DIABETES PREDICTION REPORT

VENKATA SAI VIJAYA TEJASWINI AKELLA

NATIONAL INSTITUTE OF TECHNOLOGY SILCHAR

10 – 05 – 2021 TO 31 – 05 – 2021

CONTENTS:

1. INTRODUCTION TO DIABETES:

Diabetes is a disease that occurs when your blood glucose, also called blood sugar, is too high. Blood glucose is your main source of energy and comes from the food you eat. Insulin, a hormone made by the pancreas, helps glucose from food get into your cells to be used for energy. Sometimes your body doesn't make enough—or any—insulin or doesn't use insulin well. Glucose then stays in your blood and doesn't reach your cells.

Over time, having too much glucose in your blood can cause health problems. Although diabetes has no cure, you can take steps to manage your diabetes and stay healthy.

Sometimes people call diabetes "a touch of sugar" or "borderline diabetes." These terms suggest that someone doesn't really have diabetes or has a less serious case, but every case of diabetes is serious.

• Types of diabetes

The most common types of diabetes are type 1, type 2, and gestational diabetes.

Type 1 diabetes

If you have type 1 diabetes, your body does not make insulin. Your immune system attacks and destroys the cells in your pancreas that make insulin. Type 1 diabetes is usually diagnosed in children and young adults, although it can appear at any age. People with type 1 diabetes need to take insulin every day to stay alive. Type 1 diabetes (T1D), previously known as juvenile diabetes, is an autoimmune disease that is a form of diabetes in which very little or no insulin is produced by the islets of Langerhans (containing beta cells) in the pancreas. Insulin is a hormone required for the cells to use blood sugar for energy and it helps regulate normal glucose levels in the bloodstream. Before treatment this results in high blood sugar levels in the body. The common symptoms are frequent urination, increased thirst, increased hunger, and weight loss. Additional symptoms may include blurry vision, tiredness, and slow wound healing. Symptoms typically develop over a short period of time, often a matter of weeks.

The cause of type 1 diabetes is unknown, but it is believed to involve a combination of genetic and environmental factors. Risk factors include having a family member with the condition. The underlying mechanism involves an autoimmune destruction of the insulin-producing beta cells in the pancreas. Diabetes is diagnosed by testing the level of sugar or glycated haemoglobin (HbA1C) in the blood. Type 1 diabetes can be distinguished from type 2 by testing for the presence of autoantibodies.

There is no known way to prevent type 1 diabetes. Treatment with insulin is required for survival. Insulin therapy is usually given by injection just under the skin but can also be delivered by an insulin pump. A diabetic diet and exercise are important parts of management. If left untreated, diabetes can cause many complications. Complications of relatively rapid onset include diabetic ketoacidosis and nonketotic hyperosmolar coma. Long-term complications include heart

disease, stroke, kidney failure, foot ulcers and damage to the eyes. Furthermore, since insulin lowers blood sugar levels, complications may arise from low blood sugar if excessive amount of insulin is taken than necessary.

Type 2 diabetes

If you have type 2 diabetes, your body does not make or use insulin well. You can develop type 2 diabetes at any age, even during childhood. However, this type of diabetes occurs most often in middle-aged and older people. It is the most common type of diabetes. It is characterized by insulin resistance, which may be combined with relatively reduced insulin secretion. The defective responsiveness of body tissues to insulin is believed to involve the insulin receptor. However, the specific defects are not known. Diabetes mellitus cases due to a known defect are classified separately. Type 2 diabetes is the most common type of diabetes mellitus. Many people with type 2 diabetes have evidence of prediabetes (impaired fasting glucose and/or impaired glucose tolerance) before meeting the criteria for type 2 diabetes. The progression of prediabetes to overt type 2 diabetes can be slowed or reversed by lifestyle changes or medications that improve insulin sensitivity or reduce the liver's glucose production.

Type 2 diabetes is primarily due to lifestyle factors and genetics. A number of lifestyle factors are known to be important to the development of type 2 diabetes, including obesity (defined by a body mass index of greater than 30), lack of physical activity, poor diet, stress, and urbanization. Excess body fat is associated with 30% of cases in people of Chinese and Japanese descent, 60–80% of cases in those of European and African descent, and 100% of Pima Indians and Pacific Islanders. Even those who are not obese may have a high waist–hip ratio.

