DIABETES PREDICTION WITH GENETIC OPTIMIZATION

Ms. GAYATHRI DEVI[1], Mr. P.DHARMA TEJA[2], Mr. M. VAMSHI[3], Mrs.L.T.PRIYANKA[4]

1. BTECH, NADIMPALLI SATYANARAYANA RAJU INSTITUTE OF TECHNOLOGY, SONTYAM, VISAKHAPATNAM, ANDHRA PRADESH, INDIA - 531173

2. BTECH, NADIMPALLI SATYANARAYANA RAJU INSTITUTE OF TECHNOLOGY , SONTYAM, VISAKHAPATNAM, ANDHRA PRADESH, INDIA - 531173

BTECH, NADIMPALLI SATYANARAYANA RAJU INSTITUTE OF TECHNOLOGY , SONTYAM, VISAKHAPATNAM, ANDHRA PRADESH, INDIA - 531173

3.BTECH, NADIMPALLI SATYANARAYANA RAJU INSTITUTE OF TECHNOLOGY , SONTYAM, VISAKHAPATNAM, ANDHRA PRADESH, INDIA - 531173

4. Professor COMPUTER SCIENCE AND ENGINEERING, NADIMPALLI SATYANARAYANA RAJU INSTITUTE OF TECHNOLOGY, SONTYAM, VISAKHAPATNAM, ANDHRA PRADESH, INDIA-531173

ABSTRACT

All of the cells and organs in our bodies depend on glucose, sometimes known as sugar, as a major energy source. Diabetes is caused by a continuous, excessive rise in blood sugar or glucose levels over the desirable range. Diabetes is identified through blood sugar tests on individuals. By eliminating human judgement and producing precise findings, computer vision plays a significant role in the field of human health. The improvement of diabetes classification is the main goal of this study. In this article, we concentrate on comparing algorithms to improve the predictive model's performance by utilising data mining and machine learning techniques. The UCI machine supported the "Pima Indians Diabetes Dataset" Standard, which we used.

learning archive**.** To improve the dataset's potential, feature selection is done. To assess the effectiveness of the model, a variety of algorithms are performed to the dataset, including Support Vector Machine (SVM), Naive Bayes, Decision Trees, K-Nearest Neighbors (KNN), Logistic Regression, and voting classification. To recommend the best classifier for the sample dataset, evaluation measures including Precision, Recall, Specificity, and mean absolute error are generated for each model. To compare the model's accuracy, Waikato Environment for Knowledge Analysis Toolkit was utilised. To improve generalisation of accuracy, cross-validation is used. The conclusion shows that the SVM algorithm has a 78% accuracy rate. So, it appears that the research will help in type 2 diabetes prediction (T2D).

KEY WORDS: SVM,KNN,UCI

1 INTRODUCTION

Diabetes is a condition that reduces the ability of the body to release insulin. Glucose is produced in the blood from the food we eat. By delivering glucose to the cells, insulin controls blood sugar levels. These cells carry out their specific role by converting glucose into energy. Blood glucose levels rise as a result of inadequate insulin secretion.

Type 1, Type 2, and gestational diabetes are the three main subtypes [1].

Insufficient insulin production is the primary cause of type 1 diabetes. The body's beta cells that make insulin are destroyed by the immune system. As the body produces relatively little insulin, insulin must be administered intravenously to the body in order to keep blood glucose levels stable.

Despite being present in adults, it is most common in children. Type 1 diabetes is an unclear specific cause, however there are some things that may indicate a higher risk, such as family history, environmental variables, and the presence of immune system cells that can cause damage.

Because type 2 diabetes does not use the glucose produced by the body, it is also known as insulin resistance. Diabetes is brought on by a massive buildup of glucose in the blood. It affects adults the most frequently. Researchers are still trying to figure out why some people get type 2 diabetes and others don't. Yet, it is obvious that some factors—including weight, race, family history, age, high blood pressure, abnormal cholesterol, and family history—increase the risk.

Pregnancy is the primary cause of gestational diabetes. shift in

This situation is caused by the hormones generated. Only during pregnancy does it occur. The unborn child will probably develop type 2 diabetes in the future. Excessive growth, low blood sugar, type 2 diabetes later in life, and even death are complications that might affect your unborn child. Age, a family history of the condition, weight, and race are all risk factors for gestational diabetes.

A chronic condition is a sickness or illness that persists over time or has long-term repercussions. These disorders had a significant negative impact on quality of life. One of the most severe illnesses, diabetes is widespread. A leading explanation of mortality in adults across the globe is this chronic ailment.

The cost of chronic illnesses is also a factor. the majority of Governments and individuals spend a considerable amount of money on chronic illnesses. According to global diabetes data from 2013, there were 382 million people living with the disease worldwide. In 2012, it was the eighth major cause of death for both sexes and the fifth greatest cause of death in women. Diabetes is more likely to be present in nations with higher incomes. Worldwide, 451 million adults received diabetes treatment in 2017. The number of people with diabetes is expected to reach about 693 million by 2045, with half of them going untreated. Moreover, 850 million USD were spent in 2017 on diabetic patients.

Diagnosis levels of diabetes:

A1C tests: These show a person's blood history for the most recent months. The table lists the range of the various classes. Diabetes and prediabetes are treated with it.Tests for FPG Fasting plasma glucose levels are used to distinguish between diabetes and prediabetes.

OGT: Oral glucose testing is the blood test used to diagnose diabetes, gestational diabetes, and prediabetes.

Effects of diabetes

Diabetes affects several different bodily organs, including the pancreas, heart disease risk, hypertension, kidney problems, and pancreatic complications.nerve damage, foot problems, ketoacidosis, disturbing visual effects, various eye problems, waterfalls, and glaucoma. Cardiovascular illness is one of the many body parts that might affect diabetes. Diabetes significantly raises the risk of a number of cardiovascular issues, such as coronary artery disease with chest discomfort (angina), heart attacks, strokes, and arterial narrowing (atherosclerosis). Diabetes increases your risk of developing heart disease or stroke.Excess sugar can harm the walls of the tiny blood arteries (capillaries) that nourish your nerves, especially in your legs, causing nerve damage (neuropathy). The tingling, numbness, burning, or pain that may result from this typically starts at the tips of the toes or fingers and progressively moves higher. The damaged limbs may become completely devoid of feeling. Problems with nausea, vomiting, diarrhoea, or constipation can result from damage to the nerves that control digestion. It might cause erectile dysfunction in men.The kidneys' glomeruli, which are millions of microscopic blood vessel clusters, filter waste from your blood to cause kidney damage (nephropathy). This delicate filtering system can be harmed by diabetes. Kidney failure or irreversible end-stage renal disease, which may require dialysis, can result from severe damage. renal transplantation.Retinopathy, or eye damage, is a condition where diabetes affects the retina's blood vessels and can result in blindness. Diabetes also raises the risk of glaucoma and cataracts, two devastating eye diseases.Foot injury: A number of foot issues are made more likely by nerve damage in the feet or insufficient blood supply to the feet. Blisters and injuries that go untreated can get seriously infected and heal badly. An eventual toe, foot, or limb amputation may be necessary due to these illnesses.

Skin conditions:

Diabetes may make you more prone to bacterial and fungal infections, among other skin issues.Type 2 diabetes, often known as diabetes mellitus, is a metabolic condition that elevates blood sugar levels. Diabetes Type 2 can be reversed, unlike Type 1 diabetes. Depending on the patient, the course of treatment may vary. Some people don't even need to take insulin; they just need to make lifestyle adjustments like losing weight, having a healthy lifestyle, etc. Yet others require medical care that includes medication, insulin injections, and a healthy lifestyle to regulate blood sugar levels. Food consumed during metabolic activity is transformed to energy. This process needs the insulin hormone, which aids in turning sugars into energy. Type 2 diabetes develops as the body gradually loses ability to take up insulin, which controls the sugar level, the body's sugar levels won't be kept under control, resulting in increased sugar readings in Type 2 diabetes sufferers. Diabetes that develops later in life is often known as "adult onset" diabetes. As the body is displaying resistance to absorbing insulin, this is also known as "insulin resistance." When compared to other types of diabetes, this one is more prevalent. According to statistics, 90 out of every 100 people with diabetes have Type 2 diabetes.

2. LITERATURE SURVEY AND RELATED WORK

People all across the world are very aware of the rise in diabetes in recent years. Modern technological advancements have encouraged medical experts to use cutting-edge methods to anticipate diseases. For the diabetes categorization dataset, a number of already-existing models were used.

2.1 Type 2 diabetes mellitus prediction model based on data mining [6]

An innovative model for predicting type 2 diabetes mellitus based on data mining techniques (T2DM). This research focuses mostly on increasing the prediction model's accuracy and making it adaptable to multiple datasets. The model consists of two components: the modified K-means algorithm and the logistic regression algorithm, which are based on a series of preprocessing steps. To compare our findings with those of other researchers, the Waikato Environment for Knowledge Analysis toolbox and the Pima Indians Diabetes Dataset were both used. We used it to analyse two more diabetes datasets in order to assess the effectiveness of our model in more detail. The concept is thus demonstrated to be helpful for the practical management of diabetes health.

2.2 Analysis of diabetes mellitus for early prediction using early optimal features selection [7]

The goal of this research is to leverage significant information, create a prediction algorithm using data mining, and choose the best classifier to produce results that are as close to clinical outcomes as possible. The suggested approaches concentrate on identifying the characteristics that, when used in predictive analysis, fail in the early diagnosis of diabetic millets. The analysis of diabetic data demonstrates that the decision tree algorithm and random forest have the maximum specificity of 98% and 95%, respectively. The best accuracy, according to naive Bayesian results, is 82%. In order to increase classification accuracy, the research additionally generalises the selection of the best features from the database.

Analyzing Feature Importance for Diabetes Prediction using Machine Learning

Obesity, elevated blood sugar, and other factors are the primary causes of diabetes. Thus, we will learn what the essential components of diabetes' cause are in this essay. In areas of application where datasets containing tens or thousands of elements are available, variable and feature choice has become the focus of significant research. Similarly, in order to determine whether a person has a possibility of developing diabetes in the future, we will focus on the most important characteristics. [8]

2.3 Comparison of Data Mining Algorithms In The Diagnosis Of Type2 Diabetes

Medical data mining has produced significant, worthwhile, and efficient accomplishments that can improve medical knowledge and help clinicians make critical judgements. In this study, data mining algorithms are used to diagnose type -2 diabetes. 768 diabetic patient samples from the Pima Indians Dataset are included in the dataset used for the diagnosis of type 2 diabetes. In this study, the data mining techniques utilised to identify type II diabetes include Naive Bayes, RBF Network, and J48. Weka is used by the so-called algorithms to carry out diagnosis. In order to identify whether algorithm was more accurate at diagnosing type II diabetes, they were compared at the end [16].

2.4 Analyzing Diabetes Datasets using Data Mining [15]

The writers of this research concentrated on patient data with diabetes. The inability of the body of a diabetic patient to control blood glucose levels can have an impact on other physiological mechanisms. Other physiological and psychological characteristics may also begin to malfunction as a result, including skin folding and weight loss. These parameters could be a useful source of information for the study. Data mining can help reduce the chance of contracting several common diseases like diabetes, heart disease, and cancer in the medical field. When it comes to automated content analysis and using some machine learning techniques to help humanity predict non-communicable diseases like diabetes, the explosion in digital information has created a number of obstacles. In this study, many classification methods, including as Random Forest, Naive Bayes, MLP, J.48, ZeroR, and Regression were used

to illustrate the outcome. The purpose of the research being done is to glean knowledge from the provided data set and produce thorough and insightful findings.

