Diabetes Analysis Using Machine Learning Algorithms

Kadeejath Salaha [1], Adheena Ranjith[2], Annapoorna Shetty[3]
[1],[2],[3] Department of IT, AIMIT, Mangalore, Karnataka, India
kadeejasalaha123@gmail.com, adheenaranjith46@gmail.com, annapoorna@staloysius.ac.in

Abstract*— *Diabetes is a chronic metabolic disease caused by high blood sugar that can damage the heart, blood vessels, eyes, kidneys, and nerves over time. The purpose of this analysis is to create a system that can predict a person's risk of developing diabetes. To determine the best model for diabetes testing, we investigated three classification algorithms: Random Forest (RF), Light Gradient Boosting Machine (Light GBM), and Extreme Gradient Boosting (XGB). Comparative analysis reveals differences in the accuracy of these models. The aim of the project is not only to provide the accurate model, but also to demonstrate the effectiveness of machine learning in predicting diabetes. After rigorous analysis, our results show that the XG Booster ML model achieves an accuracy of 90.13%, outperforming similar models. This shows the superiority of XG Booster in predicting the risk of diabetes.

***Keywords* –** *diabetes, machine learning, prediction, dataset.*

I. INTRODUCTION

Diabetes is a chronic metabolic disease caused by high blood sugar levels and is a global health problem. According to the World Health Organization (WHO), approximately 463 million adults had diabetes in 2019, and this number is expected to increase to 700 million by 2045. The impact of diabetes on a healthy person, lifestyle and health has revealed the urgent need. For effective diagnosis and management strategies. Early diagnosis of diabetes is important for timely intervention and thus reducing the risk of complications such as heart disease, kidney failure and vision problems.

1.1 Types of Diabetes

Type 1 Diabetes: Type 1 diabetes usually begins in childhood or adolescence and is caused by autoimmune damage to insulin-producing beta cells in the pancreas. When the body cannot produce insulin or produces too little insulin. An autoimmune process, in which the immune system mistakenly targets cells that produce insulin, is typically the cause of type 1 diabetes. Type 1 diabetes affects about 10% of those with diabetes. Diabetes Type 2: The most prevalent kind of diabetes, type 2, is typically diagnosed later in life and is brought on by a combination of genetics, physical inactivity, and weight. The body produces too little insulin or suppresses its effects. This is caused by two defects: insulin resistance and the inability to produce enough insulin to overcome this resistance. Type 2 diabetes is the most common type of diabetes, accounting for 80-90% of diabetics worldwide. Gestational Diabetes: This type of temporary diabetes occurs due to hormonal changes during pregnancy that affect the action of insulin. This condition usually goes away after birth but increases the risk of type 2 diabetes later in life.

1.2 Symptoms

Diabetes is a type of metabolic disease caused by high blood sugar due to insufficient insulin or insulin secretion. The body cannot use insulin effectively. The symptoms of diabetes are diverse and common. It is important to remember that not everyone with diabetes will experience all of these symptoms, and some may not even notice any symptoms. Diabetes most commonly manifests as increased thirst (polydipsia), frequent urine (polyuria), increased hunger (polyphagia), exhaustion, blurred eyesight, and slower wound healing.

II. RELATED WORKS

Muhammad Exell Febrian et al. Using machine learning supervision, the authors used two k-nearest neighbors(KNN) and Naive Bayes algorithm to predict diabetes based on various health factors in the dataset.After carefully analyzing the results and using the fuzzy algorithm to evaluate the matrix,
We found that the Naive Bayes algorithm is more accurate than the ANN algorithm in predicting diabetes. Damacharla Vidyasagar Rao et al. This study includes the use of optimization methods, support vector machines and logistic regression algorithms in predicting diabetes. As a result of the overall evaluation, we

found that the conversion boost algorithm outperforms support vector machine and logistic regression predictions in terms of accuracy.

Victor Chan et al., in this study, we described supervised machine learning models (Naive Bayes classifier, random forest classifier and J48 decision tree model) used in the R programming language. Pima Indian Diabetes Factbook< br> was meticulously evaluated. Analysis results show that the Naive Bayes algorithm has the highest accuracy among the three models.