Dietary factors such as sugar-sweetened drinks are associated with an increased risk. The type of fats in the diet is also important, with saturated fat and trans fats increasing the risk and polyunsaturated and monounsaturated fat decreasing the risk. Eating white rice excessively may increase the risk of diabetes, especially in Chinese and Japanese people. Lack of physical activity may increase the risk of diabetes in some people.

Adverse childhood experiences (ACEs), including abuse, neglect, and household difficulties, increase the likelihood of type 2 diabetes later in life by 32%, with neglect having the strongest effect

Gestational diabetes

Gestational diabetes develops in some women when they are pregnant. Most of the time, this type of diabetes goes away after the baby is born. However, if you've had gestational diabetes, you have a greater chance of developing type 2 diabetes later in life. Sometimes diabetes diagnosed during pregnancy is actually type 2 diabetes. Gestational diabetes resembles type 2 diabetes in several respects, involving a combination of relatively inadequate insulin secretion and responsiveness. It occurs in about 2–10% of all pregnancies and may improve or disappear after delivery. It is recommended that all pregnant women get tested starting around 24–28 weeks gestation. It is most often diagnosed in the second or third trimester because of the increase in insulin-antagonist

hormone levels that occurs at this time. However, after pregnancy approximately 5–10% of women with gestational diabetes are found to have another form of diabetes, most commonly type 2. Gestational diabetes is fully treatable, but requires careful medical supervision throughout the pregnancy. Management may include dietary changes, blood glucose monitoring, and in some cases, insulin may be required.

Though it may be transient, untreated gestational diabetes can damage the health of the foetus or mother. Risks to the baby include macrosomia (high birth weight), congenital heart and central nervous system abnormalities, and skeletal muscle malformations. Increased levels of insulin in a foetus's blood may inhibit foetal surfactant production and cause infant respiratory distress syndrome. A high blood bilirubin level may result from red blood cell destruction. In severe cases, perinatal death may occur, most commonly as a result of poor placental perfusion due to vascular impairment. Labour induction may be indicated with decreased placental function. A caesarean section may be performed if there is marked foetal distress or an increased risk of injury associated with macrosomia, such as shoulder dystocia.

- ## Other types of diabetes

Less common types include monogenic diabetes, which is an inherited form of diabetes, and cystic fibrosis-related diabetes. Maturity onset diabetes of the young (MODY) is a rare autosomal dominant inherited form of diabetes, due to one of several single-gene mutations causing defects in insulin production. It is significantly less common than the three main types, constituting 1–2% of all cases. The name of this disease refers to early hypotheses as to its nature. Being due to a defective gene, this disease varies in age at presentation and in severity according to the specific gene defect; thus there are at least 13 subtypes of MODY. People with MODY often can control it without using insulin.

Some cases of diabetes are caused by the body's tissue receptors not responding to insulin (even when insulin levels are normal, which is what separates it from type 2 diabetes); this form is very uncommon. Genetic mutations (autosomal or mitochondrial) can lead to defects in beta cell function. Abnormal insulin action may also have been genetically determined in some cases. Any disease that causes extensive damage to the pancreas may lead to diabetes (for example, chronic pancreatitis and cystic fibrosis). Diseases associated with excessive secretion of insulin-antagonistic hormones can cause diabetes (which is typically resolved once the hormone excess is removed). Many drugs impair insulin secretion and some toxins damage pancreatic beta cells, whereas others increase insulin resistance (especially glucocorticoids which can provoke "steroid diabetes"). The ICD-10 (1992) diagnostic entity, *malnutrition-related diabetes mellitus* (MRDM or MMDM, ICD-10 code E12), was deprecated by the World Health Organization (WHO) when the current taxonomy was introduced in 1999. Yet another form of diabetes that people may develop is double diabetes. This is when a type 1 diabetic becomes insulin resistant, the hallmark for type 2 diabetes or has a family history for type 2 diabetes.