2.5 Short Survey on Naive Bayes Algorithm

A classification technique called Naive Bayes is based on the Bayes theorem and makes strong and naive assumptions about independence. By presuming that features are independent of a specific class, it simplifies learning. The hidden naive Bayes algorithm, text categorization, standard naive Bayes, and machine learning are discussed in this paper's examination of the naive Bayes algorithm. Also illustrates augmented naive Bayes through examples. In order to better comprehend the algorithm, some naive Bayes applications and their benefits and drawbacks have been discussed..

2.6 Machine Learning Group at the University of Waikato.

Automatically removing useful information from data is a task that machine learning is concerned with. Finding patterns that help us comprehend and forecast the domain from which the data was obtained is the goal. By enabling software to learn from observations, machine learning plays a crucial role in artificial intelligence by allowing it to adjust its behaviour, make predictions, and provide users with insight into observed data. These patterns can be automatically extracted using effective algorithms in the form of simple. All machine learning algorithms are included in WEKA, which is widely used for research, instruction, and commercial applications.

Implementation Study

PYTHON LIBRARIES:

- Numpy
- Matplotlib
- pandas
- sk-learn
- seaborn

Numpy:
Numerical Python is referred to as NumPy. The Python package NumPy is used to manipulate arrays. Moreover, it has matrices, fourier transform, and functions for working in the area of linear algebra. A general-purpose array processing package is called Numpy. It offers a multidimensional array object with outstanding speed as well as capabilities for interacting with these arrays. It is Python's foundational scientific computing package. In addition to its apparent scientific applications, Numpy is a powerful multi-dimensional data container. The equivalent of arrays in Python are lists, although they take a long time to execute. The goal of NumPy is to offer array objects that are up to 50 times faster than conventional Python lists. A practical and effective method for handling the enormous amount of data. Moreover, NumPy makes matrix multiplication and data manipulation incredibly simple. It is reasonable to work with a huge number of data because NumPy is quick.

Matplotlib:
A tool for visualising data, Matplotlib is a low-level graph charting framework written in Python. For platform compatibility, Matplotlib is primarily written in Python, with a small amount of code written in C, Objective-C, and JavaScript. For Python and its numerical extension NumPy, Matplotlib is a cross-platform data visualisation and graphical charting package. As a result, it presents a strong open-source substitute for MATLAB. The APIs (Application Programming Interfaces) for matplotlib allow programmers to incorporate graphs into GUI applications. The way a Python matplotlib script is written makes it possible to create a visual data plot in the majority of cases with just a few lines of code.

Pandas
For the purpose of manipulating and analysing data, the Python programming language has a software package called pandas. It includes specific data structures and procedures for working with time series and mathematical tables. Pandas is primarily used for tabular data analysis and related manipulation in Data Frames. Data can be imported into Pandas from a variety of file types, including Microsoft Excel, JSON, Parquet, SQL database tables, and comma-separated values. Data cleaning and wrangling functions, as well as operations like merging, restructuring, and choosing, are all supported by Pandas. Several of the R programming language's proven functionality for working with Data Frames were brought into Python with the introduction of pandas. The NumPy library, which is focused on effectively working with arrays rather than the characteristics of working with Data Frames, is the foundation upon which the Panda library is based. Pandas is a term used to describe an open-source Python library that offers high-performance data manipulation. Pandas, which means an Econometrics from Multidimensional Data, gets its name from the phrase panel data. Wes McKinney created it in 2008 and uses Python to analyse data.

Scikit-learn

Sklearn, formerly known as scikits.learn, is a free machine learning package for the Python computer language. Support-vector machines, random forests, gradient boosting, k-means, and DBSCAN are just a few of the classification regression and clustering algorithms it offers. It is also built to work with the Python scientific and numerical libraries Numpy and Scipy. The most effective and reliable Python machine learning library is called Skearn (Skit-Learn). Via a Python consistency interface, it offers a variety of effective tools for statistical modelling and machine learning, including classification, regression, clustering, and dimensionality reduction. This library is based on NumPy, SciPy, and Matplotlib and was written primarily in Python.

Seaborn

A matplotlib-based Python data visualisation library is called Seaborn. It offers a high-level drawing tool for creating visually appealing and educational statistical visuals. Python's Seaborn visualisation module is fantastic for plotting statistical visualisations. It offers lovely default styles and colour schemes to enhance the appeal of statistics charts. It is constructed on top of the Matplotlib toolkit and is tightly integrated with the Pandas data structures. With Seaborn, visualisation will be at the heart of data exploration and comprehension. For a better comprehension of the dataset, it offers dataset-oriented APIs that allow us to switch between various visual representations for the same variables. An open source, BSD-licensed Python package called Seaborn offers a high level API for data visualisation using the Python programming language. A collaboration between business and academia called the International Skin Imaging Collaboration (ISIC): Melanoma Project aims to make it easier to use digital skin imaging to reduce the mortality rate from skin cancer. ISIC began organising international competitions for the examination of skin lesions for the diagnosis and detection of melanoma in 2015.

The foundation of MobileNets is the idea of streamlined architectures, which employ depth-wise separable convolutions followed by point-wise convolutions to significantly reduce the amount of learnable parameters and support the construction of lightweight deep neural networks. Essentially, it reduces the overall amount of necessary floating point calculations, which is helpful for embedded and mobile computer vision applications when there is a lack of processing capacity.

3 PROPOSED WORK AND ALGORITHM

In this study, we provide a three-level architecture model for type 2 diabetes prediction. Its three main approaches are data pre-processing, data modelling, and evaluation metrics. The architecture of the entire system is depicted in Fig. 1.

The Diabetes dataset is used as the initial input for building the models. The dataset is initially made up of unnecessary and duplicated information. In order to improve the model's accuracy, pre-processing procedures like data cleaning and feature selection are used to the dataset. It is carried out to solve the overfitting issue. Following that, the output data is divided into two groups: training data and testing data. 25% of our data were used for testing, while the remaining 75% were for training.

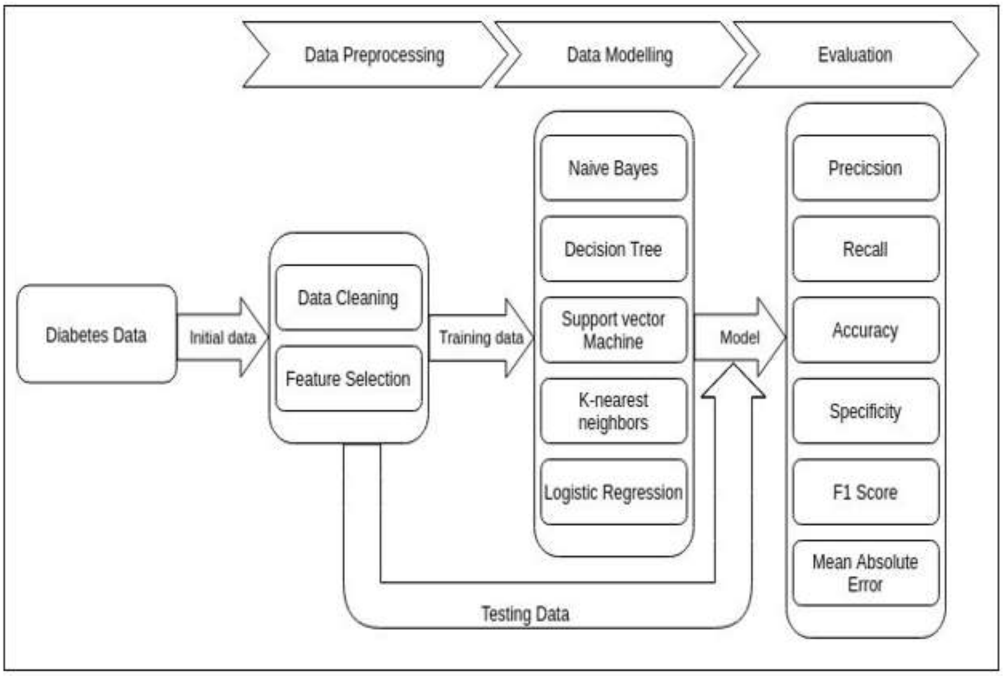


Fig: SYSTEM ARCHITECTURE

Several methods, including Naive Bayes, Decision Tree, Support Vector Machine, K-nearest Neighbors, and Logistic

Regression, are used in the second phase.

in excess of the training dataset. Models are built using a variety of algorithms. The created model is then put to the test using the test dataset.

The accuracy of the model is predicted in the third phase using evaluation metrics including Precision, Recall, Specificity, Accuracy, and Mean Absolute Error. All of the models' evaluation indicators are compared. Cross Validation is used in the following stage to increase the model's correctness.

Algorithm:-

The majority of evolutionary algorithms are adaptive heuristic search algorithms known as genetic algorithms (GAs). Natural selection and genetics are the foundations of genetic algorithms. These are creative uses of provided random search the search into the area of superior performance in the solution space using historical data. They are frequently employed to produce excellent answers to optimisation and search-related issues.

Natural selection is simulated by genetic algorithms, which means that only those species that can adapt to changes in their environment will be able to survive, procreate, and pass on to the next generation. In order to solve an issue, they essentially replicate "survival of the fittest" among individuals of successive generations. Each generation consists of a population of people, and each person represents a potential solution or a point in the search space. Every person is represented by a string of characters, integers, floats, and bits. This string resembles a chromosome.

Foundation of Genetic Algorithms

Genetic algorithms are based on an analogy with the population's chromosomes' structure and behaviour in terms of genetics. The foundation of GAs based on this comparison is as follows:

1. Population members compete with one another for resources and mate
2. Successful people (the fittest) mate to have more children than less successful people.
3. The "fittest" parent's genes spread across the generation, which means that occasionally parents can produce children that are superior to both of them.

As a result, each next generation becomes more environment-friendly.

4 METHODOLOGIES

The following concepts are used in methodology.

The majority of evolutionary algorithms are adaptive heuristic search algorithms known as genetic algorithms (GAs). Natural selection and genetics are the foundations of genetic algorithms. These are clever uses of random search that are supported by historical data to focus the search on areas with superior performance in the solution space. They are frequently employed to produce excellent answers to optimisation and search-related issues.

Dataset

The anticipated outcome is significantly impacted by the data quality. The data being considered has a major impact on accuracy. The UCI machine learning repository supported the "Pima Indians Diabetes Dataset" standard, which we utilised [12]. The values in this common dataset were taken from the
actual examples.

Size of the dataset: (768, 9)

It has 768 instances, and there are 9 attributes, all of which have numeric values, attached to each instance. The names, descriptions, and value ranges for each attribute are displayed in the table.

Data Pre-processing

The process of gathering and transforming data into the required format is known as preprocessing. The acquired data is likely to be riddled with inaccuracies and may be redundant, inconsistent, noisy, and devoid of particular behaviours or trends. Pre-processing data is a tried-and-true way to fix these problems. Data preprocessing aids in the resolution of this kind of data.

The bioinformatics datasets gathered in the field are highly undeveloped and frequently exhibit the traits listed below. Before using data mining techniques to analyse this data, processing is required.