KM Jyoti Rani found a way to create diabetes in this work. Five introduced machine learning classification techniques, including K-nearest neighbors, logistic regression, random forests, support vector machines, and decision trees, and confirmed their use of the John Diabetes database. The purpose of this study is to test the algorithms using different metrics. The test results clearly demonstrated the effectiveness of the design, and an accuracy of 99% was achieved using the decision tree algorithm. Among the tested algorithms, the decision tree algorithm performed better than other algorithms.

Aishwarya Mujumdar et al., in this study used a series of machine learning algorithms for the dataset, allowing a clear analysis of the classification process, especially logistic regression, reaching 96%. value. . Additionally, the AdaBoost classifier using the pipeline proved to be the best model with 98.8% accuracy. In this study, they compared the accuracy of learning algorithms by considering two different models. This result shows the importance of algorithm selection, where logistic regression and AdaBooststand out as the best predictive power in this study.

III. METHODOLOGY

In this section, you will learn about the different tools used in machine learning to predict diabetes. We also discuss our plans to improve accuracy. Five different methods were used in this study. The various methods used are described below. Output is the accurate measurement for machine learning models. This model can be used to make predictions.

3.1 Dataset Description

The secondary data used in this study were originally collected from Kaggle.com, an online site maintained by the National Institute of Diabetes and Digestive and Kidney Diseases, and the same data were also used for analysis. The goal is to use these tests to determine whether a patient has diabetes. This database contains a total of 768 cases, each containing 8 clinical parameters and 1 target variable.

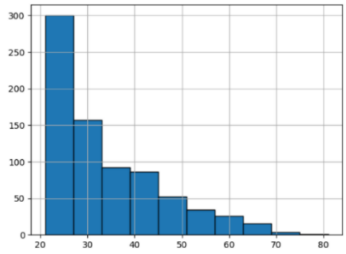
	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
0	6	148	72	35	0	33.6	0.627	50	1
1	1	85	66	29	0	26.6	0.351	31	0
2	8	183	64	0	0	23.3	0.672	32	1
3	1	89	66	23	94	28.1	0.167	21	0
4	0	137	40	35	168	43.1	2.288	33	1