- ## How common is diabetes?

As of 2015, 30.3 million people in the United States, or 9.4 percent of the population, had diabetes. More than 1 in 4 of them didn't know they had the disease. Diabetes affects 1 in 4 people over the age of 65. About 90-95 percent of cases in adults are type 2 diabetes.

You are more likely to develop type 2 diabetes if you are age 45 or older, have a family history of diabetes, or are overweight. Physical inactivity, race, and certain health problems such as high blood pressure also affect your chance of developing type 2 diabetes. You are also more likely to develop type 2 diabetes if you have prediabetes or had gestational diabetes when you were pregnant. Learn more about risk factors for type 2 diabetes.

- ## What health problems can people with diabetes develop?

Over time, high blood glucose leads to problems such as

- Heart Disease
- Stroke
- Kidney Disease
- Eye Problems
- Dental Disease
- Nerve Damage
- Foot Problems

2. OVERVIEW OF DATA SET:

This dataset was taken from Kaggle. This dataset is originally from the National Institute of Diabetes and Digestive and Kidney Diseases. The objective of the dataset is to diagnostically predict whether or not a patient has diabetes, based on certain diagnostic measurements included in the dataset. Several constraints were placed on the selection of these instances from a larger database. In particular, all patients here are females at least 21 years old of Pima Indian heritage. The datasets consists of several medical predictor variables and one target variable, Outcome. Predictor variables includes the number of pregnancies the patient has had, their BMI, insulin level, age, and so on.

- Pregnancies: Number of times pregnant
- Glucose: Plasma glucose concentration a 2 hours in an oral glucose tolerance test
- Blood Pressure: Diastolic blood pressure (mm Hg)
- Skin Thickness: Triceps skin fold thickness (mm)
- Insulin: 2-Hour serum insulin (mu U/ml)
- BMI: Body mass index (weight in kg / (height in m)^2)
- Diabetes Pedigree Function: Diabetes pedigree function
- Age: Age (years)
- Outcome: Class variable (0 or 1) 268 of 768 are 1, the others are 0

3. ANALYSIS OF DATA SET:

With the shape attribute applied on the data set, we come to know that the dataset has 768 rows and 9 columns out of which the 'Outcome' column is the categorical variable which we need to predict. With the info method applied on the data frame we observe that the pregnancies column has 768 values out of which all the values are non null values and the data type is int64. Coming to the next column that is the glucose column it has 768 values, all of them being non null values with data type int64. The next column is blood pressure with a length of 768, all non null values and the data type is int64. Skin thickness is the next column with same length of 768 non null values of data type int64. Insulin is the next column which indicates the insulin levels of a person with the length of 768 non null values of data type int64. The Body Mass Index (BMI) is the next column with 768 non null float64 values. Next comes the diabetes Pedigree function with 768 non null float 64 values. The last feature column is the age which indicate the age of a person in years obviously of the data type int64 and 768 non null values. Finally the last column of the data set is the categorical column which is the label to be predicted by the implementation of models that is the outcome column with 768 non null in 64 values which predicts whether the person has diabetes or not. To summarize, we have 7 int64 values along with 2 float64 values which have an overall memory usage of 54.1 KB. Firstly we import the necessary libraries such as the Pandas for the manipulation of data frame, followed by the numpy which stands for numerical Python used to perform several numerical calculations on the data set, consequently we import the matplotlib library which is used to visualise the data with the help of several plots such as scatter plot, pie plot, bar graph etc. Lastly we import seaborn for advanced data visualisation. We import all of the libraries with short names like PD NP PLT SNS respectively for convenience. After importing all the necessary libraries we load the data set which is the diabetes.csv from kaggle as a comma separated file using the read_csv() function of the Pandas. With the help of head() and tail() methods we can have a glimpse of the first 10 as well as last 10 values of the data set. With the isnull().sum() method applied on the data we come to know the number of null values in the data set. After applying the method we find that all the columns in the data set have not even a single