Data Preprocessing Methods

The likelihood of noise, missing numbers, and consistency issues in raw data is very high. Results of data mining are impacted by the quality of the data. Raw data is pre-processed to increase efficiency and simplify the mining process while also enhancing the quality of the data and the results. Data Pre-processing, which deals with the preparation and transformation of the initial dataset, is one of the most important processes in the data mining process. The following categories are used to group data pre-processing techniques. As follows:

1. Data Cleaning.

2. Data Integration.

3. Data Transformation.

4. Data Reduction.

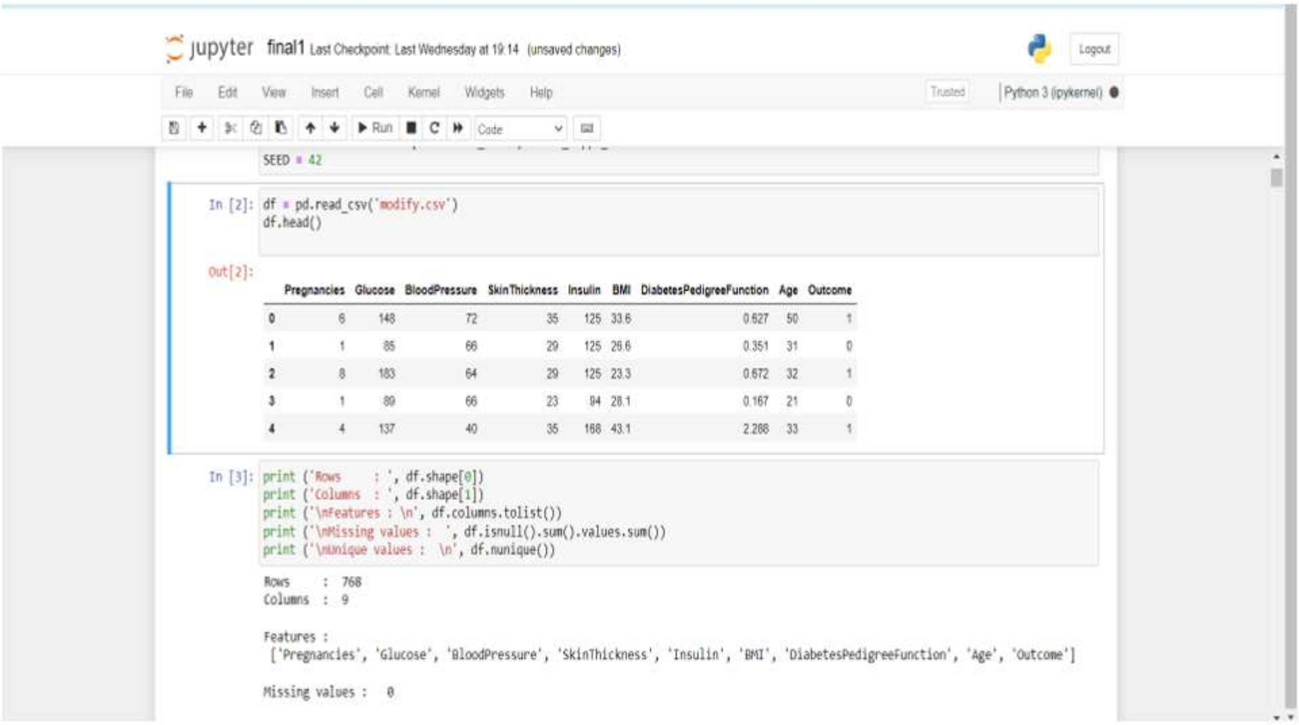
Random Forest Classifier

A multi-tree classifier called Random Forest can be applied to classification and regression problems. The fundamental units of the random forest model are decision trees. It creates a decision tree for each data sample after randomly dividing the data into various samples. Each tree's projections are put up for vote. In classification issues, the class with the most votes will be taken into consideration. The average of all the class predictions will be taken into account in regression issues. The quantity of trees built has a significant impact on the model's accuracy.

Naive Bayes Algorithm

A probabilistic classifier based on the Bayes theorem with naive independence assumptions, nave Bayes is a classification algorithm. The naive Bayesian approach uses the dataset as its input, analyses it, and then extrapolates Bayes' Theorem to forecast the class label. It assists in predicting the class of the ambiguous data sample by calculating the probability of class in the input data. It is an effective classification method that works with enormous datasets. It posits the idea that the features are unrelated to the specified class. The Bayes Theorem offers a method for calculating the likelihood of a hypothesis in light of our prior knowledge.

6 RESULTS AND DISCUSSIONSCREENSHOTS


jupyter
final1 Last Checkpoint: Last Wednesday at 19:14 (unsaved changes)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3 (ipykernel)
Run
Code

```
SEED = 42
```

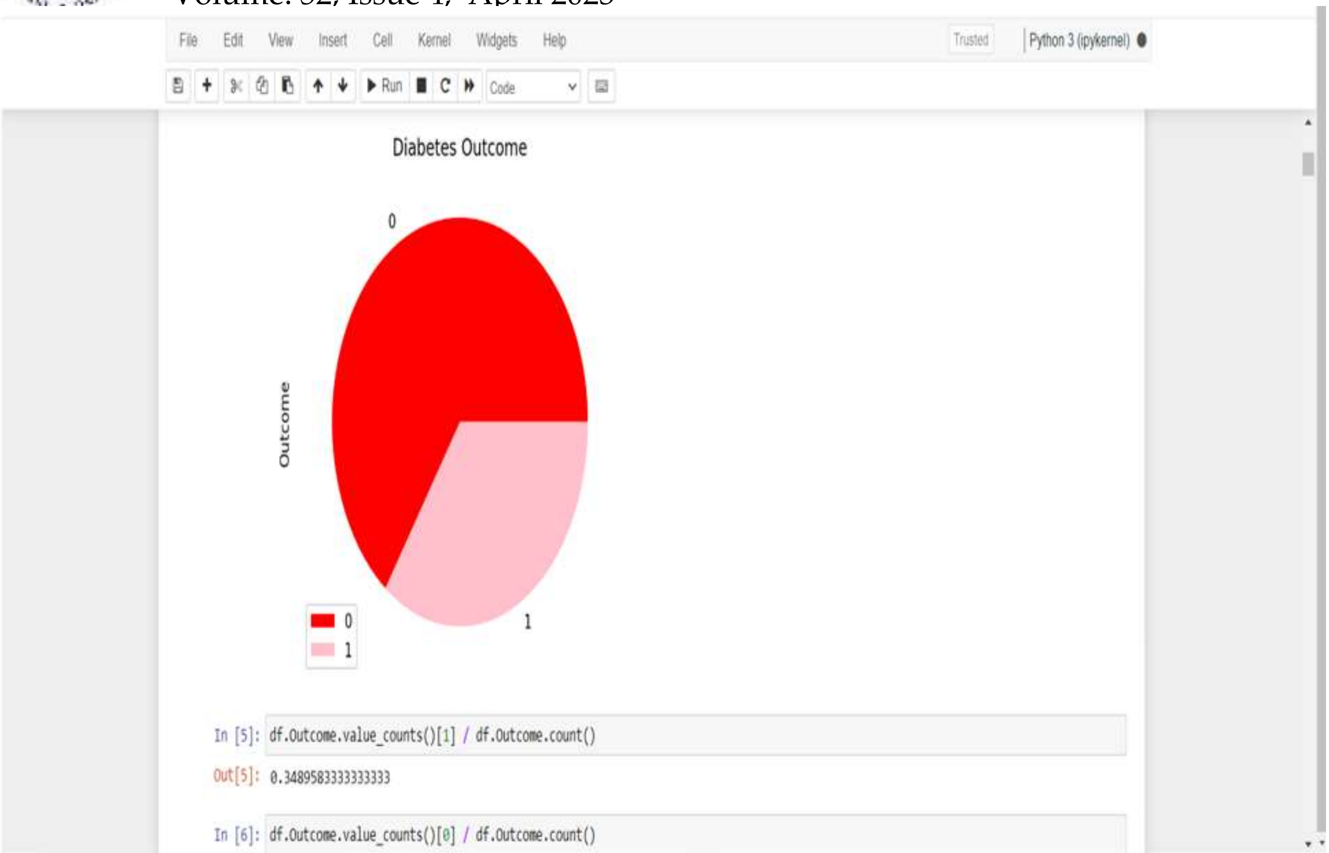
In [2]:
```
df = pd.read_csv('modify.csv')
df.head()
```

Out[2]:

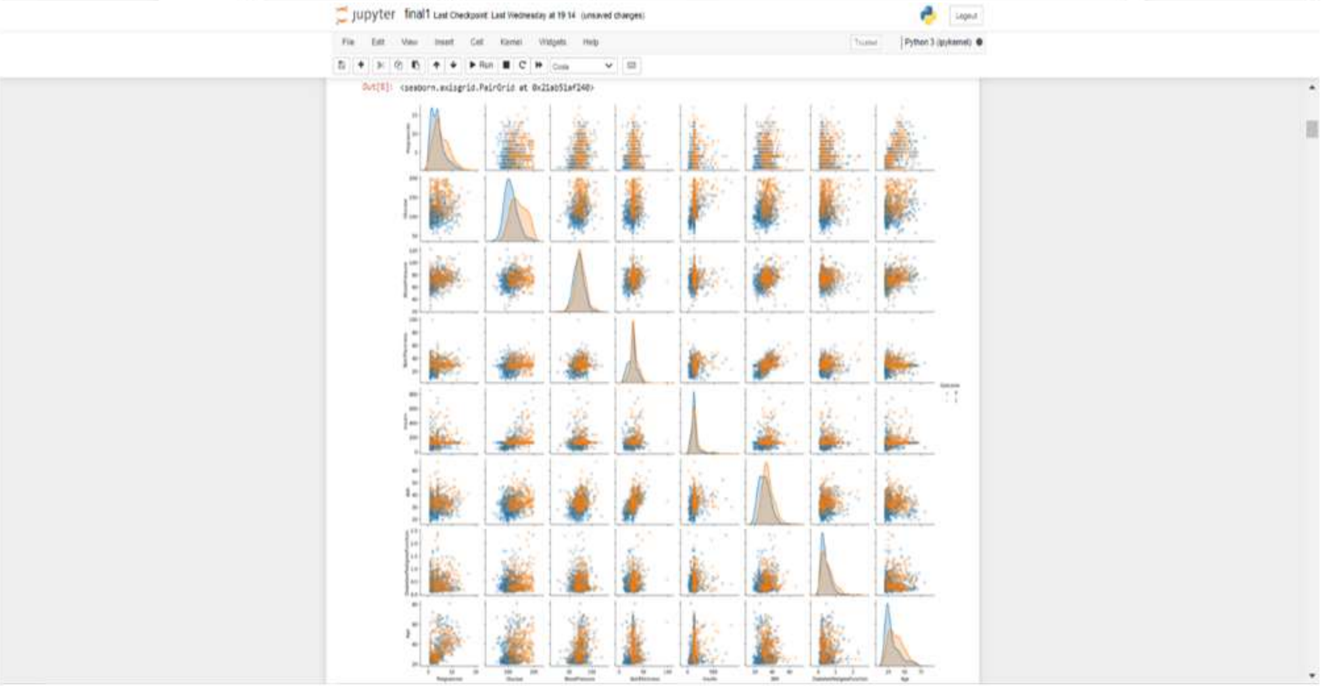
	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
0	6	148	72	35	125	33.6	0.627	50	1
1	1	85	66	29	125	26.6	0.351	31	0
2	8	183	64	29	125	23.3	0.672	32	1
3	1	89	66	23	94	28.1	0.167	21	0
4	4	137	40	35	168	43.1	2.288	33	1

In [3]:
```
print ('Rows     : ', df.shape[0])
print ('Columns  : ', df.shape[1])
print ('\nFeatures : \n', df.columns.tolist())
print ('\nMissing values :  ', df.isnull().sum().values.sum())
print ('\nUnique values :  \n', df.nunique())
```