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 768 entries, 0 to 767
Data columns (total 9 columns):
 #   Column                    Non-Null Count  Dtype
---  ------                    --------------  -----
 0   Pregnancies               768 non-null    int64
 1   Glucose                   768 non-null    int64
 2   BloodPressure             768 non-null    int64
 3   SkinThickness             768 non-null    int64
 4   Insulin                   768 non-null    int64
 5   BMI                       768 non-null    float64
 6   DiabetesPedigreeFunction  768 non-null    float64
 7   Age                       768 non-null    int64
 8   Outcome                   768 non-null    int64
dtypes: float64(2), int64(7)
memory usage: 54.1 KB
```

(Fig 1)

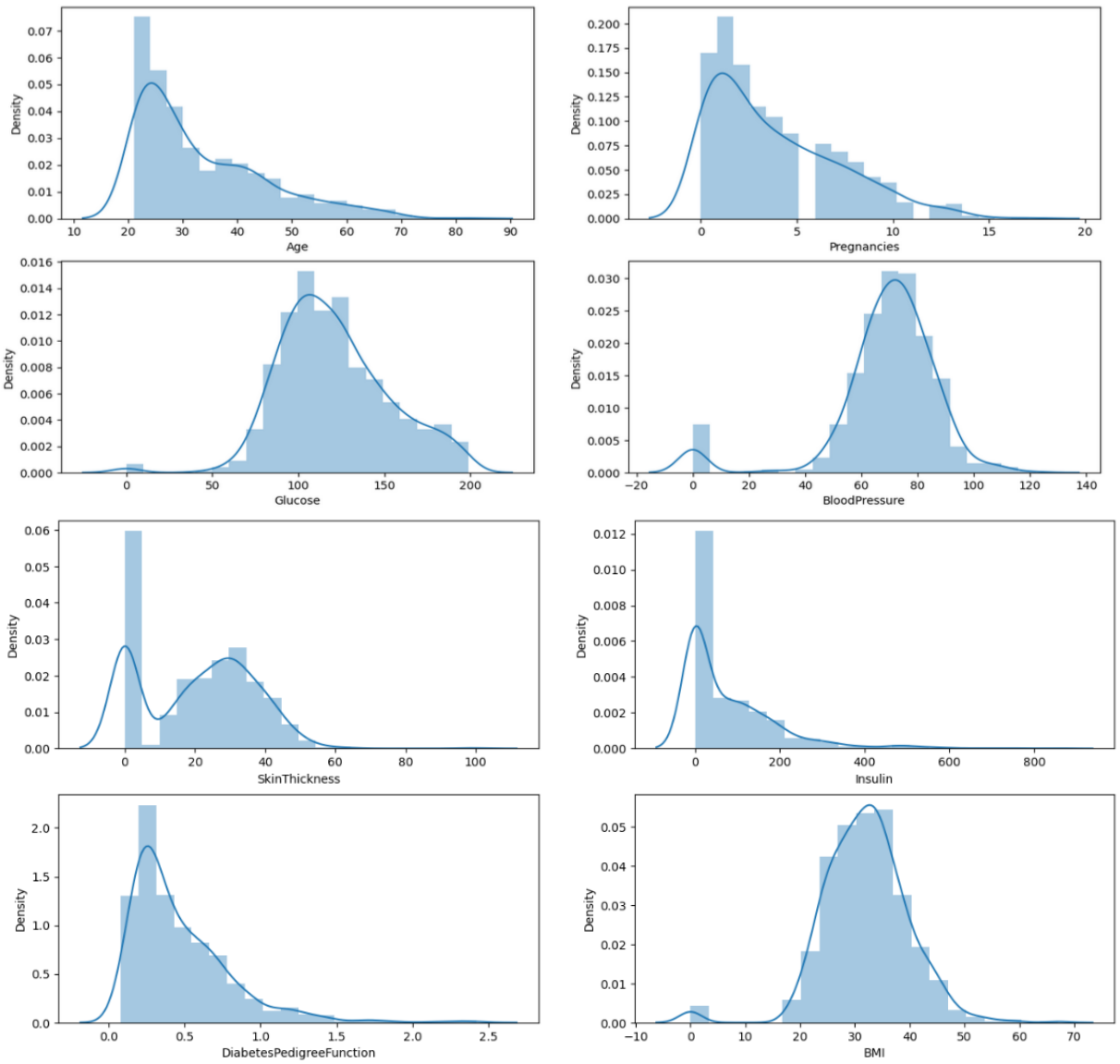
3.1 Visualization

3.2.1 Histogram



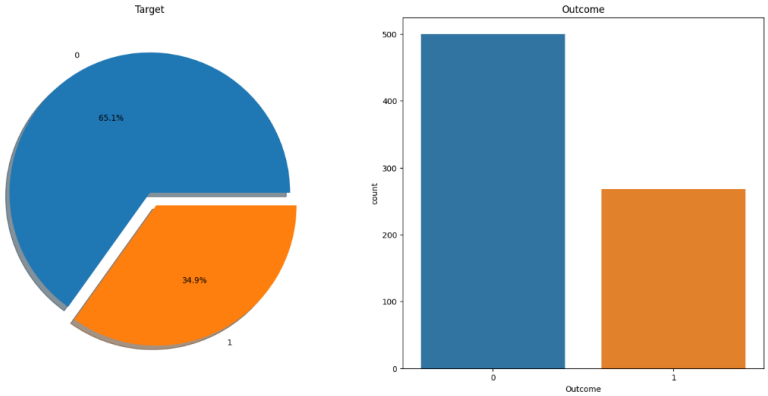
The histagram of the Age variable