null value. With the value_counts() function applied on the data set which is imported as df we come to know the number of values existing in each column and their counts respectively. We observe from the analysis done till now that the columns of glucose blood pressure skin thickness insulin and BMI have some of their values equal to zero which is not at all practically possible, so in order to become convenient with the calculation we convert these values into np.nan which is the numerical representation of a null value or a missing value in Python. After replacing all the 0 values with np.nan (not a number), we again implement the isnull().sum() function on the data frame by which we can see that the number of null values has drastically increased. The glucose column has 5 null values the blood pressure column has 35 null values the skin thickness column has 227 null values in column has 374 null values BMI column has 11 null values respectively. So we treat these zero values as outliers and since the data set is not of large size the best method to treat this outliers is to impute them. In this method of imputation, we replace the null values either with the mean of the column or the mode or median or any respective statistical parameter of that respective column. Here we implement the practice of replacing the null values with the mean of the respective column. We first visually plot the number of null values existing in all the columns using the bar plot in the seaborn library horizontally annotated with the labels. We use plt.show() function in order to display the plot. Subsequently we implement the imputation method by using the replace function. We run a loop to all the the columns of glucose blood pressure skin thickness insulin BMI where the value 0 is not practically feasible. We round of the mean to three decimal places and replace the missing values in the column with the main rounded off to three positions in place. After the imputation is implemented we again apply the head method and visualise the first 10 columns of the data frame and make sure that the null values are successfully replaced. With the describe() method applied on the data set we can get the whole statistical summary of the data set columns. We get the gist of statistical summary like the count which represents the length of each column the mean standard deviation minimum value 25th percentile 50th percentile 75th percentile and the maximum values of each column in the data set respectively. After successful observation on the data we now come to the trend analysing part of the data. We find the respective correlations of each and every column existing in the data. We plot the correlation using the heatmap function from the seaborn library with annotations and the respective formatting of colour rounding off values etc. Correlation of two columns indicates the relationship between the two columns in other words it indicates how strongly two columns are related to each other and are affected by each other. From the heat map we can observe that there are no strong relations between any of the columns (>0.5) The greatest correlation is between the pregnancy and the age column followed by the correlation between glucose and output columns respectively followed by glucose and insulin columns. Positive correlation indicates that two columns are proportional to each other directly where as a negative correlation indicates that the two columns are inversely proportional to each other. In order to better visualise the correlation between the existing columns in the data set we use the pair plot function in the seaborn library from this we obtain a scatter plot of the correlations between each and every column in the data set. We now visualise the individual relation of each column with the output column. With the crosstab() function of the Pandas library we plot a bar plot of the number

of people suffering from diabetes and not suffering from diabetes with respective medical characteristics. We also observe the characteristics of the people by plotting individual histograms of the medical characteristics of the people having diabetes (or output as 1) After the analysing part we now come to the implementation. We divide the data set into X and y. X stands for the features column where as y stands for the output column we now split the data into training data set and testing data set with the test data set of size 20%. For this we import the train_test_split module from the sklearn.model_selection and observe the lengths of each X_train, X_test, y_train, y_test respectively. Another important feature to be noted while implementing any model is that the difference in the values of the columns can make the model to be biased that implies if a model has relatively higher values than others, then the model might get confused that the particular column has higher significance than others. In order to avoid this chaos, we use the StandardScaler() module of sklearn.preprocessing using which we can scale all the values into a particular range of equal importance so that the model does not get confused. We fit the scaler to the training data set and test data set and once again visualise the statistical trend of each column.

4. BULDING THE MODELS:

We start the implementation of each model.

a. K NEAREST NEIGHBORS
b. LOGISTIC REGRESSION
c. RANDOM FOREST CLASSIFIER
d. DECISION TREE CLASSIFIER
e. SUPPORT VECTOR MACHINE
f. GAUSSIAN NAÏVE BYERS
g. XG BOOST

- **K NEAREST NEIGHBORS:**

- K-Nearest Neighbour 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 similar to the available categories.
- K-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 K-NN 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 similar to the new data.
- The K-NN working can be explained on the basis of the below algorithm:
- Step-1: Select the number K of the neighbors
- Step-2: Calculate the Euclidean distance of K number of neighbors
- Step-3: Take the K nearest neighbors as per the calculated Euclidean distance.
- Step-4: Among these k neighbors, count the number of the data points in each category.
- Step-5: Assign the new data points to that category for which the number of the neighbor is maximum.
- Step-6: Our model is ready.
- It is simple to implement.
- It is robust to the noisy training data
- It can be more effective if the training data is large.Always needs to determine the value of K which may be complex some time.
- The computation cost is high because of calculating the distance between the data points for all the training samples.

