in a human-understandable manner.

Model-agnostic:
It means that LIME can give explanations for any supervised learning model by treating it as a 'black box' separately. This means that LIME can be implemented on any model.

WORKING OF LIME:

LIME operates under the assumption that every complex model on a local scale is linear. LIME attempts to implement a simplified model centered on a single data point that mimics the local behavior of the global model. This basic model can then be used to explain the complex model's predictions locally.

EXPLANATION OF THE LIME ALGORITHM:

1. The data point to be explained is perturbed n times to create replicated data with slight value modifications. This perturbed data is a fake data created around the data point by LIME to build the local linear model.
2. The outcome is predicted for the perturbed data.
3. The distance from each perturbed data point to the original observation is calculated.
4. The distance is then converted to a similarity score.
5. From the perturbed data, m features are selected that best describe the predictions.
6. A simple model is fitted to the perturbed data for the selected features.
7. The feature weights of the simple model are the explanations of the observation.

SHAP

SHAP stands for **Shapley Additive Explanations**. This method aims to explain the prediction of an instance/observation by computing the contribution of each feature to the prediction.

Shapley values can help you in:

1. Global model interpretability
2. Local interpretability

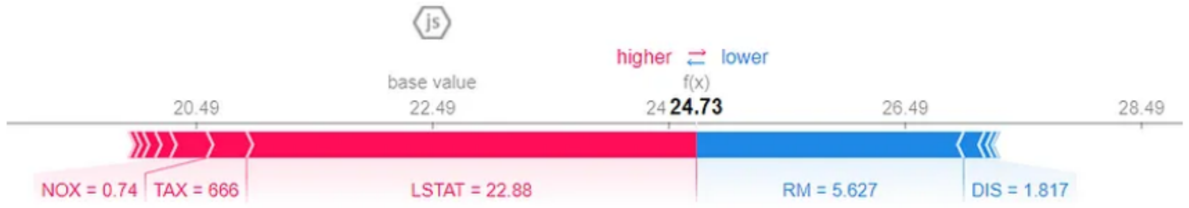
Different SHAP model explainers which are used to explain the predictions are:

TreeExplainer:
This method implements TreeSHAP algorithm and is useful for tree based algorithms such as Decision Tree, Random Forest etc.

DeepExplainer:
This method implements DeepLIFT algorithm and is used for deep learning models.
LinearExplainer:
As the name itself says, this method is ideal for linear models.
KernelExplainer:
This method is a model-agnostic method. Means it can be used to explain any model — linear models, tree models or deep learning models.

<u>FORCE PLOT:</u>

Force plots are used to explain the prediction of individual data points. One example of a Force plot is :