```
<Axes: xlabel='BMI', ylabel='Density'>
```



Histogram and density graphs of all variables were accessed.

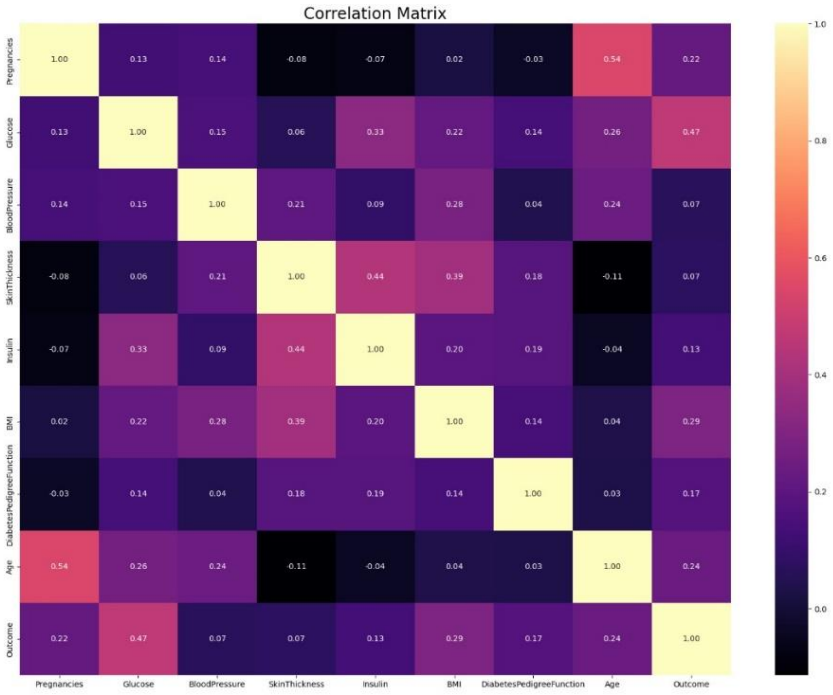
3.2.2 Bar Plot for Outcome Class



Access configuration success information. What is the relationship between different topics?

- If the correlation value > 0, there is similarity. When the value of one variable increases, the value of the other variable also increases.
- Correlation = 0 means there is no correlation.
- If the correlation is <0, it means there is a negative correlation. As you go up, the difference decreases.
- When examining the relationship, there are 2 variables that are similar to salary, depending on the variable.
- These changes are glucose. As these increase, different values also increase.

3.2.3 Correlation Matrix



Correlation between all the features before cleaning

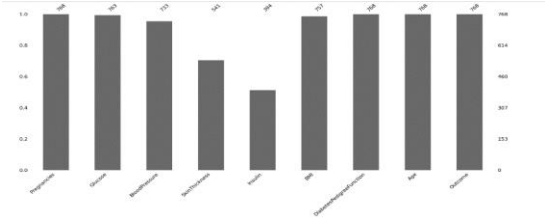
3.2 Data Preprocessing

3.3.1 Missing Observation Analysis

In df.head() I see that some features contain 0, which doesn't make sense here and indicates missing values. The following replaces the 0 value with NaN:

```
Pregnancies                 0
Glucose                     5
BloodPressure              35
SkinThickness             227
Insulin                   374
BMI                        11
DiabetesPedigreeFunction    0
Age                         0
Outcome                     0
dtype: int64
```

3.3.2 Visualizing missing observations using the missingno library



In the image above, we can clearly see that the dataset does not contain null values.

3.3.3 The missing values will be filled with the median values of each variable:

The median value for those who are not sick and the median value for those who are sick are given as values. Due to incomplete observation.

[31]:

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
0	6	148.0	72.0	35.0	169.5	33.6	0.627	50	1
1	1	85.0	66.0	29.0	102.5	26.6	0.351	31	0
2	8	183.0	64.0	32.0	169.5	23.3	0.672	32	1
3	1	89.0	66.0	23.0	94.0	28.1	0.167	21	0
4	0	137.0	40.0	35.0	168.0	43.1	2.288	33	1

[32]:

```
Pregnancies                 0
Glucose                     0
BloodPressure               0
SkinThickness               0
Insulin                     0
BMI                         0
DiabetesPedigreeFunction    0
Age                         0
Outcome                     0
dtype: int64
```

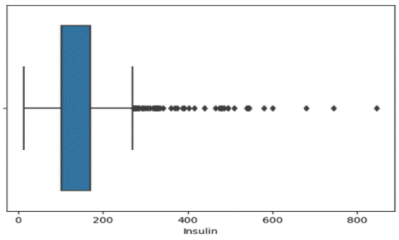
Here, we first check whether there is a null value in the IsNull() function according to the output above. We then take the sum of all missing values from the sum() function and conclude that no values are missing.

3.3 Outlier Observation Analysis

We asked whether there were any outlier observations in the dataset compared to the 25% and 75% quartiles. This was determined to be an outlier observation