- **LOGISTIC REGRESSION:**

- Comes under the Supervised Learning technique. It is used for predicting the categorical dependent variable using a given set of independent variables.

- Logistic regression predicts the output of a categorical dependent variable. Therefore the outcome must be a categorical or discrete value. It can be either Yes or No, 0 or 1, true or False, etc. but instead of giving the exact value as 0 and 1, it gives the probabilistic values which lie between 0 and 1.
- Logistic Regression is much similar to the Linear Regression except that how they are used. Linear Regression is used for solving Regression problems, whereas Logistic regression is used for solving the classification problems.
- In Logistic regression, instead of fitting a regression line, we fit an "S" shaped logistic function, which predicts two maximum values (0 or 1).
- The curve from the logistic function indicates the likelihood of something such as whether the cells are cancerous or not, a mouse is obese or not based on its weight, etc.
- Logistic Regression is a significant machine learning algorithm because it has the ability to provide probabilities and classify new data using continuous and discrete datasets.
- Logistic Regression can be used to classify the observations using different types of data and can easily determine the most effective variables used for the classification.
- Logistic regression uses the concept of predictive modeling as regression; therefore, it is called logistic regression, but is used to classify samples; Therefore, it falls under the classification algorithm.
- The dependent variable must be categorical in nature.
- The independent variable should not have multi-collinearity.
- On the basis of the categories, Logistic Regression can be classified into three types:
- Binomial: In binomial Logistic regression, there can be only two possible types of the dependent variables, such as 0 or 1, Pass or Fail, etc.
- Multinomial: In multinomial Logistic regression, there can be 3 or more possible unordered types of the dependent variable, such as "cat", "dogs", or "sheep"
- Ordinal: In ordinal Logistic regression, there can be 3 or more possible ordered types of dependent variables, such as "low", "Medium", or "High".

- **RANDOM FOREST:**

- belongs to the supervised learning technique. It can be used for both Classification and Regression problems in ML. It is based on the concept of ensemble learning, which is a process of *combining multiple classifiers to solve a complex problem and to improve the performance of the model.*
- As the name suggests, *"Random Forest is a classifier that contains a number of decision trees on various subsets of the given dataset and takes the average to improve the predictive accuracy of that dataset."* Instead of relying on one decision tree, the random forest takes the prediction from each tree and based on the majority votes of predictions, and it predicts the final output.
- Since the random forest combines multiple trees to predict the class of the dataset, it is possible that some decision trees may predict the correct output, while others may not. But together, all the trees predict the correct output. Therefore, below are two assumptions for a better Random forest classifier:
- There should be some actual values in the feature variable of the dataset so that the classifier can predict accurate results rather than a guessed result.
- The predictions from each tree must have very low correlations.
- Below are some points that explain why we should use the Random Forest algorithm:
- It takes less training time as compared to other algorithms.
- It predicts output with high accuracy, even for the large dataset it runs efficiently.
- It can also maintain accuracy when a large proportion of data is missing.
- The Working process can be explained in the below steps and diagram:
- Step-1: Select random K data points from the training set.
- Step-2: Build the decision trees associated with the selected data points (Subsets).
- Step-3: Choose the number N for decision trees that you want to build.
- Step-4: Repeat Step 1 & 2.
- Step-5: For new data points, find the predictions of each decision tree, and assign the new data points to the category that wins the majority votes.
- Random Forest is capable of performing both Classification and Regression tasks.
- It is capable of handling large datasets with high dimensionality.
- It enhances the accuracy of the model and prevents the overfitting issue.

- Although random forest can be used for both classification and regression tasks, it is not more suitable for Regression tasks.