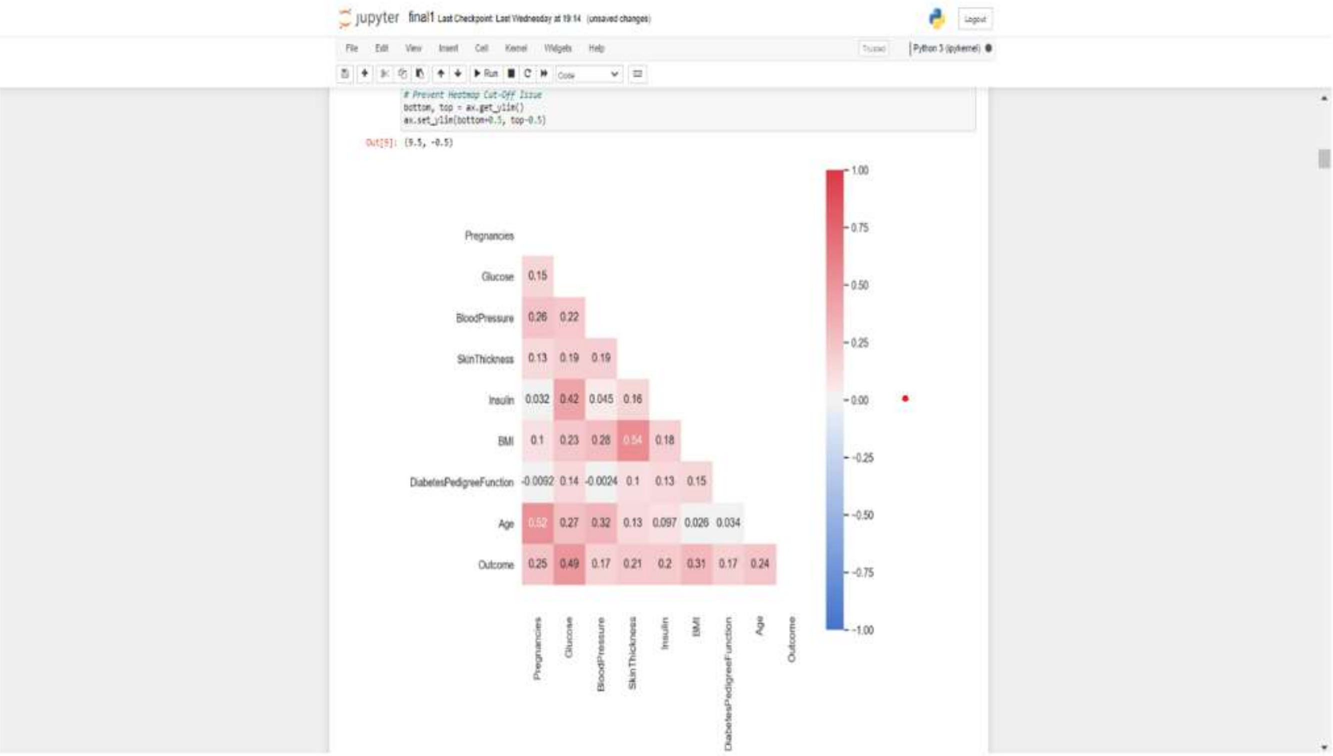
```
Rows     :  768
Columns  :  9

Features :
 ['Pregnancies', 'Glucose', 'BloodPressure', 'SkinThickness', 'Insulin', 'BMI', 'DiabetesPedigreeFunction', 'Age', 'Outcome']

Missing values :   0
```



```
In [5]: df.Outcome.value_counts()[1] / df.Outcome.count()
Out[5]: 0.3489583333333333

In [6]: df.Outcome.value_counts()[0] / df.Outcome.count()
```



```
Out[8]: <seaborn.axisgrid.PairGrid at 0x21ab51af240>
```

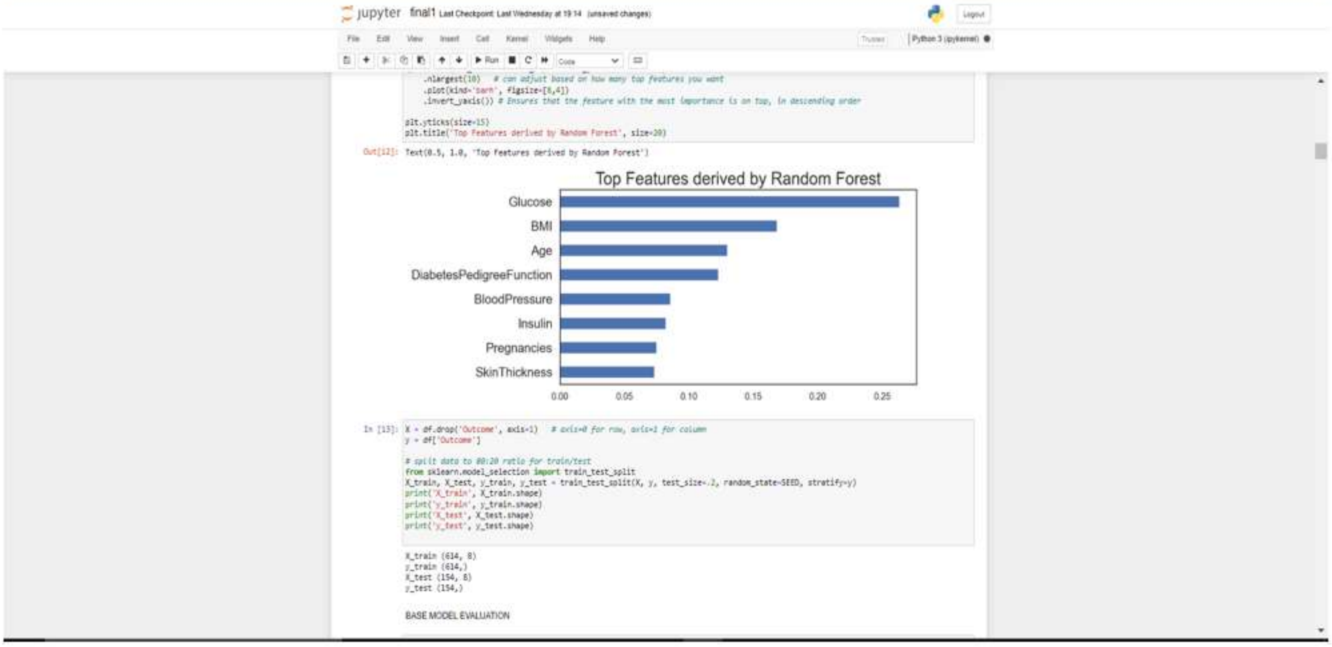


jupyter final1 Last Checkpoint: Last Wednesday at 19:14 (unsaved changes)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3 (ipykernel)
Run
Code

```
# Prevent Heatmap Cut-Off Issue
bottom, top = ax.get_ylim()
ax.set_ylim(bottom+0.5, top-0.5)
```

```
Out[9]: (9.5, -0.5)
```

Pregnancies
Glucose 0.15
BloodPressure 0.26 0.22
SkinThickness 0.13 0.19 0.19
Insulin 0.032 0.42 0.045 0.16
BMI 0.1 0.23 0.28 0.54 0.18
DiabetesPedigreeFunction -0.0092 0.14 -0.0024 0.1 0.13 0.15
Age 0.52 0.27 0.32 0.13 0.097 0.026 0.034
Outcome 0.25 0.49 0.17 0.21 0.2 0.31 0.17 0.24
Pregnancies
Glucose
BloodPressure
SkinThickness
Insulin
BMI
DiabetesPedigreeFunction
Age
Outcome
1.00
0.75
0.50
0.25
0.00
-0.25
-0.50
-0.75
-1.00



jupyter final1 Last Checkpoint: Last Wednesday at 19:14 (unsaved changes)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3 (ipykernel)
Run
Code

```
    .nlargest(10)   # can adjust based on how many top features you want
    .plot(kind='barh', figsize=[8,4])
    .invert_yaxis()) # Ensures that the feature with the most importance is on top, in descending order

plt.yticks(size=15)
plt.title('Top Features derived by Random Forest', size=20)
```

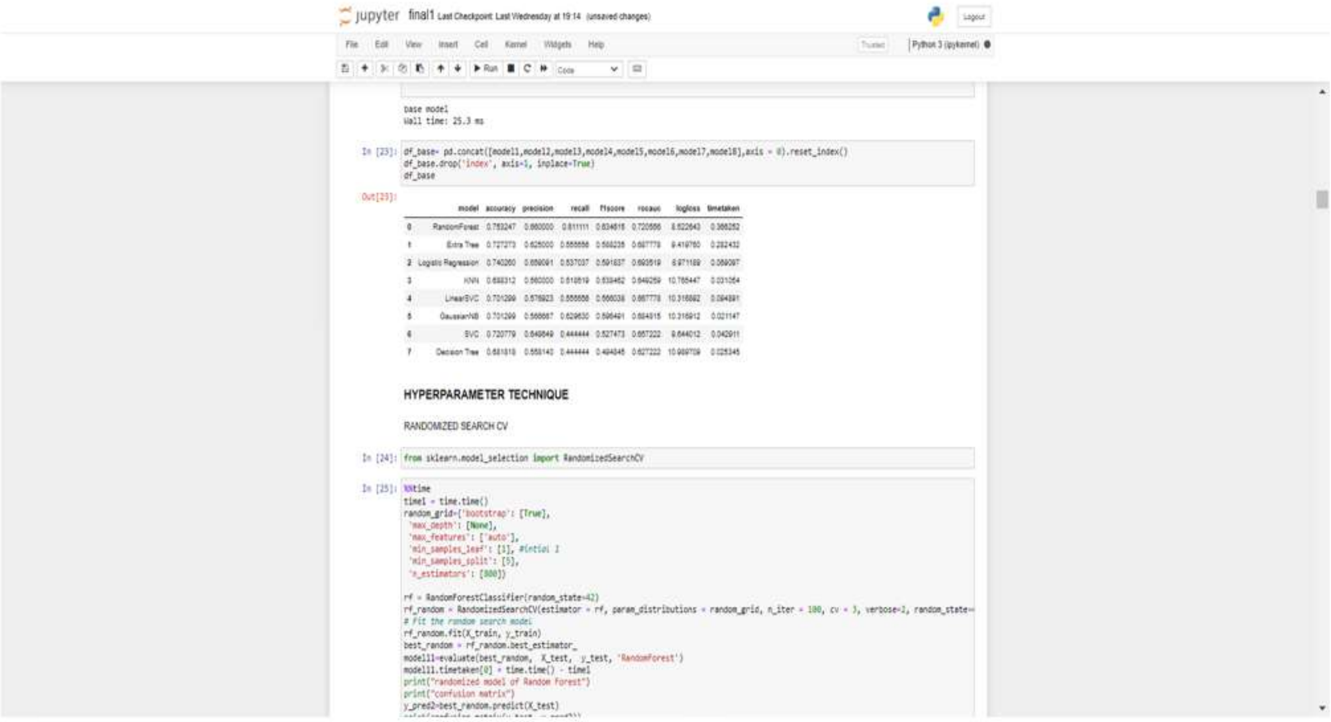
```
Out[12]: Text(0.5, 1.0, 'Top Features derived by Random Forest')
```

Top Features derived by Random Forest
Glucose
BMI
Age
DiabetesPedigreeFunction
BloodPressure
Insulin
Pregnancies
SkinThickness
0.00
0.05
0.10
0.15
0.20
0.25

```
In [13]: X = df.drop('Outcome', axis=1)   # axis=0 for row, axis=1 for column
         y = df['Outcome']

         # split data to 80:20 ratio for train/test
         from sklearn.model_selection import train_test_split
         X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=.2, random_state=SEED, stratify=y)
         print('X_train', X_train.shape)
         print('y_train', y_train.shape)
         print('X_test', X_test.shape)
         print('y_test', y_test.shape)
```

```
X_train (614, 8)
y_train (614,)
X_test (154, 8)
y_test (154,)
```

BASE MODEL EVALUATION


jupyter final1 Last Checkpoint: Last Wednesday at 19:14 (unsaved changes)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3 (ipykernel)
Run
Code

```
base model
Wall time: 25.3 ms
```

```
In [23]: df_base= pd.concat([model1,model2,model3,model4,model5,model6,model7,model8],axis = 0).reset_index()
         df_base.drop('index', axis=1, inplace=True)
         df_base
```

Out[23]:

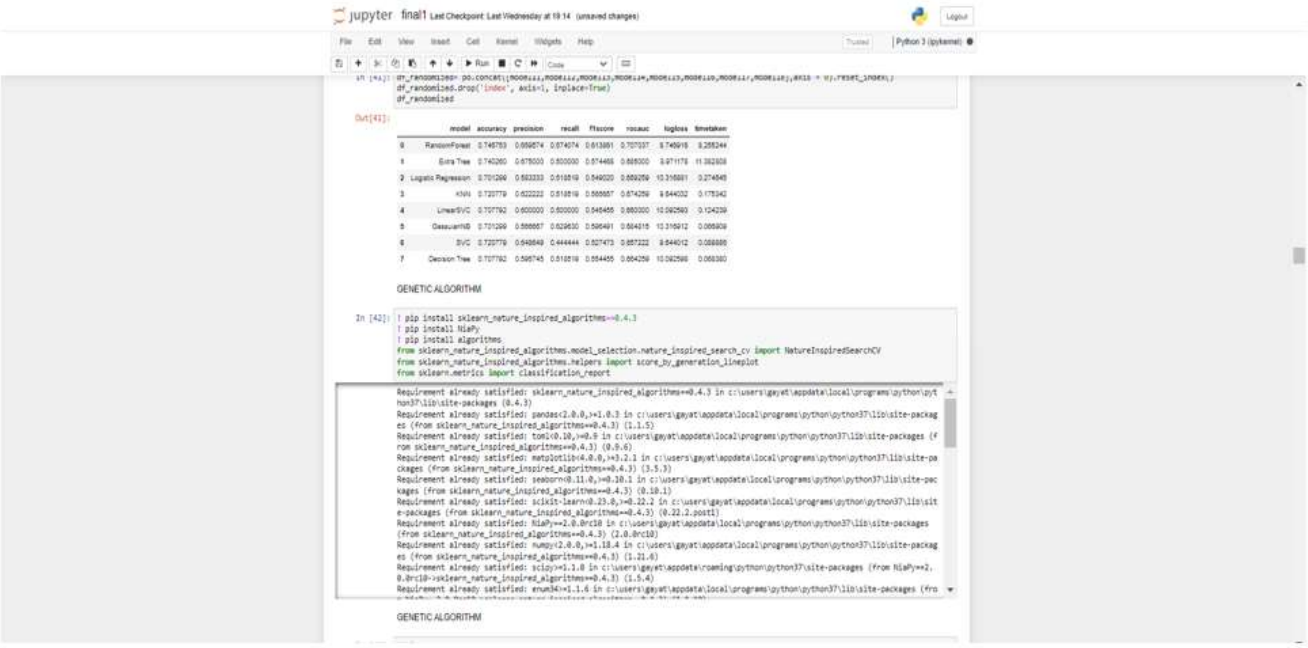
	model	accuracy	precision	recall	f1score	rocauc	logloss	timetaken
0	RandomForest	0.753247	0.660000	0.611111	0.634615	0.720556	8.522643	0.366252
1	Extra Tree	0.727273	0.625000	0.555556	0.588235	0.687778	9.419760	0.282432
2	Logistic Regression	0.740260	0.659091	0.537037	0.591837	0.693519	8.971189	0.059097
3	KNN	0.688312	0.560000	0.518519	0.538462	0.649259	10.765447	0.031064
4	LinearSVC	0.701299	0.576923	0.555556	0.566038	0.667778	10.316682	0.094891
5	GaussianNB	0.701299	0.566667	0.629630	0.596491	0.684815	10.316912	0.021147
6	SVC	0.720779	0.648649	0.444444	0.527473	0.657222	9.644012	0.042911
7	Decision Tree	0.681818	0.558140	0.444444	0.494845	0.627222	10.989709	0.025345

HYPERPARAMETER TECHNIQUE

RANDOMIZED SEARCH CV

```
In [24]: from sklearn.model_selection import RandomizedSearchCV
```

```
In [25]: %%time
         time1 = time.time()
         random_grid={'bootstrap': [True],
          'max_depth': [None],
          'max_features': ['auto'],
          'min_samples_leaf': [1], #intial 1
          'min_samples_split': [5],
          'n_estimators': [800]}

         rf = RandomForestClassifier(random_state=42)
         rf_random = RandomizedSearchCV(estimator = rf, param_distributions = random_grid, n_iter = 100, cv = 3, verbose=2, random_state=
         # Fit the random search model
         rf_random.fit(X_train, y_train)
         best_random = rf_random.best_estimator_
         model11=evaluate(best_random,  X_test,  y_test, 'RandomForest')
         model11.timetaken[0] = time.time() - time1
         print("randomized model of Random Forest")
         print("confusion matrix")
         y_pred2=best_random.predict(X_test)
```


jupyter final1 Last Checkpoint: Last Wednesday at 19:14 (unsaved changes)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3 (ipykernel)
Run
Code

```
In [41]: df_randomized= pd.concat([model11,model12,model13,model14,model15,model16,model17,model18],axis = 0).reset_index()
         df_randomized.drop('index', axis=1, inplace=True)
         df_randomized
```

Out[41]:

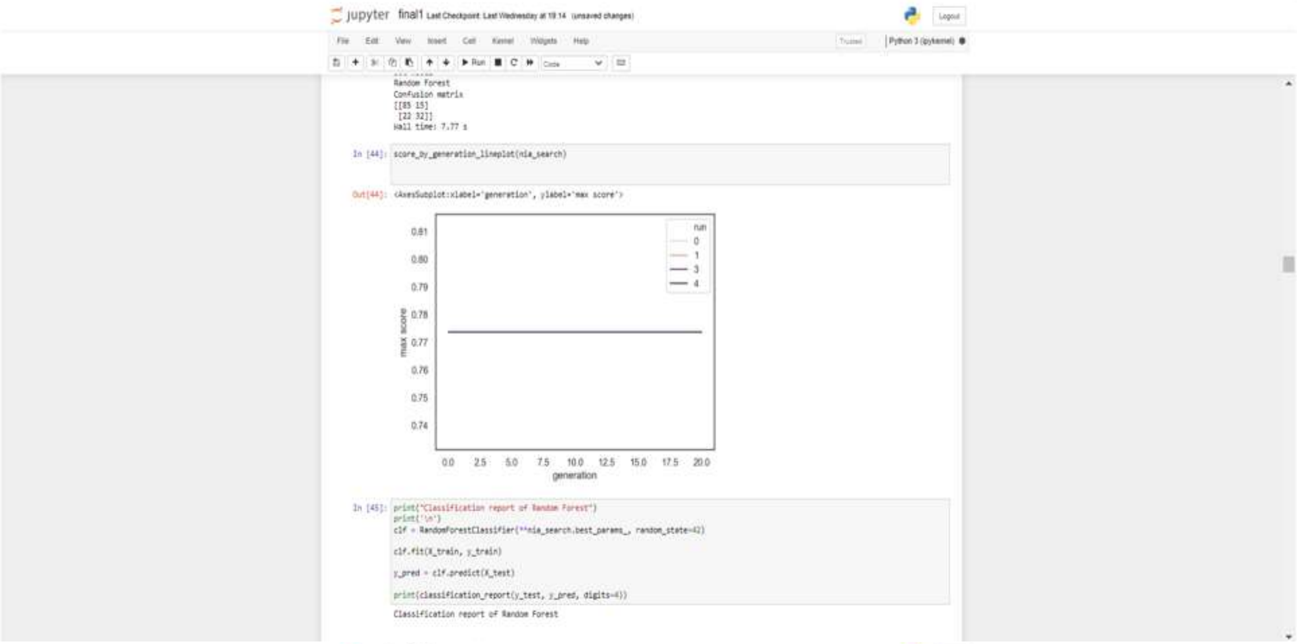
	model	accuracy	precision	recall	f1score	rocauc	logloss	timetaken
0	RandomForest	0.746753	0.659574	0.574074	0.613861	0.707037	8.745915	3.255244
1	Extra Tree	0.740260	0.675000	0.500000	0.574468	0.685000	8.971175	11.382808
2	Logistic Regression	0.701299	0.583333	0.518519	0.549020	0.669259	10.316881	0.274846
3	KNN	0.720779	0.622222	0.518519	0.565657	0.674259	9.644032	0.175342
4	LinearSVC	0.707792	0.600000	0.500000	0.545455	0.660000	10.092580	0.124239
5	GaussianNB	0.701299	0.566667	0.629630	0.596491	0.684815	10.316912	0.066909
6	SVC	0.720779	0.648649	0.444444	0.527473	0.657222	9.644012	0.088886
7	Decision Tree	0.707792	0.595745	0.518519	0.554455	0.664259	10.092598	0.068380

GENETIC ALGORITHM

```
In [42]: ! pip install sklearn_nature_inspired_algorithms==0.4.3
         ! pip install NiaPy
         ! pip install algorithms
         from sklearn_nature_inspired_algorithms.model_selection.nature_inspired_search_cv import NatureInspiredSearchCV
         from sklearn_nature_inspired_algorithms.helpers import score_by_generation_lineplot
         from sklearn.metrics import classification_report
```

```
Requirement already satisfied: sklearn_nature_inspired_algorithms==0.4.3 in c:\users\gayat\appdata\local\programs\python\pyt
hon37\lib\site-packages (0.4.3)
Requirement already satisfied: pandas<2.0.0,>=1.0.3 in c:\users\gayat\appdata\local\programs\python\python37\lib\site-packag
es (from sklearn_nature_inspired_algorithms==0.4.3) (1.1.5)
Requirement already satisfied: toml<0.10,>=0.9 in c:\users\gayat\appdata\local\programs\python\python37\lib\site-packages (f
rom sklearn_nature_inspired_algorithms==0.4.3) (0.9.6)
Requirement already satisfied: matplotlib<4.0.0,>=3.2.1 in c:\users\gayat\appdata\local\programs\python\python37\lib\site-pa
ckages (from sklearn_nature_inspired_algorithms==0.4.3) (3.5.3)
Requirement already satisfied: seaborn<0.11.0,>=0.10.1 in c:\users\gayat\appdata\local\programs\python\python37\lib\site-pac
kages (from sklearn_nature_inspired_algorithms==0.4.3) (0.10.1)
Requirement already satisfied: scikit-learn<0.23.0,>=0.22.2 in c:\users\gayat\appdata\local\programs\python\python37\lib\sit
e-packages (from sklearn_nature_inspired_algorithms==0.4.3) (0.22.2.post1)
Requirement already satisfied: NiaPy==2.0.0rc10 in c:\users\gayat\appdata\local\programs\python\python37\lib\site-packages
(from sklearn_nature_inspired_algorithms==0.4.3) (2.0.0rc10)
Requirement already satisfied: numpy<2.0.0,>=1.18.4 in c:\users\gayat\appdata\local\programs\python\python37\lib\site-packag
es (from sklearn_nature_inspired_algorithms==0.4.3) (1.21.6)
Requirement already satisfied: scipy>=1.1.0 in c:\users\gayat\appdata\roaming\python\python37\site-packages (from NiaPy==2.
0.0rc10->sklearn_nature_inspired_algorithms==0.4.3) (1.5.4)
Requirement already satisfied: enum34>=1.1.6 in c:\users\gayat\appdata\local\programs\python\python37\lib\site-packages (fro
```

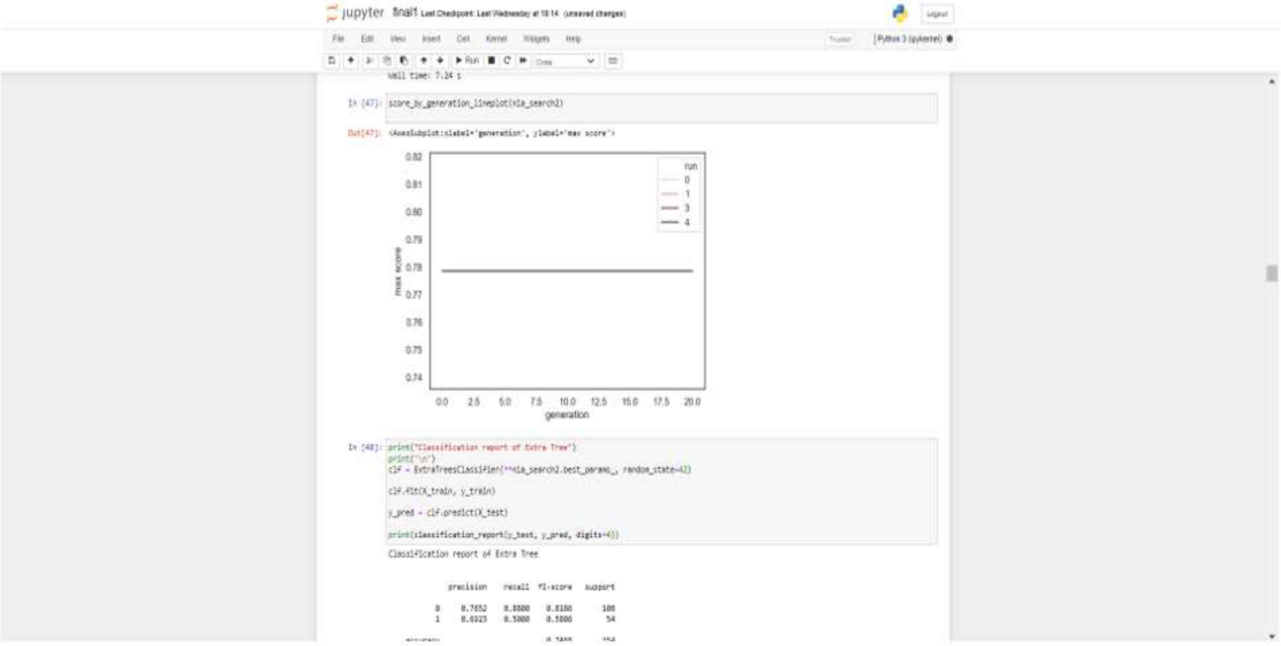
GENETIC ALGORITHM



```
Random Forest
Confusion matrix
[[85 15]
 [22 32]]
Wall time: 7.77 s
```

```
In [44]: score_by_generation_lineplot(nia_search)

Out[44]: <AxesSubplot:xlabel='generation', ylabel='max score'>
```

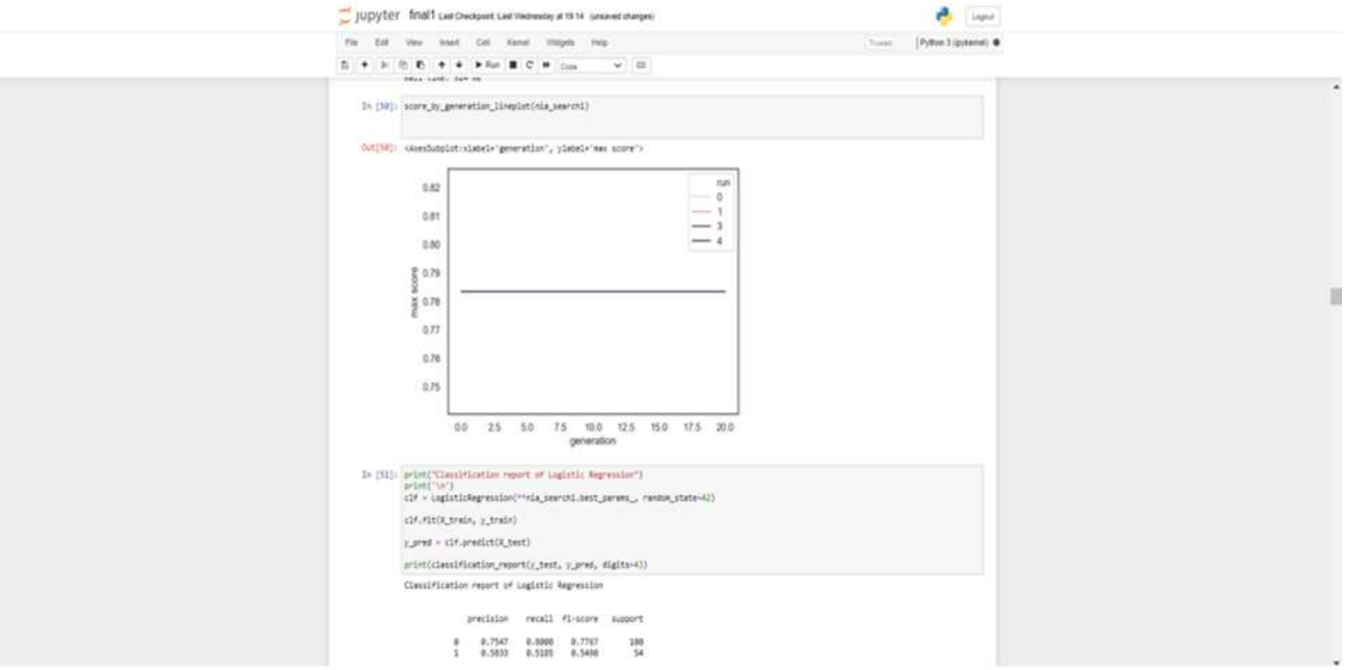
```
In [45]: print("Classification report of Random Forest")
         print('\n')
         clf = RandomForestClassifier(**nia_search.best_params_, random_state=42)

         clf.fit(X_train, y_train)

         y_pred = clf.predict(X_test)

         print(classification_report(y_test, y_pred, digits=4))

         Classification report of Random Forest
```



```
Wall time: 7.24 s
```

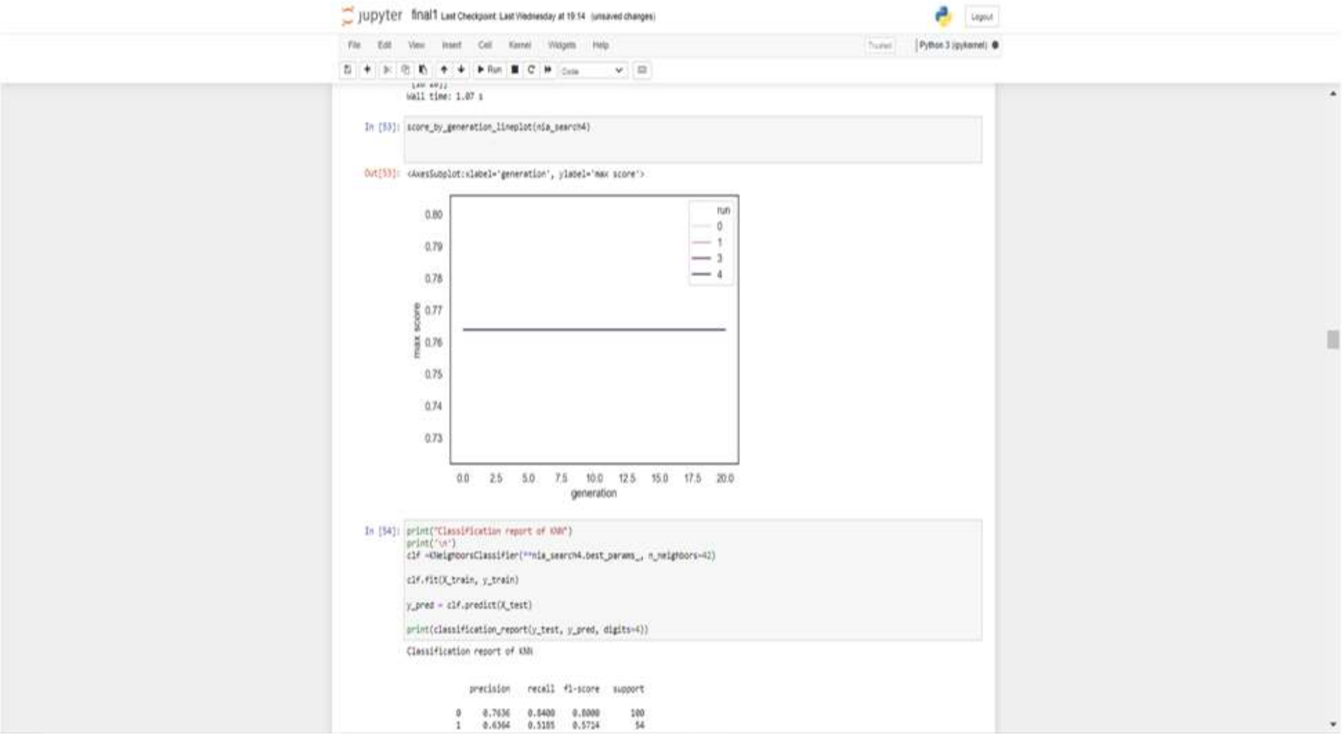
```
In [47]: score_by_generation_lineplot(nia_search2)

Out[47]: <AxesSubplot:xlabel='generation', ylabel='max score'>
```

```
In [48]: print("Classification report of Extra Tree")
         print('\n')
         clf = ExtraTreesClassifier(**nia_search2.best_params_, random_state=42)

         clf.fit(X_train, y_train)

         y_pred = clf.predict(X_test)

         print(classification_report(y_test, y_pred, digits=4))

         Classification report of Extra Tree


                      precision    recall  f1-score   support

                   0     0.7652    0.8800    0.8186       100
                   1     0.6923    0.5000    0.5806        54
```



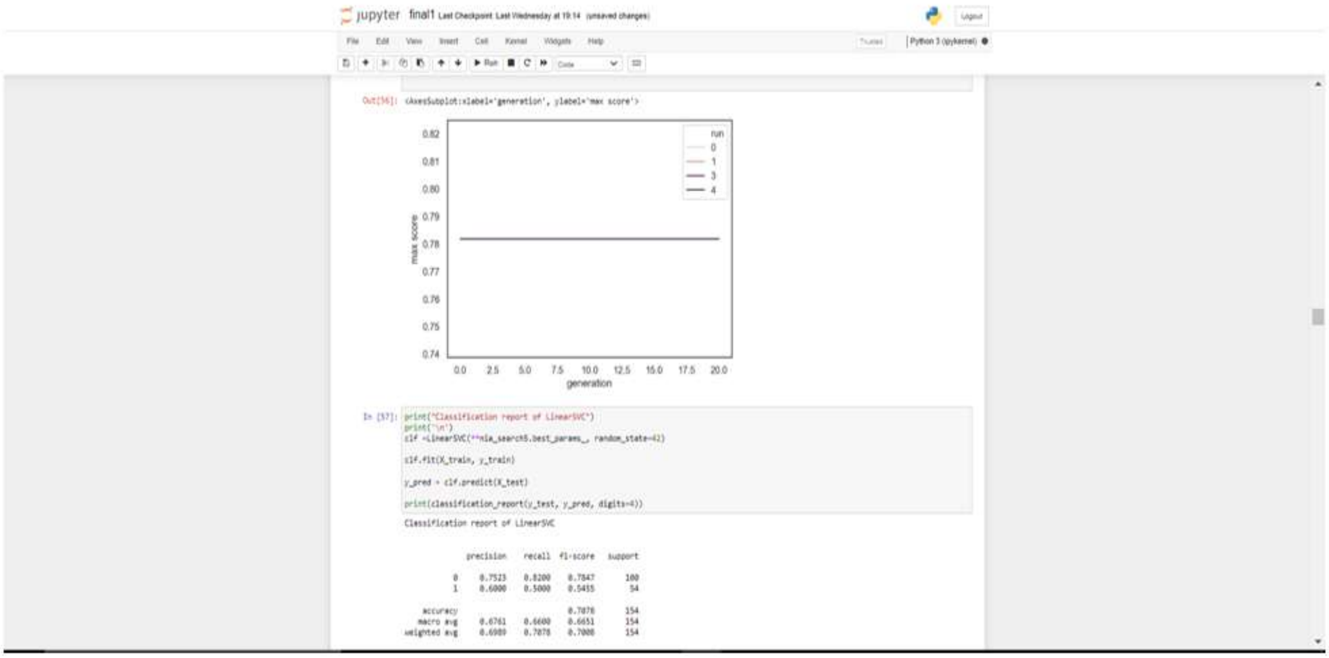
```
In [50]: score_by_generation_lineplot(nia_search1)

Out[50]: <AxesSubplot:xlabel='generation', ylabel='max score'>
```

```
In [51]: print("Classification report of Logistic Regression")
         print('\n')
         clf = LogisticRegression(**nia_search1.best_params_, random_state=42)

         clf.fit(X_train, y_train)

         y_pred = clf.predict(X_test)

         print(classification_report(y_test, y_pred, digits=4))

         Classification report of Logistic Regression


                       precision    recall  f1-score   support

                    0     0.7547    0.8000    0.7767       100
                    1     0.5833    0.5185    0.5490        54
```



```
         Wall time: 1.07 s

In [53]: score_by_generation_lineplot(nia_search4)

Out[53]: <AxesSubplot:xlabel='generation', ylabel='max score'>
```

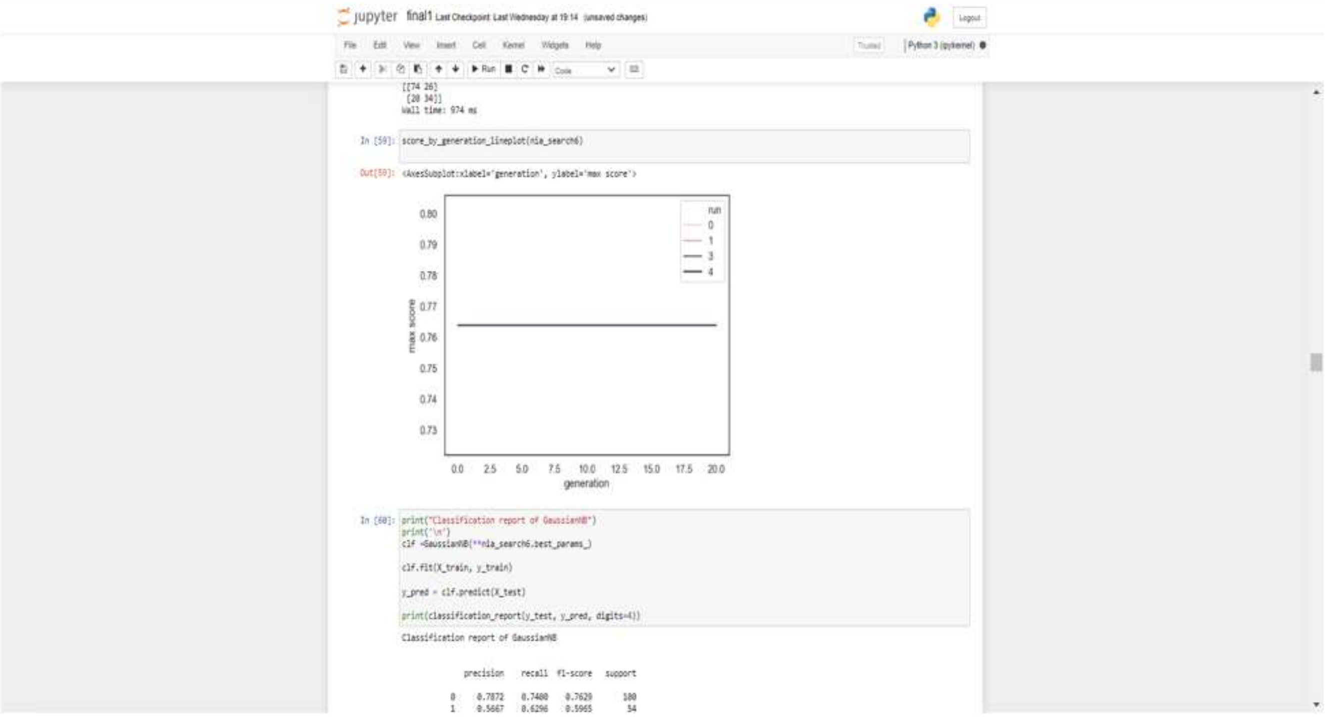
```
In [54]: print("Classification report of KNN")
         print('\n')
         clf =KNeighborsClassifier(**nia_search4.best_params_, n_neighbors=42)

         clf.fit(X_train, y_train)

         y_pred = clf.predict(X_test)

         print(classification_report(y_test, y_pred, digits=4))

         Classification report of KNN


                       precision    recall  f1-score   support

                    0     0.7636    0.8400    0.8000       100
                    1     0.6364    0.5185    0.5714        54
```



```
Out[56]: <AxesSubplot:xlabel='generation', ylabel='max score'>
```

```
In [57]: print("Classification report of LinearSVC")
         print('\n')
         clf =LinearSVC(**nia_search5.best_params_, random_state=42)

         clf.fit(X_train, y_train)

         y_pred = clf.predict(X_test)

         print(classification_report(y_test, y_pred, digits=4))
```

```
Classification report of LinearSVC


              precision    recall  f1-score   support

           0     0.7523    0.8200    0.7847       100
           1     0.6000    0.5000    0.5455        54

    accuracy                         0.7078       154
   macro avg     0.6761    0.6600    0.6651       154
weighted avg     0.6989    0.7078    0.7008       154
```



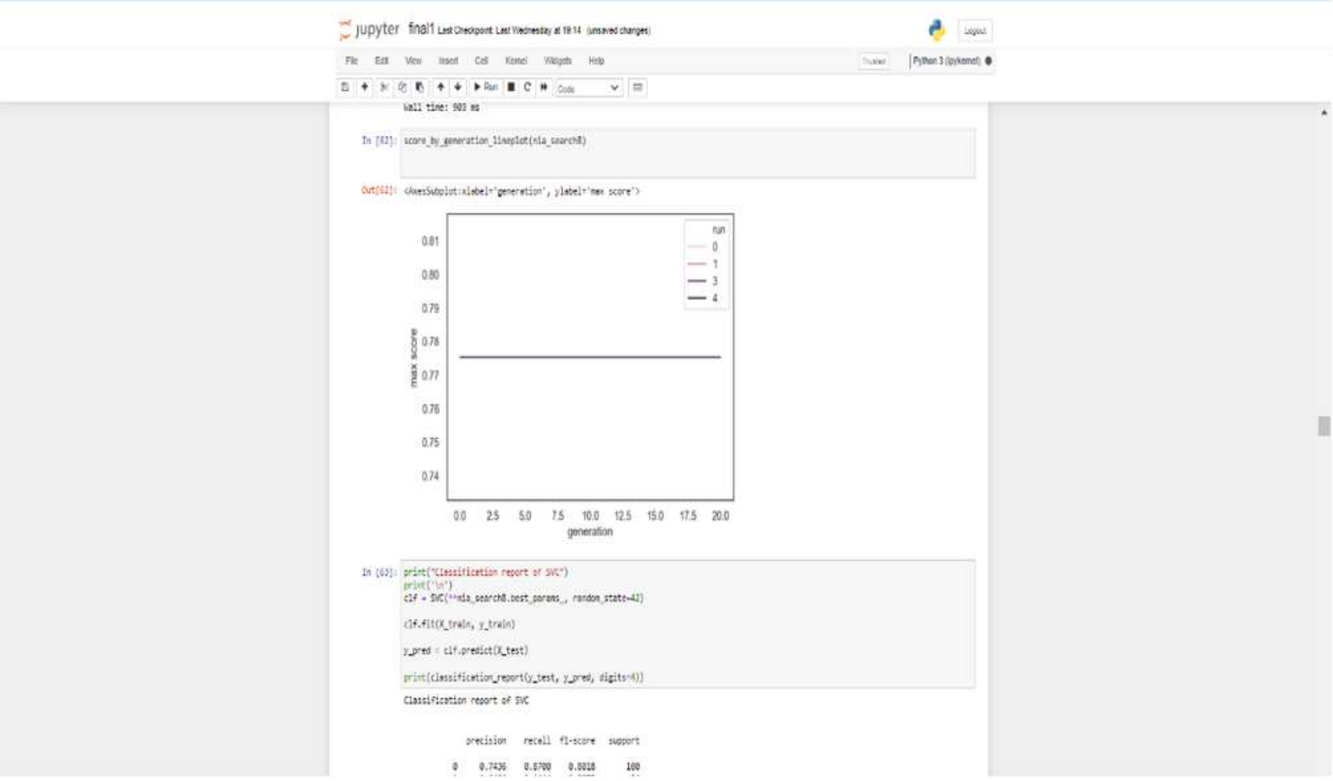
```
[[74 26]
 [20 34]]
Wall time: 974 ms
```

```
In [59]: score_by_generation_lineplot(nia_search6)
```

```
Out[59]: <AxesSubplot:xlabel='generation', ylabel='max score'>
```

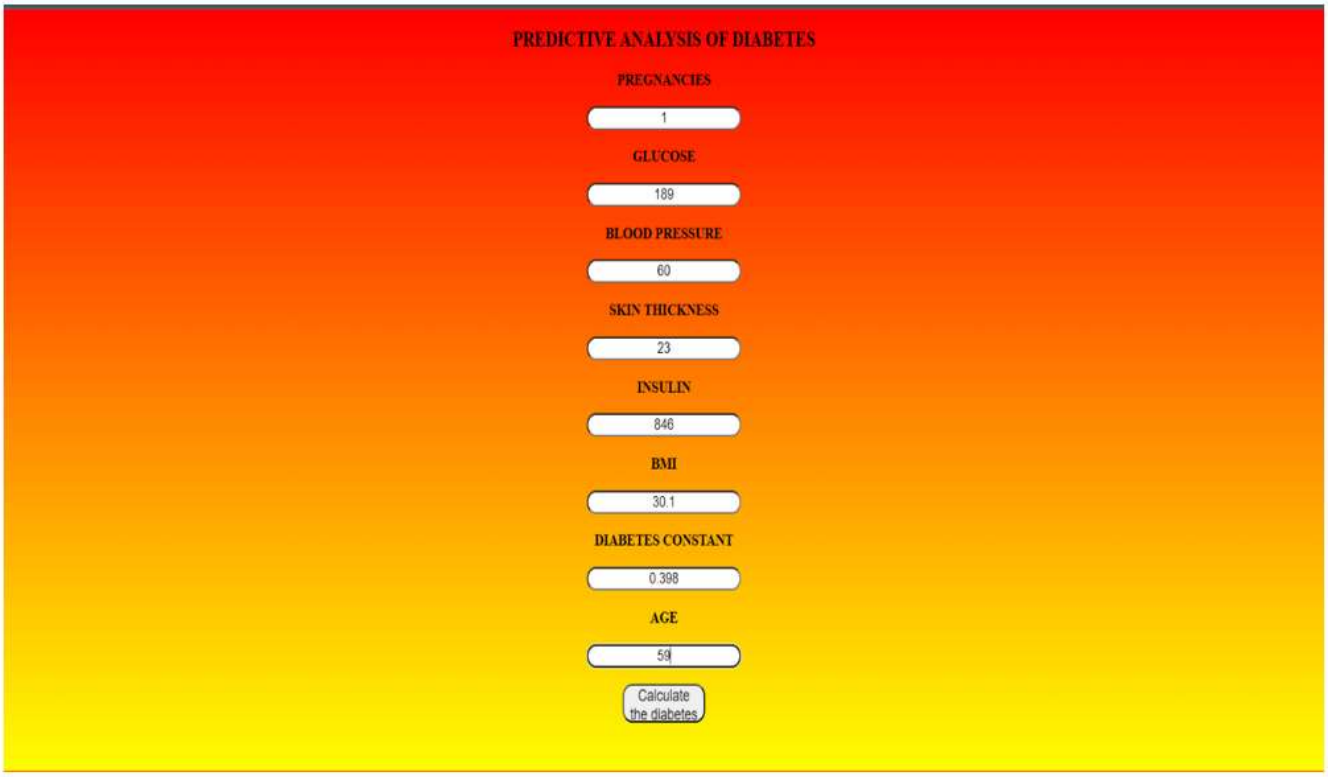
```
In [60]: print("Classification report of GaussianNB")
         print('\n')
         clf =GaussianNB(**nia_search6.best_params_)

         clf.fit(X_train, y_train)

         y_pred = clf.predict(X_test)

         print(classification_report(y_test, y_pred, digits=4))
```

```
Classification report of GaussianNB


              precision    recall  f1-score   support

           0     0.7872    0.7400    0.7629       100
           1     0.5667    0.6296    0.5965        54
```



```
Wall time: 903 ms

In [62]: score_by_generation_lineplot(nia_search8)

Out[62]: <AxesSubplot:xlabel='generation', ylabel='max score'>
```

```
In [63]: print("Classification report of SVC")
         print('\n')
         clf = SVC(**nia_search8.best_params_, random_state=42)

         clf.fit(X_train, y_train)

         y_pred = clf.predict(X_test)

         print(classification_report(y_test, y_pred, digits=4))

         Classification report of SVC


                       precision    recall  f1-score   support

                    0     0.7436    0.8700    0.8018       100
```



PREDICTIVE ANALYSIS OF DIABETES

PREGNANCIES

1

GLUCOSE

189

BLOOD PRESSURE

60

SKIN THICKNESS

23

INSULIN

846

BMI

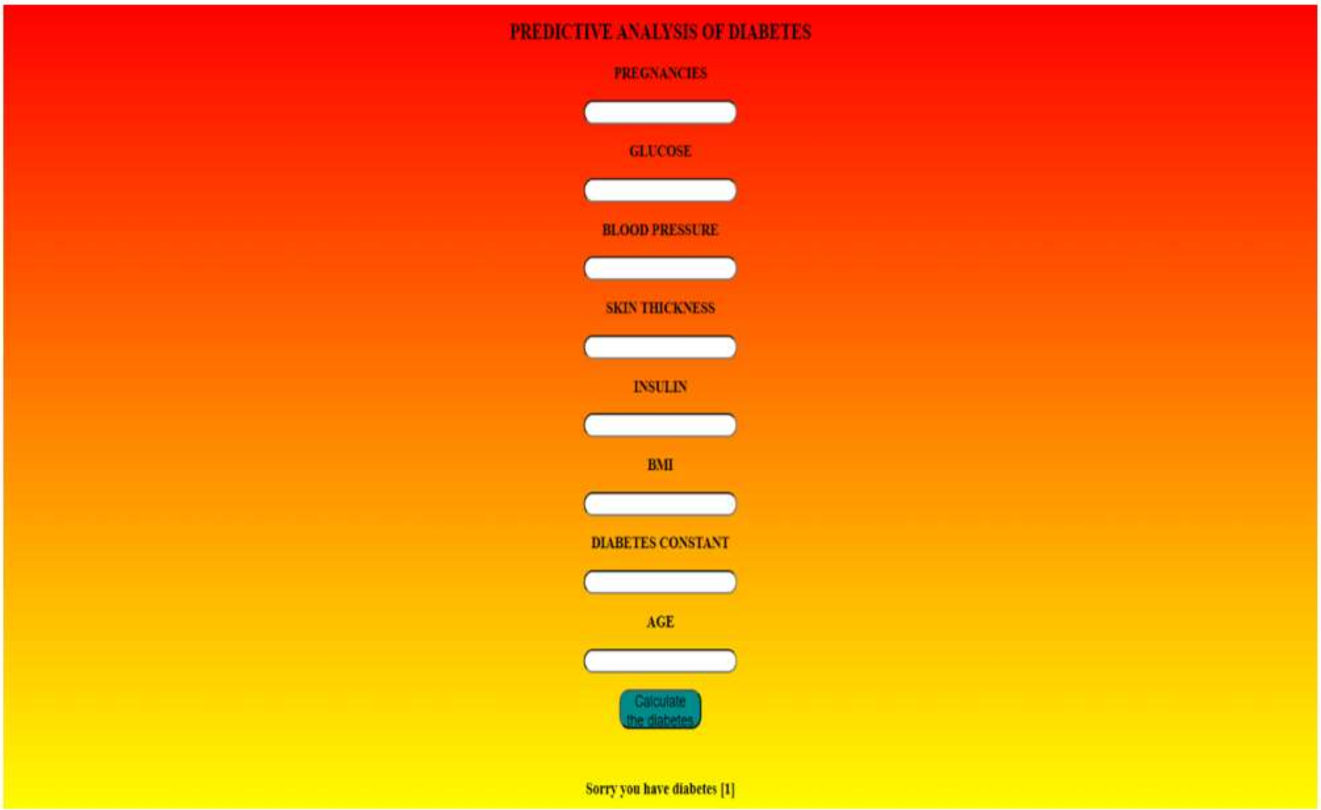
30.1

DIABETES CONSTANT

0.398

AGE

59



PREDICTIVE ANALYSIS OF DIABETES

PREGNANCIES

GLUCOSE

BLOOD PRESSURE

SKIN THICKNESS

INSULIN

BMI

DIABETES CONSTANT

AGE

Sorry you have diabetes [1]

7. CONCLUSION AND FUTURE WORK

Conclusion

In order to better the lives of people everywhere, we are working to identify and stop diabetic complications before they become serious using predictive analysis and improved categorization methods. Also, the features in the dataset are analysed in our suggested work, and the best features are chosen based on correlation values. The goal of the paper is to present a model that more accurately categorises the dataset's instances. Methods like feature selection and data cleansing have improved the dataset's potential. The accuracy of each classifier is greater than 67%. SVM has the highest accuracy (77%), while Decision Tree has the lowest accuracy (67%), placing SVM at the top of the list. Of all of the classifiers, last place. Each combination is subjected to cross-validation in order to determine the mean accuracy of each model. SVM has a 78% accuracy rate, while Logistic Regression has a 77% accuracy rate. In addition to cross-validation, we also used a machine learning toolkit. This model was developed utilising the WEKA toolkit and the same Pima Indian Diabetes Dataset in order to establish a valid comparison with other researchers' results. When compared to the WEKA tool, every classifier has produced accuracy levels that are higher. In terms of evaluation criteria like precision, recall, f1 score, and specificity, SVM and naive Bayes are comparatively better. When compared to other models, SVM has a lower mean absolute error. By way of comparison, we identify SVM as the most accurate model for the dataset involving people with and without diabetes.

Future Work

More advanced algorithms will be used in future studies to improve accuracy. Gaussian, multinomial, and Bernoulli's classifiers are the main components of naive Bayes. Just the Gaussian naive Bayes classifier has been implemented. For the Bayes classifier comparative research, the other types can also be used. Linear, non-linear, polynomial, sigmoid, and other kernels are included in SVM. As linear kernels classify our dataset the best, we used them. It is also possible to use the additional kernels to improve categorization. The Random Forest Classifier is used to identify future candidates. To improve the model's accuracy, other selection techniques including wrapped, embedded, and filters can also be applied. designing a programme that will

Provide the high-risk group logical and sensible health advice. For ongoing training and model optimisation, hospital-based real-time patient data must be incorporated. It appears that the dataset is unbalanced. The dataset should have a sufficient amount of data to support both training and prediction. Utilizing tools like neural networks could increase the classification accuracy of the dataset.

8. REFERENCES

[1] World Health Organization Global Report on Diabetes 2019.https://www.who.int/news-room/fact-sheets/detail/diabetes

[2] International Diabetes Federation. https://www.idf.org/aboutdiabetes/what-is-diabetes/facts-figures.html

[3] http://www.healthdata.org/sites/default/files/files/2017_India_State- Level_Disease_Burden_Initiative_-Full_Report%5B1%5D.pdf

[4] The increasing burden of diabetes and variations among the states of India: the Global Burden of Disease Study 1990–2016. https://doi.org/10.1016/S2214- 109X(18)30387-5

[5] Kumari, V, Chitra, R. Classification of diabetes disease using support vector machine. Int J Eng Res Appl. 2013;3(2):1797-1801.

[6] Han Wu, Shengqi Yang, Zhangqin Huang, Jian He, Xiaoyi Wang, Type 2 diabetes mellitus prediction model based on data mining, Informatics in Medicine Unlocked,Volume 10,2018,Pages 100-107,ISSN 2352-9148,

[7] Sneha, N., Gangil, T. Analysis of diabetes mellitus for early prediction using optimal features selection. *J Big Data* 6, 13 (2019). https://doi.org/10.1186/s40537- 019-0175-6

[8] Dutta, Debadri & Paul, Debpriyo & Ghosh, Parthajeet. (2018). Analysing Feature Importances for Diabetes Prediction using Machine Learning. 924-928. 10.1109/IEMCON.2018.8614871.

[9] Sharma, Himani & Kumar, Sunil. (2016). A Survey on Decision Tree Algorithms of Classification in Data Mining. International Journal of Science and Research (IJSR). 5.

[10] Guo, Gongde & Wang, Hui & Bell, David & Bi, Yaxin. (2004). KNN Model- Based Approach in Classification.

[11] Kaviani, Pouria & Dhotre, Sunita. (2017). Short Survey on Naive Bayes Algorithm. International Journal of Advance Research in Computer Science and Management. 04.

[12] Evgeniou, Theodoros & Pontil, Massimiliano. (2001). Support Vector Machines: Theory and Applications. 2049. 249-257. 10.1007/3-540-44673-7_12.

[13] Peng, Joanne & Lee, Kuk & Ingersoll, Gary. (2002). An Introduction to Logistic Regression Analysis and Reporting. Journal of Educational Research - J EDUC RES. 96. 3-14. 10.1080/00220670209598786.

[14] Vrushali Y Kulkarni,Pradeep K Sinha,Effective Learning and Classification using Random Forest Algorithm, nternational Journal of Engineering and Innovative Technology (IJEIT), Volume 3, Issue 11, May 2014, ISSN: 2277-3754.

[15] Hina S, Shaikh A, Sattar SA. Analyzing diabetes datasets using data mining. J Basic Appl Sci. 2017;13:466–71.

[16] Sa'di S, Maleki A, Hashemi R, Panbechi Z, Chalabi K. Comparison Of Data Mining Algorithms In The Diagnosis Of Type II diabetes. International Journal on Computational Science & Applications (IJCSA) 2015; 5(5).

[17] https://towardsdatascience.com/20-popular-machine-learning-metrics-part-1- classification-regression-evaluation-metrics-1ca3e282a2ce

[18] Machine Learning Group at the University of Waikato. Weka 3: Data Mining Software in Java. Retrieved September 4,2016, from http://www.cs.waikato.ac.nz/ml/weka/

[19] Pima Indians Diabetes Data Set. http://networkrepository.com/pima-indians- diabetes.php

[20] Zafar, Faizan & Raza, Saad & Khalid, Muhammad & Tahir, Muhammad. (2019). Predictive Analytics in Healthcare for Diabetes Prediction. 253-259. 10.1145/3326172.3326213.