```
Pregnancies yes
Glucose no
BloodPressure yes
SkinThickness yes
Insulin yes
BMI yes
DiabetesPedigreeFunction yes
Age yes
Outcome no
```

3.3.1 The process of visualizing the Insulin variable with boxplot

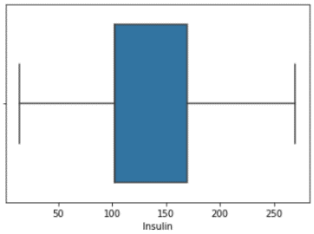


By using seaborn library

When you can observe the distribution of your data and with the use of a boxplot, Distplot is helpful in this way.In that column, you can also view outliers and other data that can be obtained from the boxplot.

3.3.2 Standalone Observation

We conduct a standalone observation review for the Insulin variable by suppressing contradictory values



3.5 Local Outlier Factor(LOF)

Use the LOF method to identify outliers between all variables.Then, we selected the threshold according to the IOF score and removed those exceeding the threshold based on it.And the size of the dataset has also been reduced.(That is, the size is reduced from (768,9).

```
(760, 9)
```

3.6 Feature Engineering

Creating new variables is important to the model. However, you will need to create a new logical variable. Several new variables were created in this dataset, corresponding to BMI, insulin, and glucose variables. According to BMI, several ranges were identified and categorical variables were assigned.

[43]:

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome	NewBMI
0	6	148.0	72.0	35.0	169.5	33.6	0.627	50	1	Obesity 1
1	1	85.0	66.0	29.0	102.5	26.6	0.351	31	0	Overweight
2	8	183.0	64.0	32.0	169.5	23.3	0.672	32	1	Normal
3	1	89.0	66.0	23.0	94.0	28.1	0.167	21	0	Overweight
4	0	137.0	40.0	35.0	168.0	43.1	2.288	33	1	Obesity 3

3.7 One Hot Encoding

Categorical variables in the dataset must be converted to numbers .

[49]:

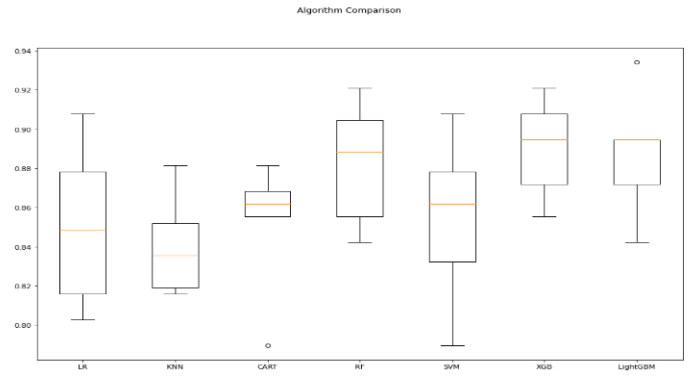
	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome	NewBMI_Obesity 1	NewBMI_Obesity 2	NewBMI_Obesity 3
0	6	148.0	72.0	35.0	169.5	33.6	0.627	50	1	1	0	0
1	1	85.0	66.0	29.0	102.5	26.6	0.351	31	0	0	0	0
2	8	183.0	64.0	32.0	169.5	23.3	0.672	32	1	0	0	0
3	1	89.0	66.0	23.0	94.0	28.1	0.167	21	0	0	0	0
4	0	137.0	40.0	35.0	168.0	43.1	2.288	33	1	0	0	1

NewBMI_Obesity 3	NewBMI_Overweight	NewBMI_Underweight	NewInsulinScore_Normal	NewGlucose_Low	NewGlucose_Normal	NewGlucose_Overweight	NewGlucose_Secret
0	0	0	0	0	0	0	1
0	1	0	1	0	1	0	0
0	0	0	0	0	0	0	1
0	1	0	1	0	1	0	0
1	0	0	0	0	0	0	1

3.8 Base Model

Validation scores of all base models

```
LR: 0.848684 (0.036866)
KNN: 0.840789 (0.023866)
CART: 0.857895 (0.024826)
RF: 0.881579 (0.026316)
SVM: 0.853947 (0.036488)
XGB: 0.890789 (0.020427)
LightGBM: 0.885526 (0.024298)
```



3.9 Methods

i. Random Forest –

This is a type of ensemble learning method that is also used for classification and regression tasks.The other he has lower accuracy compared to the two models.This method allows you to easily process huge amounts of data. Random forests improve performance of the decision trees by reducing

variance..In order to express the mode of the class, the classification, or the average prediction (regression) of each decision tree, it builds several decision trees during training.

Algorithm –

•The initial phase involves choosing features “R” from all features “m”, where R<\<M.

• The node with the best split point among the "R" features

• Using the best split method, divide the node into sub nodes.

• Continue steps a through c until the "l" number of nodes is attained.

• Created a forest by going through steps a through d "a" times, which resulted in "n" trees.

By applying the Gin-Index Cost Function, which is provided by: the random forest determines the optimal split.

$$Gini = \sum_{k=1}^{n} p_k * (1 - p_k) \; Where \; k = Each \; class \; and \; p = proption \; of \; training \; instances$$

The first stage entails scanning the options and using the fundamentals of each arbitrarily constructed decision tree to forecast the outcome, storing it periodically at the desired location. Next, determine which predicted target had the most votes for each one. This will allow you to determine which forecasted target received the highest votes based on the random forest formula's final prediction.

ii. XGBooster

The most effective ensemble and classification technique for prediction is gradient boosting. Compile weekly learners to create a potent prediction learner model. Apply a model of decision trees. This is a popular and highly efficient technique for categorizing intricate datasets. Iterations using gradient boosting enhance model performance.

Algorithm –

• Let P be a sample of the goal values.

• Calculate the target value's error.

Weights should be updated and adjusted to lower error M.

• P[x] = Alpha M[x] + p[x]

• The loss function F computes and analyzes the model learner.

• Continue doing so until you achieve the intended outcome.

iii. Light GBM

Microsoft created the gradient boosting framework known as Light GBM, or Light Gradient Boosting Machine. Light GBM is an ensemble learning technique used for both classification and regression applications, much as Random Forest and XGBoost. It is particularly renowned for its effectiveness and quickness, and huge data sets are advised to use it.

Algorithm –

• Data Split: First, A training set and a validation set are created from the data set.

• Gradient-based learning: Light GBM uses a gradient-based learning approach that focuses on reducing error in predicted values.

• Growing the tree leaf by leaf: Unlike the gradual tree growth of other algorithms, Light GBM grows the tree leaf by leaf. Select and grow leaves with the highest delta loss to increase efficiency.

• Feature Parallelism: This algorithm supports feature parallelism and can efficiently process large numbers of features.

• Processing categorical features: Light GBM has the ability to directly process categorical features, which eliminates the need for one-hot encoding and makes it more practical to use in real data sets.

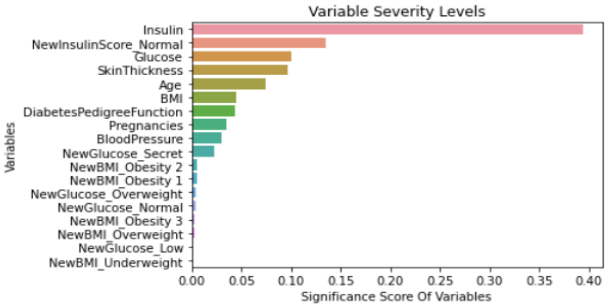
• Boost Iterations: Boost iterations continue until a predefined number is reached or a certain condition is met.

3.10 Model Tuning

```
Fitting 10 folds for each of 192 candidates, totalling 1920 fits

[Parallel(n_jobs=-1)]: Using backend LokyBackend with 4 concurrent workers.
[Parallel(n_jobs=-1)]: Done  33 tasks      | elapsed:   10.4s
[Parallel(n_jobs=-1)]: Done 154 tasks      | elapsed:   48.2s
[Parallel(n_jobs=-1)]: Done 357 tasks      | elapsed:  1.9min
[Parallel(n_jobs=-1)]: Done 640 tasks      | elapsed:  3.5min
[Parallel(n_jobs=-1)]: Done 1005 tasks      | elapsed:  5.7min
[Parallel(n_jobs=-1)]: Done 1450 tasks      | elapsed:  8.4min
[Parallel(n_jobs=-1)]: Done 1920 out of 1920 | elapsed: 11.4min finished
```

Random Forests Tuning

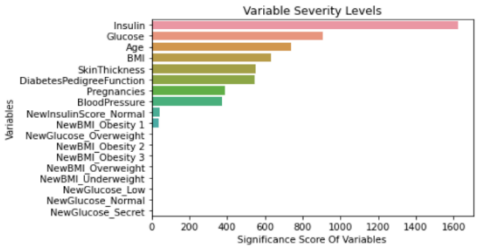


Final Model Installation

```
Fitting 10 folds for each of 45 candidates, totalling 450 fits

[Parallel(n_jobs=-1)]: Using backend LokyBackend with 4 concurrent workers.
[Parallel(n_jobs=-1)]: Done  33 tasks      | elapsed:    3.7s
[Parallel(n_jobs=-1)]: Done 154 tasks      | elapsed:   25.3s
[Parallel(n_jobs=-1)]: Done 357 tasks      | elapsed:   57.4s
[Parallel(n_jobs=-1)]: Done 450 out of 450 | elapsed:  1.1min finished
```

LightGBM Tuning

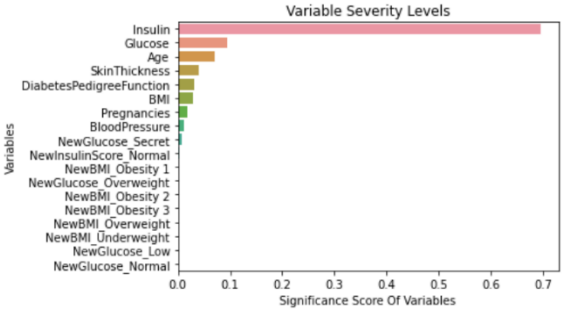


Final Model Installation

```
Fitting 10 folds for each of 720 candidates, totalling 7200 fits

[Parallel(n_jobs=-1)]: Using backend LokyBackend with 4 concurrent workers.
[Parallel(n_jobs=-1)]: Done  33 tasks      | elapsed:    4.0s
[Parallel(n_jobs=-1)]: Done 154 tasks      | elapsed:   42.3s
[Parallel(n_jobs=-1)]: Done 357 tasks      | elapsed:  1.9min
[Parallel(n_jobs=-1)]: Done 640 tasks      | elapsed:  3.1min
[Parallel(n_jobs=-1)]: Done 1005 tasks      | elapsed:  5.4min
[Parallel(n_jobs=-1)]: Done 1450 tasks      | elapsed:  8.4min
[Parallel(n_jobs=-1)]: Done 1977 tasks      | elapsed: 11.6min
[Parallel(n_jobs=-1)]: Done 2584 tasks      | elapsed: 14.8min
[Parallel(n_jobs=-1)]: Done 3273 tasks      | elapsed: 19.3min
[Parallel(n_jobs=-1)]: Done 4042 tasks      | elapsed: 23.6min
[Parallel(n_jobs=-1)]: Done 4893 tasks      | elapsed: 28.7min
[Parallel(n_jobs=-1)]: Done 5824 tasks      | elapsed: 34.5min
[Parallel(n_jobs=-1)]: Done 6837 tasks      | elapsed: 40.9min
[Parallel(n_jobs=-1)]: Done 7200 out of 7200 | elapsed: 43.2min finished
```

XGBoost Tuning

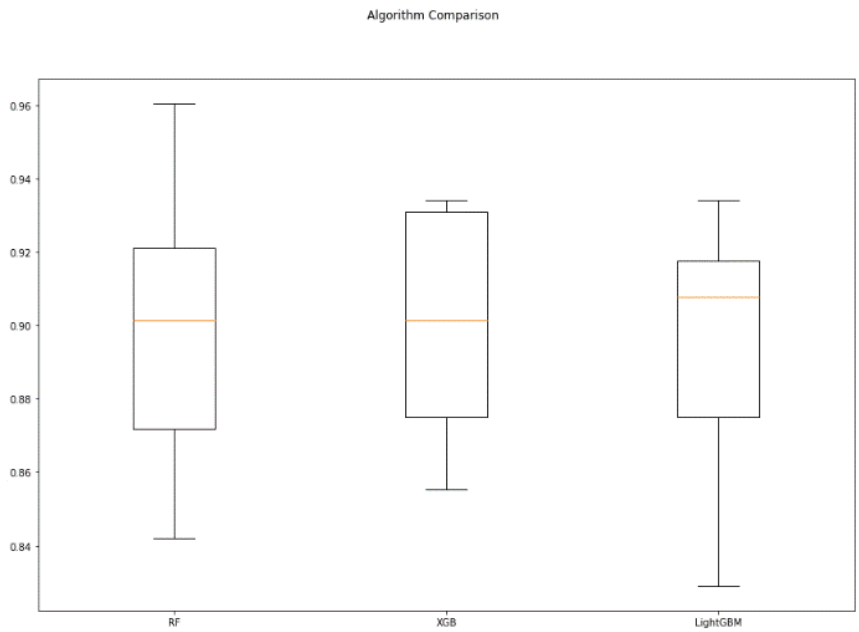


Final Model Installation

IV. COMPARISON OF FINAL MODEL

Boxplot Algorithm Comparison

```
RF: 0.897368 (0.034211)
XGB: 0.901316 (0.028373)
LightGBM: 0.896053 (0.033000)
```



V. ACCURACY COMPARISON

Algorithms	Accuracy
Random forest	0.897
XGB	0.901
Light GBM	0.896

VI. CONCLUSION

In conclusion, comparative analysis of three classification algorithms: Random Forest (RF), Light Gradient Boosting Machine (Light GBM), and Extreme Gradient Boosting (XGB) reveals subtle differences in prediction accuracy. Our main goal was not only to provide an accurate predictive model, but also to highlight the transformative potential of machine learning in predicting diabetes risk. Following a thorough analysis, the Random Forest model yielded accuracy values of 89%, Light GBM yielded accuracy values of 89.6%, and the XG Booster ML model emerged as the top performer with an exceptional accuracy of 90.13%.This remarkable result shows that XG Booster is a better tool for predicting an individual's susceptibility to diabetes compared to comparable products and highlights its effectiveness in healthcare.

VII. REFERENCE

[1] http://diabetesindia.com/

[2] https://www.sciencedirect.com/science/article/pii/S1877050920300557

[3] Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., Ye, Q., & Liu, T.-Y. (2017). LightGBM: A Highly Efficient Gradient Boosting Decision Tree. In Advances in Neural Information Processing Systems 30 (NIPS 2017) (pp. 3149–3157).

[4] Chen, T., & Guestrin, C. (2016). XGBoost: A Scalable Tree Boosting System. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (pp. 785–794).

[5] Chen, T., & Guestrin, C. (2016). XGBoost: A Scalable Tree Boosting System. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (pp. 785–794).

[6] P. Suresh Kumar and S. Pranavi “Performance Analysis of Machine Learning Algorithms on Diabetes Dataset using Big Data Analytics”,
International Conference on Infocom Technologies and Unmanned Systems, 978-1-5386-0514-1, Dec. 18-20, 2017.
[7] Mani Butwall and Shraddha Kumar,” A Data Mining Approach for the Diagnosis of Diabetes Mellitus using Random Forest Classifier”,International Journal of Computer Applications, Volume 120 - Number 8,2015.
[8] K. Rajesh and V. Sangeetha, “Application of Data Mining Methods and Techniques for Diabetes Diagnosis”, International Journal of Engineering and Innovative Technology (IJEIT) Volume 2, Issue 3, September 2012.
[9]Humar Kahramanli and Novruz Allahverdi,”Design of a Hybrid System for the Diabetes and Heart Disease”, Expert Systems with Applications: An International Journal, Volume 35 Issue 1-2, July, 2008.
[10] B.M. Patil, R.C. Joshi and Durga Toshniwal,”Association Rule for Classification of Type-2 Diabetic Patients”, ICMLC '10 Proceedings of the 2010 Second International Conference on Machine Learning and Computing, February 09 - 11, 2010