- **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 tree-structured 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 on the basis of 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, similar to a tree, it starts with the root node, which expands on further branches and constructs a tree-like structure.
- In order to build a tree, we use the CART algorithm, which stands for Classification and Regression Tree algorithm.
- A decision tree simply asks a question, and based on the answer (Yes/No), it further split the tree into subtrees.
- Decision Trees usually mimic human thinking ability while making a decision, so it is easy to understand.
- The logic behind the decision tree can be easily understood because it shows a tree-like structure.
- In a decision tree, for predicting the class of the given dataset, the algorithm starts from the root node of the tree. This algorithm compares the values of root attribute with the record (real dataset) attribute and, based on the comparison, follows the branch and jumps to the next node.

- For the next node, the algorithm again compares the attribute value with the other sub-nodes and move further. It continues the process until it reaches the leaf node of the tree. The complete process can be better understood using the below algorithm:
- Step-1: Begin the tree with the root node, says S, which contains the complete dataset.
- Step-2: Find the best attribute in the dataset using Attribute Selection Measure (ASM).
- Step-3: Divide the S into subsets that contains possible values for the best attributes.
- Step-4: Generate the decision tree node, which contains the best attribute.
- Step-5: Recursively make new decision trees using the subsets of the dataset created in step -3. Continue this process until a stage is reached where you cannot further classify the nodes and called the final node as a leaf node.
- It is simple to understand as it follows the same process which a human follow while making any decision in real-life.
- It can be very useful for solving decision-related problems.
- It helps to think about all the possible outcomes for a problem.
- There is less requirement of data cleaning compared to other algorithms. The decision tree contains lots of layers, which makes it complex.
- It may have an overfitting issue, which can be resolved using the Random Forest algorithm.
- For more class labels, the computational complexity of the decision tree may increase.

- **SUPPORT VECTOR MACHINE:**

- 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 n-dimensional 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.

- SVM chooses the extreme points/vectors that help in creating the hyperplane. These extreme cases are called as support vectors, and hence algorithm is termed as Support Vector Machine.
- SVM can be of two types:
- Linear SVM: Linear SVM is used for linearly separable data, which means if a dataset can be classified into two classes by using a single straight line, then such data is termed as linearly separable data, and classifier is used called as Linear SVM classifier.
- Non-linear SVM: Non-Linear SVM is used for non-linearly separated data, which means if a dataset cannot be classified by using a straight line, then such data is termed as non-linear data and classifier used is called as Non-linear SVM classifier.

- **GAUSSIAN NAÏVE BYERS:**

- Naïve Bayes algorithm is a supervised learning algorithm, which is based on Bayes theorem and used for solving classification problems.
- It is mainly used in *text classification* that includes a high-dimensional training dataset.
- Naïve Bayes Classifier is one of the simple and most effective Classification algorithms which helps in building the fast machine learning models that can make quick predictions.
- It is a probabilistic classifier, which means it predicts on the basis of the probability of an object.
- Some popular examples of Naïve Bayes Algorithm are spam filtration, Sentimental analysis, and classifying articles.The Naïve Bayes algorithm is comprised of two words Naïve and Bayes, Which can be described as:
- Naïve: It is called Naïve because it assumes that the occurrence of a certain feature is independent of the occurrence of other features. Such as if the fruit is identified on the bases of color, shape, and taste, then red, spherical, and sweet fruit is recognized as an apple. Hence each feature individually contributes to identify that it is an apple without depending on each other.
- Bayes: It is called Bayes because it depends on the principle of Bayes' Theorem. Bayes' theorem is also known as Bayes' Rule or Bayes' law, which is used to determine the probability of a hypothesis with prior knowledge. It depends on the conditional probability.

- Working of Naïve Bayes' Classifier can be understood with the help of the below example:
- Suppose we have a dataset of weather conditions and corresponding target variable "Play". So using this dataset we need to decide that whether we should play or not on a particular day according to the weather conditions. So to solve this problem, we need to follow the below steps:
- Convert the given dataset into frequency tables.
- Generate Likelihood table by finding the probabilities of given features.
- