DIABETIES DETECTION

A report submitted in partial fulfillment of the requirements for the Award of Degree of

BACHELOR OF TECHNOLOGY

in

ARTIFICIAL INTELLIGENCE & MACHINE LEARNING

By

JAYASRI GATLA

Reg. No. : 23B95A6116

Submitted to

Sri R Shiva Shankar

(Assistant Professor)

& **Dr DNSB Kavitha**

(Assistant Professor)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

S.R.K.R. ENGINEERING COLLEGE(A)

SRKR MARG, CHINNA AMIRAM, BHIMAVARAM-534204, A.P

(Recognized by A.I.C.T.E New Delhi) (Accredited by NBA & NAAC)

(Affiliated to JNTU, KAKINADA)

SAGI RAMA KRISHNAM RAJU ENGINEERING COLLEGE

(Autonomous)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE

This is to certify that the ML-CONNECT Report titled “DIABETIES DETECTION” is the bonafide work done by Ms.JAYASRI GATLA bearing Register Number :23B95A6116 in the second year first semester at SRKR Engineering College, Bhimavaram from 25th Novemeber 2023 to 26th November 2023 in partial fulfillment of the requirements for the award of the degree of Bachelor of Technology in Computer Science and Engineering.

Dr V Chandra Sekhar	**Sri R Shiva Shankar**	**Dr DNSB Kavitha**
(Professor and Head)	**(Assistant Professor)**	**(Assistant Professor)**

Table of Contents

Abstract

This project aims to develop an effective machine learning model for the early detection of diabetes, leveraging a dataset with essential health parameters. The study focuses on employing predictive analytics to enhance diabetes risk assessment using a dataset comprising the key parameters

This project deals with analyzing specific health data related to pregnancies, glucose levels, blood pressure, skin thickness, insulin, BMI, hereditary factors, and age to predict the likelihood of diabetes. They're using machine learning techniques like decision trees, support vector machines, and neural networks to train a model for this prediction.

The goal is to improve diabetes detection accuracy, potentially allowing for earlier intervention and better outcomes for patients. This research could offer valuable insights into healthcare analytics and personalized medicine, helping healthcare professionals make more informed decisions about preventing and managing diabetes.

Chapter 1

1.1 What are the different types of Machine Learning?

How Machine Learning Works?

There are three types of machine learningSupervised learning, which trains a model on known input and output data so that it can predictfuture outputs, and Unsupervised learning, which finds hidden patterns or intrinsic structures ininput data.

<u>Supervised Machine Learning:</u>

Supervised machine learning builds a model that makes predictions based on evidence in thepresence of uncertainty. A supervised learning algorithm takes a known set of input data and knownresponses to the data (output) and trains a model to generate reasonable predictions for the responseto new data. Use supervised learning if you have known data for the output you are trying to predict.Supervised learning uses Regression and Classification techniques to develop predictive models.

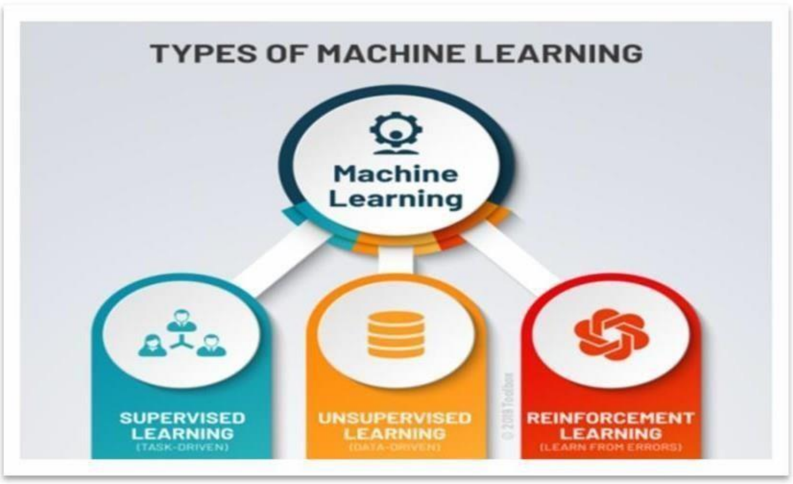

TYPES OF MACHINE LEARNING
Machine Learning
SUPERVISED LEARNING (TASK-DRIVEN)
UNSUPERVISED LEARNING (DATA-DRIVEN)
REINFORCEMENT LEARNING (LEARN FROM ERRORS)

Fig 1.1 Types of Machine Learning

Regression techniques predict continuous responses - for example, changes in temperature orfluctuations in power demand. Typical applications include electricity load forecasting andalgorithmic trading.Use regression techniques if you are working with a data range or if the nature of your response is areal number, such as temperature or the time until failure for a piece of equipment.Common regression algorithms include linear model, nonlinear model, stepwise regression,Gradient Descent Regression, Support Vector Regression, Ridge and Lasso Regressions.Classification techniques predict discrete responses - for example, whether an email is genuine orspam, or whether a tumour is cancerous or benign. Classification models classify input data into categories. Typical applications include medical imaging, speech recognition, and credit scoring.Use classification if your data can be tagged, categorized, or separated into specific groups or classes. For example, applications for hand-writing recognition use classification to recognize letters and numbers. In image processing and computer vision, unsupervised pattern recognitiontechniques are used for object detection and image segmentation.Common algorithms for performing classification include support vector machine (SVM), boostedand bagged decision trees, k-nearest neighbour, Naïve Bayes, discriminant analysis, logisticregression, and neural networks.Using Supervised Learning to Predict Heart Attacks: Suppose clinicians want to predict whethersomeone will have a heart attack within a year. They have data on previous patients, including age,weight, height, and blood pressure. They know whether the previous patients had heart attackswithin a year. So, the problem is combining the existing data into a model that can predict whether anew person will have a heart attack within a year.

Unsupervised Learning:

Unsupervised learning finds hidden patterns or intrinsic structures in data. It is used to drawinferences from datasets consisting of input data without labelled responses.Clustering is the most common unsupervised learning technique. It is used for exploratory dataanalysis to find hidden patterns or groupings in data. Applications for cluster analysis include genesequence analysis and market research.For example, if a cell phone company wants optimize the locations where they build cell phonetowers, they can use machine learning to estimate the number of clusters of people relying on theirtowers. A phone can only talk to one tower at a time, so the team uses clustering algorithms todesign the best placement of cell towers to optimize signal reception for groups, or clusters, of theircustomers.Common algorithms for performing clustering include k-means and k-medoids, Apriori algorithms,hierarchical clustering, Gaussian mixture models and hidden Markov models.

1.2 Benefits of Using Machine Learning in Candidate Placement Prediction.

Exploring the Advantages and Disadvantages of Machine Learning:

- When it comes to learning technology, we should be aware of the pros and cons of that technology. The reason is so that we can understand the capabilities of that subject.
- That is exactly what we are doing here. Understanding the advantages and disadvantages of Machine Learning will help us to unlock many doors.
- The advantages of Machine Learning are vast. It helps us to create ways of modernizing technology. The disadvantages of Machine Learning tell us its limits and side effects. This helps us to find different innovative ways to reduce these problems.

Advantages of Machine Learning:

- **Automation of Everything:** Machine Learning is responsible for cutting the workload and time. By automating things, we let the algorithm do the hard work for us. Automation is now being done almost everywhere. The reason is that it is very reliable. Also, it helps us to think more creatively. Due to ML, we are now designing more advanced computers. These computers can handle various Machine Learning models and algorithms efficiently. Even though automation is spreading fast, we still don’t completely rely on it. ML is slowly transforming the industry with its automation.

- **Wide Range of Applications:** ML has a wide variety of applications. This means that we can apply ML on any of the major fields. ML has its role everywhere from medical, business, banking to science and tech. This helps to create more opportunities. It plays a major role in customer interactions. Machine Learning can help in the detection of diseases more quickly. It is helping to lift up businesses. That is why investing in ML technology is worth it.

- **Scope of Improvement:** Machine Learning is the type of technology that keeps on evolving. There is a lot of scope in ML to become the top technology in the future. The reason is it has a lot of research areas in it. This helps us to improve both hardware and software. In hardware, we have various laptops and GPU. These have various ML and Deep Learning networks in them. These help in the faster processing power of the system. When it comes to software, we have various UI and libraries in use. These help in designing more efficient algorithms.
- **Efficient Handling of Data:** Machine Learning has many factors that make it reliable. One of them is data handling. ML plays the biggest role when it comes to data currently. It can handle any type of data. Machine Learning can be multidimensional or different types of data. It can process and analyse these data those normal systems can't. Data is the most important part of any Machine Learning model. Also, studying and handling of data is a field.
- **Best for Education and Online Shopping:**
  ML would be the best tool for education in the future. It provides very creative techniques to help students study.
  Recently in China, a school has started to use ML to improve student focus. In online shopping, the ML model studies your searches. Based on your search history, it would provide advertisements. These will be about your search preferences in previous searches. I

2.0 Our Dataset content:

We focuses on creating a powerful machine learning tool to spot diabetes early. They're using important health data like pregnancies, glucose levels, blood pressure, skin thickness, insulin, BMI, hereditary factors, and age to predict who might get diabetes. To do this, they're teaching computers using decision trees, support vector machines, and neural networks.
The main aim is to make finding diabetes more accurate. If successful, this could mean catching diabetes sooner and giving patients better care. This research could teach us a lot about healthcare data and how to personalize treatments, which could help doctors make smarter choices in preventing and treating diabetes.

- Pregnancies ○
  Glucose levels
  ○ Blood
  pressure ○
  Skin thickness

- Insulin ○ BMI
  ○ Age
- Diabetes
  Pedigree
  Function

3.0 ML Modelling and Results

3.1 Your Problem of Statement

Diabetes mellitus is a chronic metabolic disorder characterized by elevated blood glucose levels, and its prevalence is steadily increasing globally. Early detection and intervention play a crucial role in managing diabetes and preventing complications. However, existing healthcare systems often face challenges in identifying individuals at risk of developing diabetes, leading to delayed diagnoses and suboptimal patient outcomes.

The need for an accurate and efficient diabetes prediction system is evident to address this gap in healthcare. Current predictive models often lack the integration of diverse data sources, fail to incorporate advanced machine learning techniques, and struggle with real-time applicability. Furthermore, the complexity of diabetes and its multifactorial nature, involving genetic, lifestyle, and clinical factors, necessitates a comprehensive and integrated approach for prediction.

To address these challenges, there is a pressing need for the development of an intelligent Diabetes Prediction System (DPS) that leverages advanced machine learning algorithms, incorporates a wide range of patient data, and provides timely and accurate predictions. The system should be capable of processing diverse datasets, including genetic information, clinical records, and lifestyle data, to enhance the accuracy of predictions and facilitate personalized healthcare.

The successful implementation of such a system would contribute significantly to early diabetes detection, enabling healthcare providers to initiate timely interventions, optimize treatment plans, and ultimately improve patient outcomes. The development of an intelligent DPS aligns with the broader goal of advancing personalized medicine and transforming healthcare delivery through data-driven decision-making.

3.2 Data Science Project Life Cycle

Data Science is a multidisciplinary field of study that combines programming skills, domain expertise and knowledge of statistics and mathematics to extract useful insights and knowledge from data.

In simple terms, a data science life cycle is nothing but a repetitive set of steps that you need to taketo complete and deliver a project/product to your client.

Although the data science projects and the teams involved in deploying and developing the model will be different, every data science life cycle will be slightly different in every other company. However, most of the data science projects happen to follow a somewhat similar process.

In order to start and complete a data science-based project, we need to understand the various roles and responsibilities of the people involved in building, developing the project.

3.2.1 Data Exploratory Analysis:

Exploratory Data Analysis refers to the critical process of performing initial investigations on data so as to discover patterns, to spot anomalies, to test hypothesis and to check assumptions with the help of summary statistics and graphical representations.

3.2.2 Data Pre-processing

Data pre-processing, a component of data preparation, describes any type of processing performed on raw data to prepare it for another data processing procedure. It has traditionally been an important preliminary step for the data mining process. More recently, data preprocessing techniques have been adapted for training machine learning models and AI models and for running inferences against them.
Data pre-processing transforms the data into a format that is more easily and effectively processed in data mining, machine learning and other data science tasks. The techniques are generally used at the earliest stages of the machine learning and AI development pipeline to ensure accurate results.

3.2.1.1 Check the Duplicate and low variation data

Types of Duplicate Features in Machine Learning:

Two things distinguish top data scientists from others in most cases: Feature Creation and Feature Selection. i.e., creating features that capture deeper/hidden insights about the business or customer and then making the right choices about which features to choose for your model.

1. Duplicate Values (Same value for each record)
2. Duplicate Index (value of two features are different but they occur at the same index) Keeping duplicate features in your dataset introduces the problem of multicollinearity. — In the case of linear models, weights distribution between the two features will be problematic.

— If you are using tree-based modes, it won't matter unless you are looking at feature importance.

— In the case of distance-based models, it will make that feature count more in the distance.

3.2.2.2 Identify and address the missing variables What are missing data?

Missing data are values that are not recorded in a dataset. They can be a single value missing in a single cell or missing of an entire observation (row). Missing data can occur both in a continuous variable (e.g., height of students) or a categorical variable (e.g., gender of a population).

Missing data are common in any field of natural or behavioral science, but it is particularly commonplace in social sciences research data.

In programming languages, missing values are represented as NA or Nan or simply an empty cell in an object.

• The origins of missing data

So where do the missing values come from, and why do they even exist?

Let's give an example. You are administering a questionnaire survey among a sample of respondents; and in the questionnaire, you are asking a question about household income. Now,what if a respondent refuses to answer that question? Would you make that up or rather leave the field empty? You'd probably leave that cell empty — creating an instance of missing value.

• Problems caused

Missing values are undesirable, but it is difficult to quantify the magnitude of effects in statistical and machine learning projects. If it's a large dataset and a very small percentage of data is missing the effect may not be detectable at all.

However, if the dataset is relatively small, every data point counts. In these situations, a missing data point means loss of valuable information.

In any case, generally missing data creates imbalanced observations, cause biased estimates, and in extreme cases, can even lead to invalid conclusion.

Case deletion: if the dataset is relatively large delete the complete record with a missing valueSubstitution: substitute missing cells with (a) column mean, (b) mean of nearest neighbors, (c) moving average, or (c) filling with the last observation.
Statistical imputation: a regression can be an effective way to determine the value of missing cell given other information in the dataset.
Sensitivity analysis: if the sample is small or missing values are relatively large then conduct a sensitivity analysis with multiple variations of outcomes.

3.2.2.2 Identify objects and convert into numerical values:

Defining data types when reading a CSV file
Creating a custom function to convert data type as type () vs. to numeric ()

When doing data analysis, it is important to ensure correct data types. Otherwise, you may get unexpected results or errors. In the case of Pandas, it will correctly infer data types in many cases, and you can move on with your analysis without any further thought on the topic.
Despite how well pandas works at some point in your data analysis process you will likely need to explicitly convert data from one type to another. This article will discuss how to change data to a numeric type. More specifically, you will learn how to use the Pandas built-in methods as type () and to numeric () to deal with the following common problems:

Converting string/int to int/float

Converting float to int

Converting a column of mixed data types

Handling missing values

Converting a money column to float

Converting Boolean to 0/1

Converting multiple data columns at once

3.2.2.3 Handling of Outliers

What is an outlier?

An outlier is an observation that lies an abnormal distance from other values in a random sample from a population.
There is, of course, a degree of ambiguity. Qualifying a data point as an anomaly leaves it up to the analyst or model to determine what is abnormal—and what to do with such data points.

There are also different degrees of outliers:

- Mild outliers lie beyond an “inner fence” on either side.
- Extreme outliers are beyond an “outer fence.”

Why do outliers occur? According to Tom Barenberg, chief economist and data consultant at Unity Marketing, “It can be the result of measurement or recording errors, or the unintended and truthful outcome resulting from the set’s definition.”
Outliers may contain valuable information. Or be meaningless aberrations caused by measurement and recording errors. In any case, they can cause problems with repeatable A/B test results, so it’s important to question and analyze outliers.

3.2.2.4 Categorical data and Encoding Techniques

What is Categorical Data?

Since we are going to be working on categorical variables in this article, here is a quick refresher onthe same with a couple of examples. Categorical variables are usually represented as ‘strings’ or ‘categories’ and are finite in number. Here are a few examples:

1. The city where a person lives: Delhi, Mumbai, Ahmedabad, Bangalore, etc.
2. The department a person works in: Finance, Human resources, IT, Production.
3. The highest degree a person has: High school, Diploma, Bachelors, Masters, PhD
4. The grades of a student: A+, A, B+, B, B- etc.

In the above examples, the variables only have definite possible values. Further, we can see thereare two kinds of categorical data-

- Ordinal Data: The categories have an inherent order
- Nominal Data: The categories do not have an inherent order.

Label Encoding:

- We use this categorical data encoding technique when the categorical feature is ordinal. In this case, retaining the order is important. Hence encoding should reflect the sequence.
- In Label encoding, each label is converted into an integer value. We will create a variable that contains the categories representing the education qualification of a person.

Binary Encoding:

- Binary encoding is a combination of Hash encoding and one-hot encoding. In this encoding scheme, the categorical feature is first converted into numerical using an ordinal encoder. Then the numbers are transformed in the binary number. After that binary value is split into different columns.

class.The also gives us p-value of the variables which tells us about significance of each independent variable.

3.2.4.2 Model 02(Decision Tree Classifier)

Decision Tree is a Supervised learning technique that can be used for both classification and Regression problems, but mostly it is preferred for solving Classification problems. It is a treestructured classifier, where internal nodes represent the features of a dataset, branches represent the decision rules, and each leaf node represents the outcome.

In a Decision tree, there are two nodes, which are the Decision Node and Leaf Node. Decision nodes are used to make any decision and have multiple branches, whereas Leaf nodes are the output of those decisions and do not contain any further branches.

The decisions or the test are performed based on features of the given dataset.
It is a graphical representation for getting all the possible solutions to a problem/decision based on given conditions.

It is called a decision tree because, like a tree, it starts with the root node, which expands on further branches and constructs a tree-like structure.
Decision Trees (DTs) are a non-parametric supervised learning method used for classification and regression. The goal is to create a model that predicts the value of a target variable by learning simple decision rules inferred from the data features. A tree can be seen as a piece wise constant approximation.

3.2.4.3 Model 03(Random Forest Classifier)

Random forest is an algorithm that consists of many decision trees. It was first developed by Leo Bierman and Adele Cutler. The idea behind it is to build several trees, to have the instance classified by each tree, and to give a "vote" at each class. The model uses a "bagging" approach and the random selection of features to build a collection of decision trees with controlled

variance. The instance's class is to the class with the highest number of votes, the class that occurs the most within the leaf in which the instance is placed.
The error of the forest depends on:

• Trees correlation: the higher the correlation, the higher the forest error rate.

• The strength of each tree in the forest. A strong tree is a tree with low error. By using trees that classify the instances with low error the error rate of the forest decreases.

3.2.4.4 Model 04(Extra Trees Classifier)

This class implements a meta estimator that fits several randomized decision trees (a.k.a. extra- trees) on various sub-samples of the dataset and uses averaging to improve the predictive accuracy and control over-fitting.

3.2.4.5 Model 05(KNN Classifier)

K-Nearest Neighbor is one of the simplest Machine Learning algorithms based on Supervised Learning technique K-NN algorithm assumes the similarity between the new case/data and available cases and put the new case into the category that is most like the available categories'-NN algorithm stores all the available data and classifies a new data point based on the similarity. This means when new data appears then it can be easily classified into a well suite category by using KNN algorithm.
K-NN algorithm can be used for Regression as well as for Classification but mostly it is used for the Classification problems K-NN is a non-parametric algorithm, which means it does not make any assumption on underlying data. It is also called a lazy learner algorithm because it does not learn from the training set immediately instead it stores the dataset and at the time of classification, it performs an action on the dataset KNN-algorithm at the training phase just stores the dataset and when it gets new data, then it classifies that data into a category that is much like the new data.

3.2.4.6 Model 06(Naïve Bayes)

Naïve Bayes is a probabilistic machine learning algorithm used for many classification functions and is based on the Bayes theorem. Gaussian Naïve Bayes is the extension of naïve Bayes. While other functions are used to estimate data distribution, Gaussian or normal distribution is the simplest to implement as you will need to calculate the mean and standard deviation for the training data.

What is the Naive Bayes Algorithm?

Naive Bayes is a probabilistic machine learning algorithm that can be used in several classification tasks. Typical applications of Naive Bayes are classification of documents, filtering spam, prediction and so on. This algorithm is based on the discoveries of Thomas Bayes and hence its name.

The name "Naïve" is used because the algorithm incorporates features in its model that are independent of each other. Any modifications in the value of one feature do not directly impact the value of any other feature of the algorithm. The main advantage of the Naïve Bayes algorithm is that it is a simple yet powerful algorithm.

It is based on the probabilistic model where the algorithm can be coded easily, and predictions did quickly in real-time. Hence this algorithm is the typical choice to solve realworld problems as it can be tuned to respond to user requests instantly. But before we dive deep into Naïve Bayes and Gaussian Naïve Bayes, we must know what is meant by conditional probability.

3.2.4.7 Model 07(XG Boost)

filtering spam, prediction and so on. This algorithm is based on the discoveries of Thomas Bayes and hence its name.

The name "Naïve" is used because the algorithm incorporates features in its model that are independent of each other. Any modifications in the value of one feature do not directly impact the value of any other feature of the algorithm. The main advantage of the Naïve Bayes algorithm is that it is a simple yet powerful algorithm.

It is based on the probabilistic model where the algorithm can be coded easily, and

predictions did quickly in real-time. Hence this algorithm is the typical choice to solve realworld problems as it can be tuned to respond to user requests instantly. But before we dive deep into Naïve Bayes and Gaussian Naïve Bayes, we must know what is meant by conditional probability.

3.2.4.8 Model 08(Light GBM)

Light GBM is a gradient boosting framework based on decision trees to increases the efficiency of the model and reduces memory usage.
It uses two novel techniques: Gradient-based One Side Sampling and Exclusive Feature Bundling (EFB) which fulfills the limitations of histogram-based algorithm that is primarily used in all GBDT (Gradient Boosting Decision Tree) frameworks. The two techniques of GOSS and EFB described below form the characteristics of Light GBM Algorithm. They comprise together to make the model work efficiently and provide it a cutting edge over

other GBDT frameworks.
Gradient-based One Side Sampling Technique for Light GBM:
Different data instances have varied roles in the computation of information gain. The instances with larger gradients (i.e., under-trained instances) will contribute more to the information gain. GOSS keeps those instances with large gradients (e.g., larger than a predefined threshold, or among the top percentiles), and only randomly drop those instances with small gradients to retain the accuracy of information gain estimation. This treatment can lead to a more accurate gain estimation than uniformly random sampling, with the same target sampling rate, especially when the value of information gain has a large range.

3.2.4.9 Model 09 (SVC)

Support Vector Machine or SVM is one of the most popular Supervised Learning algorithms,which is used for Classification as well as Regression problems. However, primarily, it is used for
Classification problems in Machine Learning.
The goal of the SVM algorithm is to create the best line or decision boundary that can segregate ndimensional space into classes so that we can easily put the new data point in the correct category in the future. This best decision boundary is called a hyperplane.

3.3 AI / ML Model Analysis and Final Results

We used our train dataset to build the above models and used our test data to check the accuracy and performance of our models.
We used confusion matrix to check accuracy, Precision, Recall and F1 score of our models and compare and select the best model for given auto dataset of size ~ 272252 policies.

3.3.1 Different Model codes

This section in which we used different types of model code as follows :
This code is a multiclassified code consists of

- LogisticRegression o DecisionTreeClassifier o RandomForestClassifier o ExtraTreesClassifier o KN neighborsClassifier
- SVM
- GaussianNB.

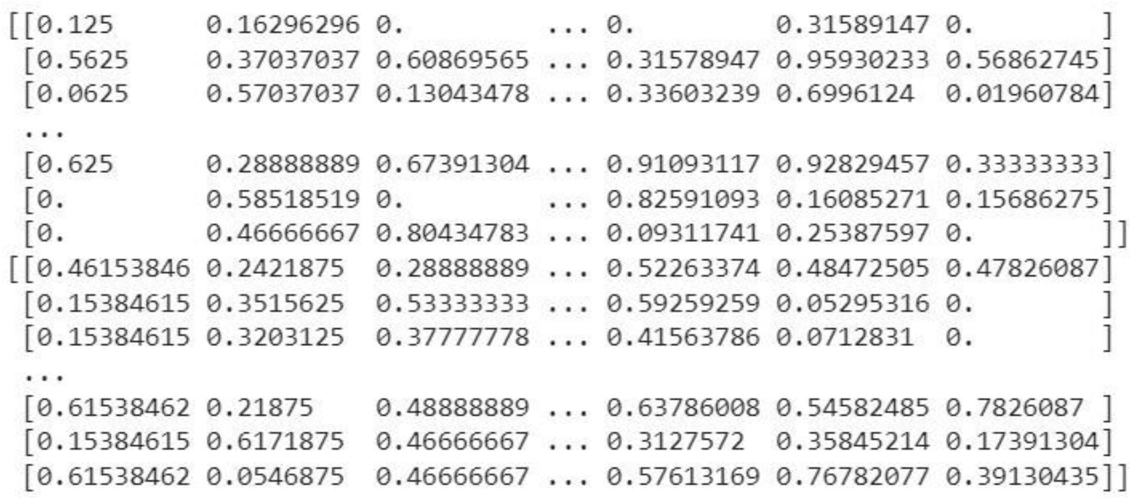
```
X=dataset.iloc[:,:-1]
y=dataset.iloc[:,-1]
```

Splitting the dataset into train and test set

```
from sklearn.model_selection import train_test_split as tts
x_train,x_test,y_train,y_test=tts(X,y,test_size=0.2,random_state=42)

x_train.shape,x_test.shape,y_train.shape,y_test.shape
```

Scaling the features using minmax scaler

```
from sklearn.preprocessing import MinMaxScaler
sc=MinMaxScaler(feature_range=(0,1))

x_train=sc.fit_transform(x_train)

x_test=sc.fit_transform(x_test)

print(x_train)
print(x_test)
```



```
[[0.125      0.16296296 0.         ... 0.         0.31589147 0.        ]
 [0.5625     0.37037037 0.60869565 ... 0.31578947 0.95930233 0.56862745]
 [0.0625     0.57037037 0.13043478 ... 0.33603239 0.6996124  0.01960784]
 ...
 [0.625      0.28888889 0.67391304 ... 0.91093117 0.92829457 0.33333333]
 [0.         0.58518519 0.         ... 0.82591093 0.16085271 0.15686275]
 [0.         0.46666667 0.80434783 ... 0.09311741 0.25387597 0.        ]]
[[0.46153846 0.2421875  0.28888889 ... 0.52263374 0.48472505 0.47826087]
 [0.15384615 0.3515625  0.53333333 ... 0.59259259 0.05295316 0.        ]
 [0.15384615 0.3203125  0.37777778 ... 0.41563786 0.0712831  0.        ]
 ...
 [0.61538462 0.21875    0.48888889 ... 0.63786008 0.54582485 0.7826087 ]
 [0.15384615 0.6171875  0.46666667 ... 0.3127572  0.35845214 0.17391304]
 [0.61538462 0.0546875  0.46666667 ... 0.57613169 0.76782077 0.39130435]]
```

Building and evaluating the models

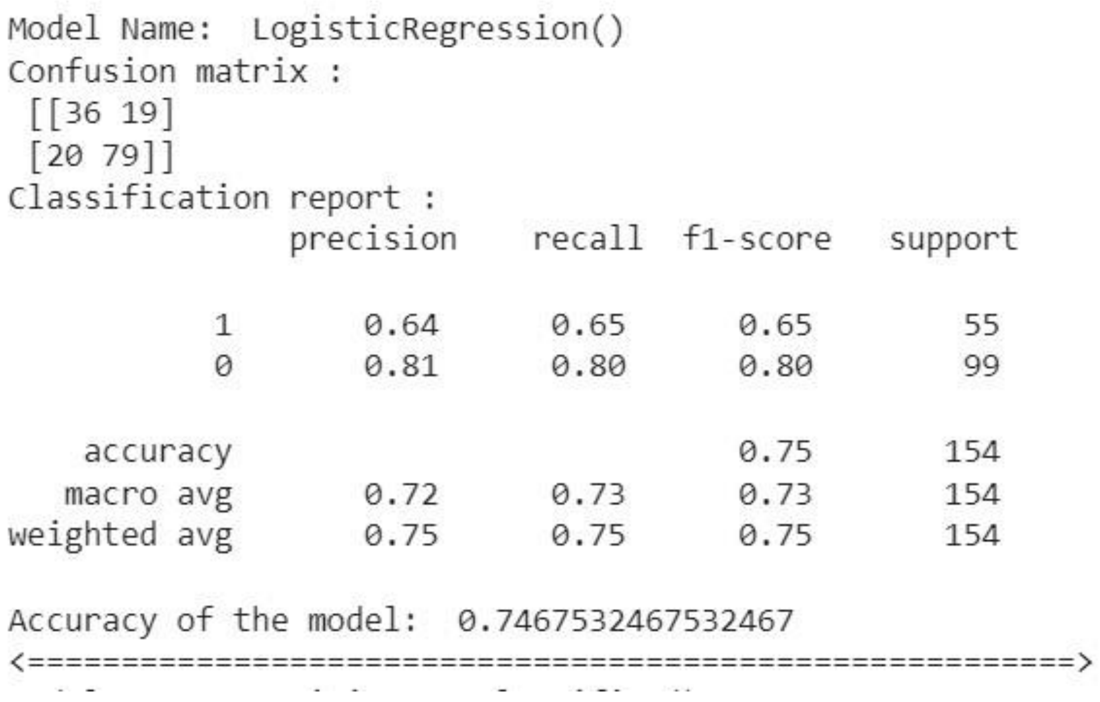
```
from sklearn.linear_model import LogisticRegression
from sklearn.tree import DecisionTreeClassifier
from sklearn.neighbors import KNeighborsClassifier
from sklearn.naive_bayes import GaussianNB

# Create objects of classification algorithms with default hyper-parameters

ModelLR = LogisticRegression()
ModelDC = DecisionTreeClassifier()
ModelKNN = KNeighborsClassifier(n_neighbors=14)
```

```
ModelGNB = GaussianNB()

MM = [ModelLR, ModelDC, ModelKNN, ModelGNB]

for models in MM:

    # Train the model training dataset

    models.fit(x_train, y_train)

    # Prediction the model with test dataset

    y_pred = models.predict(x_test)

    # Print the model name

    print('Model Name: ', models)

    # confusion matrix in sklearn

    from sklearn.metrics import confusion_matrix
from sklearn.metrics import classification_report
from sklearn.metrics import accuracy_score

    # actual values

    actual = y_test

    # predicted values

    predicted = y_pred

    # confusion matrix

    matrix = confusion_matrix(actual,predicted, labels=[1,0],sample_weight=None, normalize=None)
print('Confusion matrix : \n', matrix)


    # classification report for precision, recall f1-score and accuracy

    C_Report = classification_report(actual,predicted,labels=[1,0])

    print('Classification report : \n', C_Report)

    #accuracy score
    ac_score=accuracy_score(actual,predicted)
    print("Accuracy of the model: ",ac_score)
    print("<==========================================================>")
```

4.0 Conclusions and Future work

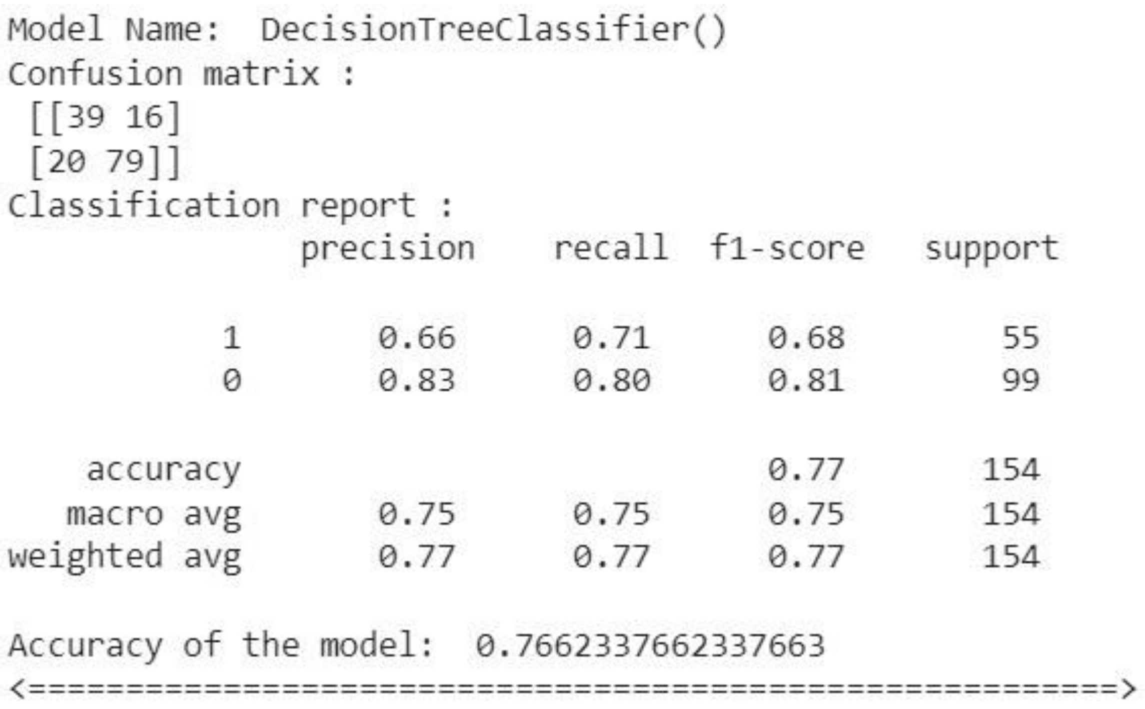
- Our problem is classification problem.
- There are total four algorithms in classification problems
  1. Logistic Regression.
  2. DecisionTreeClassifier 3. KNeighbours Classifier.
  4. GaussianNB.

Logistic Regression Output:



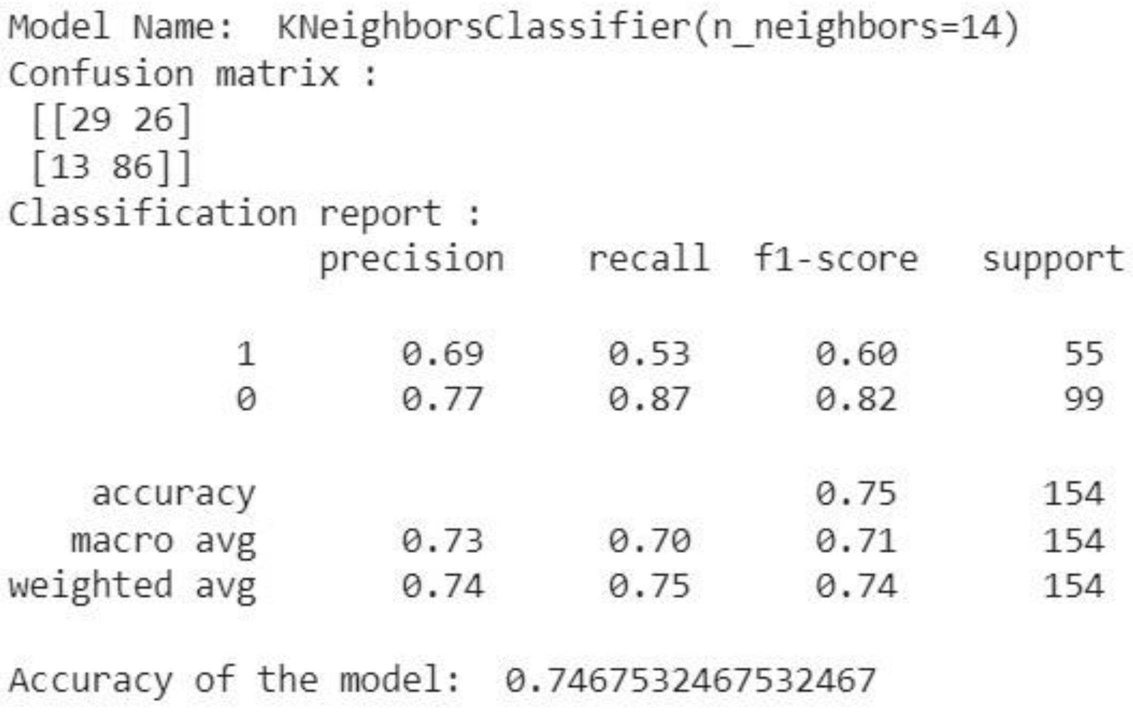
```
Model Name:  LogisticRegression()
Confusion matrix :
 [[36 19]
 [20 79]]
Classification report :
              precision    recall  f1-score   support

           1       0.64      0.65      0.65        55
           0       0.81      0.80      0.80        99

    accuracy                           0.75       154
   macro avg       0.72      0.73      0.73       154
weighted avg       0.75      0.75      0.75       154

Accuracy of the model:  0.7467532467532467
<=========================================================>
```

DecisionTreeClassifier Output:



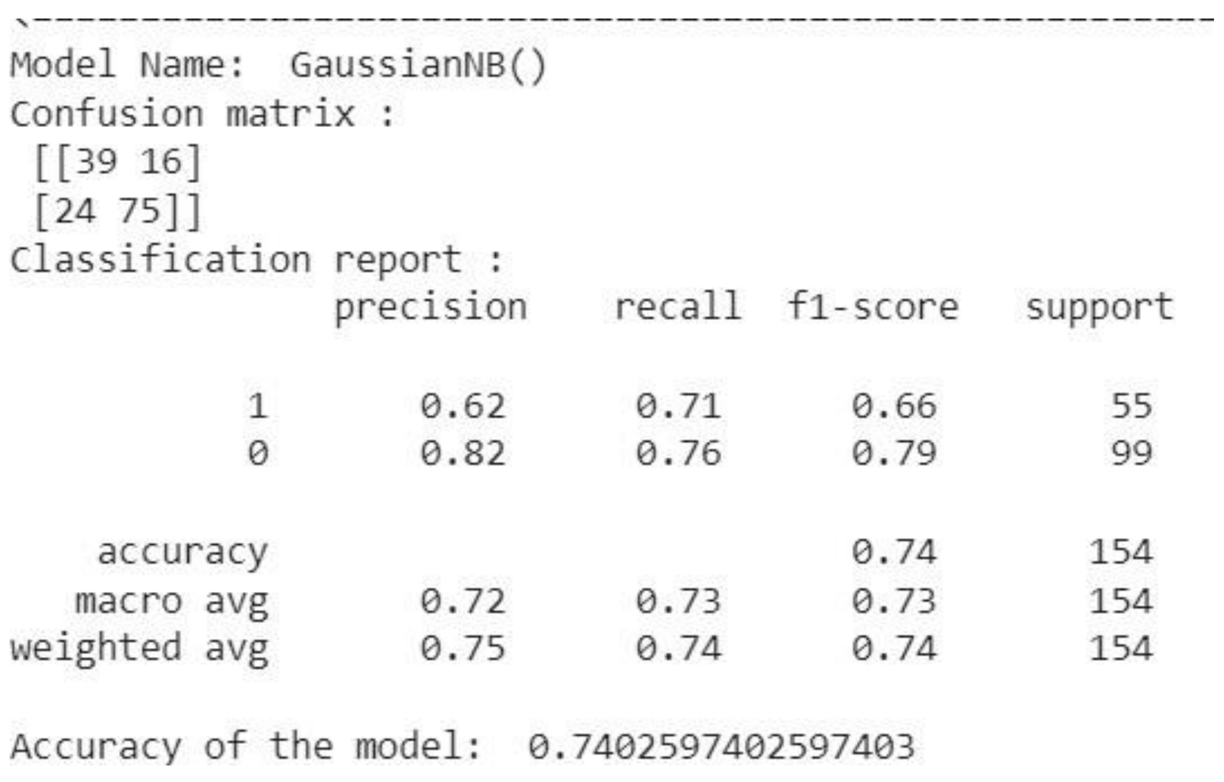
```
Model Name:  DecisionTreeClassifier()
Confusion matrix :
 [[39 16]
 [20 79]]
Classification report :
               precision    recall  f1-score   support

           1       0.66      0.71      0.68        55
           0       0.83      0.80      0.81        99

    accuracy                           0.77       154
   macro avg       0.75      0.75      0.75       154
weighted avg       0.77      0.77      0.77       154

Accuracy of the model:  0.7662337662337663
<==========================================================>
```

KNeighbours Classifier Output:



```
Model Name:  KNeighborsClassifier(n_neighbors=14)
Confusion matrix :
 [[29 26]
 [13 86]]
Classification report :
               precision    recall  f1-score   support

           1       0.69      0.53      0.60        55
           0       0.77      0.87      0.82        99

    accuracy                           0.75       154
   macro avg       0.73      0.70      0.71       154
weighted avg       0.74      0.75      0.74       154

Accuracy of the model:  0.7467532467532467
```

GaussianNB Output:



```
`-----------------------------------------------------------------
Model Name:  GaussianNB()
Confusion matrix :
 [[39 16]
 [24 75]]
Classification report :
              precision    recall  f1-score   support

           1       0.62      0.71      0.66        55
           0       0.82      0.76      0.79        99

    accuracy                           0.74       154
   macro avg       0.72      0.73      0.73       154
weighted avg       0.75      0.74      0.74       154

Accuracy of the model:  0.7402597402597403
```

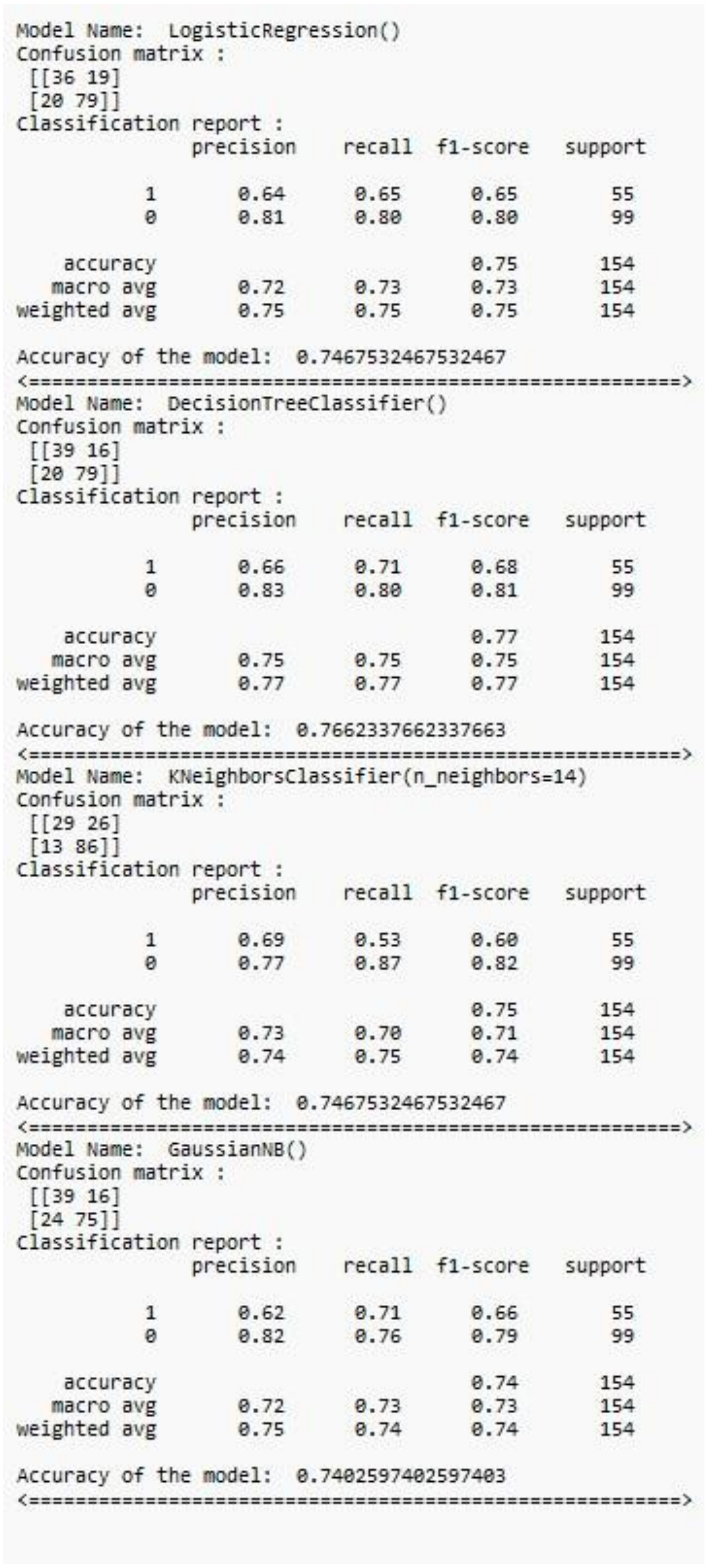
5.0 References https://www.javatpoint.com/types-of-machinelearning
https://www.kaggle.com

6.0 Appendices

- Kernel methods
- Graphical models
- Reinforcement learning
- Convex analysis
- Optimization
- Bio informatics
- Minimal description length principle
- Topics in information theory
- Decision theory
- Network algorithms
- Computational Social Science
- Natural Language Processing Recent trends in deep learning and representation learning

- Facial recognition
- Voice Recognition
- Virtual Personal Assistants

6.1 Python code Results:



```
Model Name:  LogisticRegression()
Confusion matrix :
 [[36 19]
 [20 79]]
Classification report :
               precision    recall  f1-score   support

           1       0.64      0.65      0.65        55
           0       0.81      0.80      0.80        99

    accuracy                           0.75       154
   macro avg       0.72      0.73      0.73       154
weighted avg       0.75      0.75      0.75       154

Accuracy of the model:  0.7467532467532467
<=========================================================>
Model Name:  DecisionTreeClassifier()
Confusion matrix :
 [[39 16]
 [20 79]]
Classification report :
               precision    recall  f1-score   support

           1       0.66      0.71      0.68        55
           0       0.83      0.80      0.81        99

    accuracy                           0.77       154
   macro avg       0.75      0.75      0.75       154
weighted avg       0.77      0.77      0.77       154

Accuracy of the model:  0.7662337662337663
<=========================================================>
Model Name:  KNeighborsClassifier(n_neighbors=14)
Confusion matrix :
 [[29 26]
 [13 86]]
Classification report :
               precision    recall  f1-score   support

           1       0.69      0.53      0.60        55
           0       0.77      0.87      0.82        99

    accuracy                           0.75       154
   macro avg       0.73      0.70      0.71       154
weighted avg       0.74      0.75      0.74       154

Accuracy of the model:  0.7467532467532467
<=========================================================>
Model Name:  GaussianNB()
Confusion matrix :
 [[39 16]
 [24 75]]
Classification report :
               precision    recall  f1-score   support

           1       0.62      0.71      0.66        55
           0       0.82      0.76      0.79        99

    accuracy                           0.74       154
   macro avg       0.72      0.73      0.73       154
weighted avg       0.75      0.74      0.74       154

Accuracy of the model:  0.7402597402597403
<=========================================================>
```