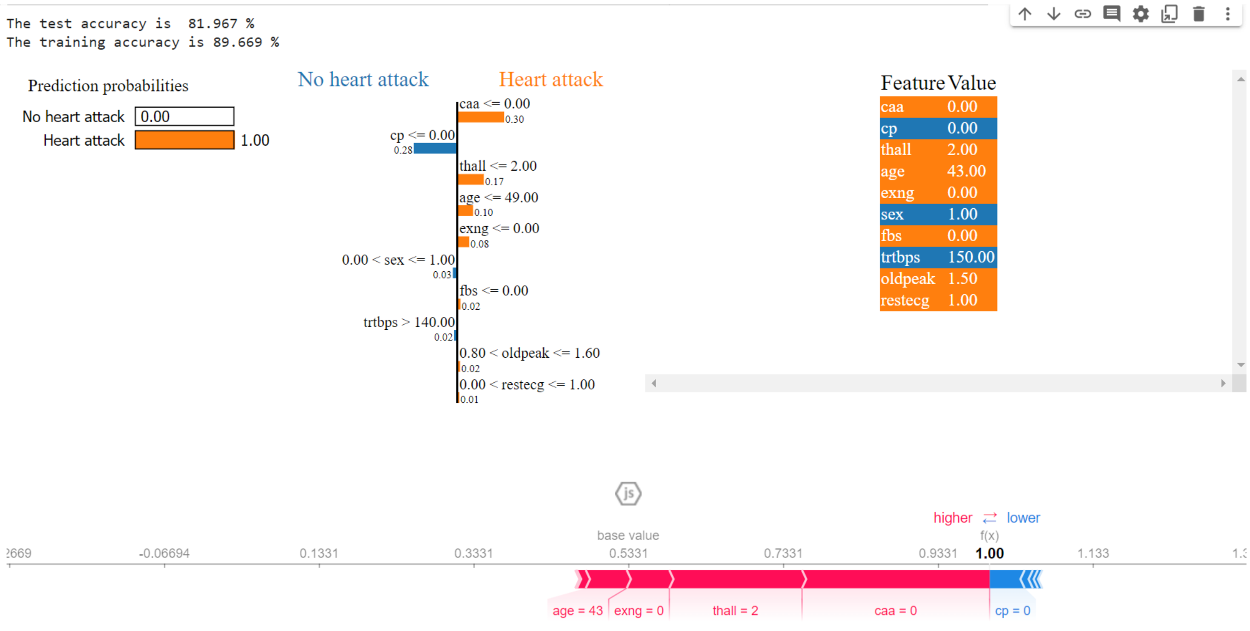
Base value is defined as the mean prediction over the entire testing dataset. It is the value that would be predicted in the absence of any features for the current output.

RESULTS

Abbreviations used in the dataset:

cp - chest pain type
trtbps - resting blood pressure
chol - serum cholesterol
fbs - fasting blood sugar
restecg - resting electrocardiographic results
thalachh - maximum heart rate achieved
exng - exercise induced angina
oldpeak - ST depression induced by exercise relative to rest
slp - slope of peak exercise ST segment
caa - number of major vessels colored by Fluoroscopy
thall - Thallium Stress Test

DECISION TREE:

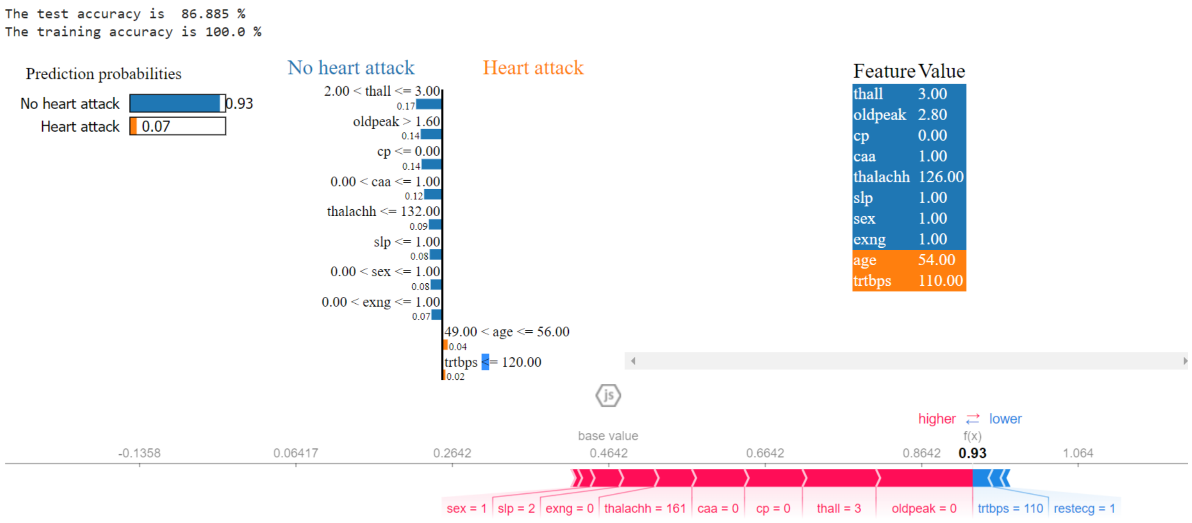


```
The test accuracy is  81.967 %
The training accuracy is 89.669 %
```

Feature	Value
caa	0.00
cp	0.00
thall	2.00
age	43.00
exng	0.00
sex	1.00
fbs	0.00
trtbps	150.00
oldpeak	1.50
restecg	1.00

The test and training accuracy were approximately 82% and 90% respectively for the Decision Tree model. The model predicted the output “Heart attack” for the particular test case. According to the LIME algorithm, the features which influenced this prediction the most are caa, thall, age in decreasing order. According to SHAP algorithm, this prediction was due to caa, thall, exng in the decreasing order.

RANDOM FOREST:

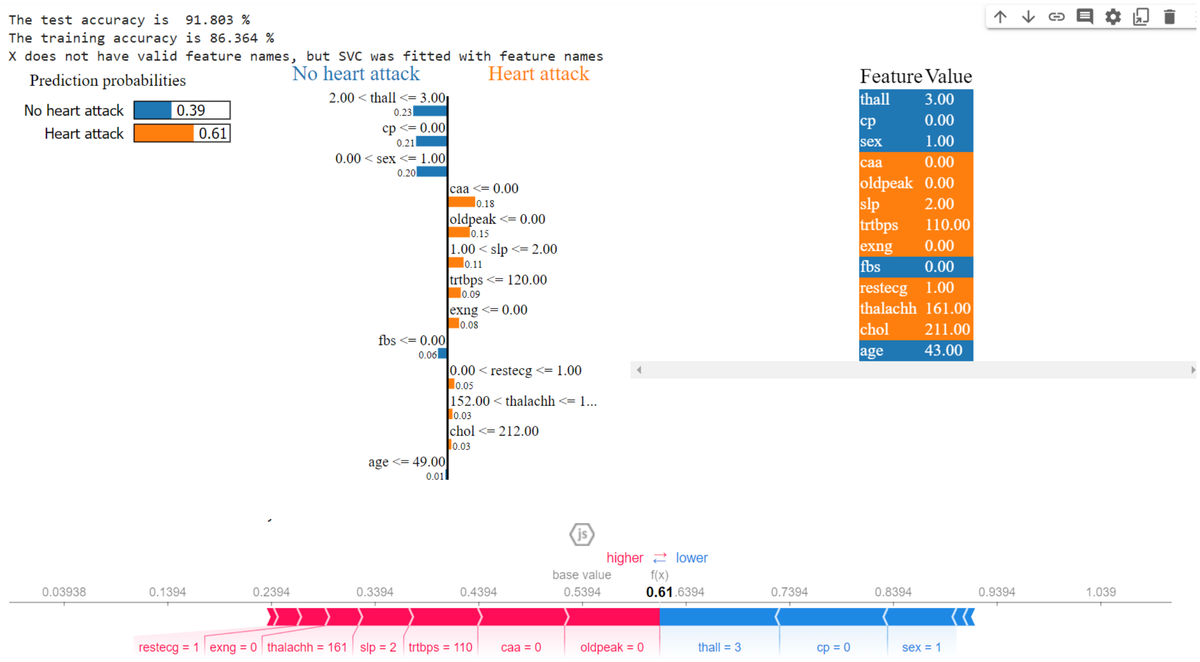


```
The test accuracy is  86.885 %
The training accuracy is 100.0 %
```

Feature	Value
thall	3.00
oldpeak	2.80
cp	0.00
caa	1.00
thalachh	126.00
slp	1.00
sex	1.00
exng	1.00
age	54.00
trtbps	110.00

The Random Forest model makes the predictions with an accuracy of approximately 87%**.** The prediction made by the model is “No heart attack” which is based on the features thall, oldpeak, cp, etc.

SUPPORT VECTOR MACHINE:

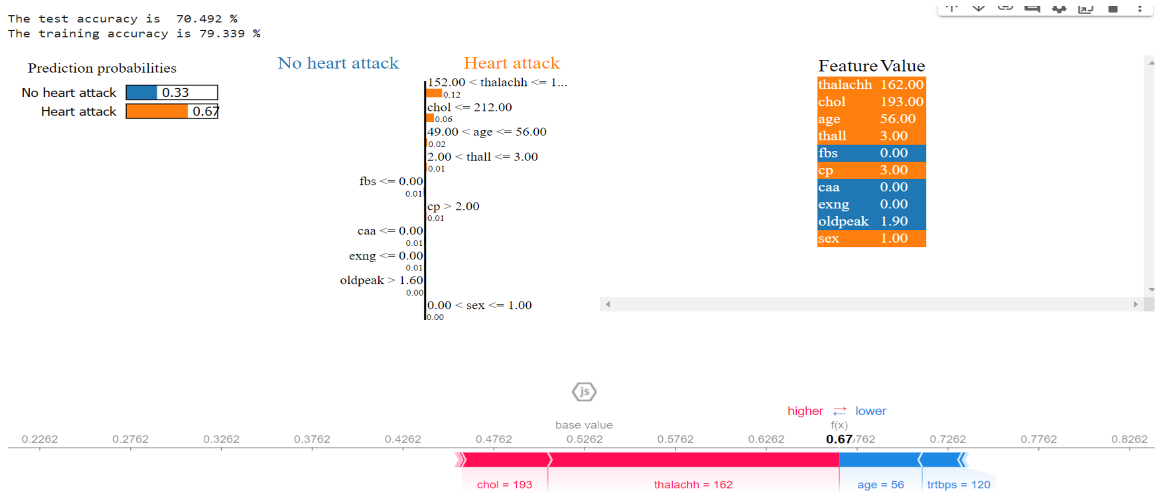


```
The test accuracy is  91.803 %
The training accuracy is 86.364 %
X does not have valid feature names, but SVC was fitted with feature names
```

Feature	Value
thall	3.00
cp	0.00
sex	1.00
caa	0.00
oldpeak	0.00
slp	2.00
trtbps	110.00
exng	0.00
fbs	0.00
restecg	1.00
thalachh	161.00
chol	211.00
age	43.00

The test and training accuracy came out to be approximately 92% and 86% respectively for Support Vector Machine. The prediction came out to be “Heart attack” in this case which was supported by the features such as caa, old peak and slp.

K - NEAREST NEIGHBORS:

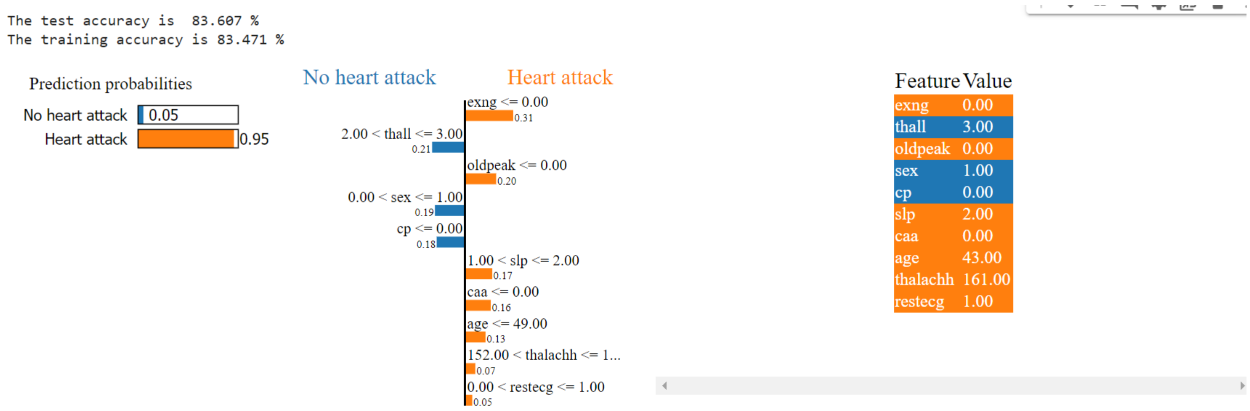


```
The test accuracy is  70.492 %
The training accuracy is 79.339 %
```

Feature	Value
thalachh	162.00
chol	193.00
age	56.00
thall	3.00
fbs	0.00
cp	3.00
caa	0.00
exng	0.00
oldpeak	1.90
sex	1.00

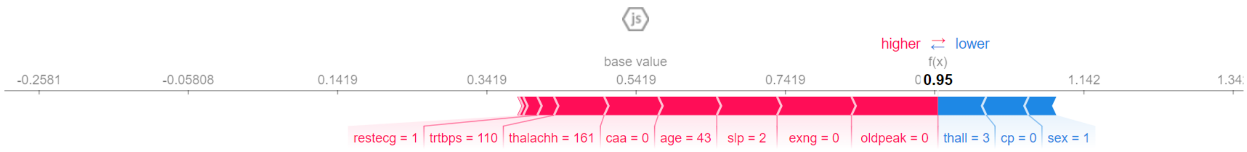
The predictions made by KNN were accurate with a test accuracy of approximately 70%. The prediction made by the model in the case above is “Heart attack” which is based on the features such as thalachh, chol, age etc.

NAIVE BAYES:



```
The test accuracy is  83.607 %
The training accuracy is 83.471 %
```

Feature	Value
exng	0.00
thall	3.00
oldpeak	0.00
sex	1.00
cp	0.00
slp	2.00
caa	0.00
age	43.00
thalachh	161.00
restecg	1.00



Naive Bayes performed quite well on the test dataset with a test accuracy of 84%. “Heart attack” was predicted by the model with a probability of 0.95. LIME was further used to explain the prediction which highlighted the key features affecting the output such as exng, oldpeak, slp etc.

CONCLUSION

It can be concluded that explainable AI plays a crucial role in developing trust in various machine learning models so that they can be deployed for real world applications. The users can now get a clear idea as to which factors influence their model and can make any desired changes.

Throughout the project, various machine learning algorithms and their predictions were observed and among all the models, Support Vector Machine (SVM) had the best performance in predicting the correct output with a test accuracy of approximately **91 percent**. The predictions made by the Naive Bayes algorithm were also quite successful with a test accuracy of **83 percent.**

Random forest outperformed the Decision tree model which is clearly evident as it is a combination of many decision trees and reduces the issue of overfitting of the model.

The predictions of various models were explained more accurately by LIME as it gives a clear understanding of various features used by the model in making the predictions and their contribution in making the final prediction.

BIBLIOGRAPHY

1. https://www.ibm.com/topics/machine-learning
2. https://www.lucidchart.com/pages/decision-tree
3. https://www.upgrad.com/blog/guide-to-decision-tree-algorithm/
4. https://www.geeksforgeeks.org/decision-tree/
5. https://towardsdatascience.com/understanding-random-forest-58381e0602d2
6. https://www.section.io/engineering-education/introduction-to-random-forest-in-machine-learning/
7. https://www.simplilearn.com/tutorials/machine-learning-tutorial/random-forest-algorithm
8. https://www.javatpoint.com/machine-learning-support-vector-machine-algorithm
9. https://www.ibm.com/topics/knn
10. https://www.geeksforgeeks.org/naive-bayes-classifiers/
11. https://www.analyticsvidhya.com/blog/2017/09/naive-bayes-explained/
12. https://www.darpa.mil/program/explainable-artificial-intelligence
13. https://www.techtarget.com/whatis/definition/explainable-AI-XAI
14. https://www.geeksforgeeks.org/introduction-to-explainable-aixai-using-lime/
15. https://analyttica.com/demystifying-lime-xai-through-leaps/
16. https://www.datacamp.com/tutorial/explainable-ai-understanding-and-trusting-machine-learning-models
17. https://www.kaggle.com/code/bextuychiev/model-explainability-with-shap-only-guide-u-need
18. https://pub.towardsai.net/how-to-explain-your-machine-learning-predictions-with-shap-values-a8332c3e5a11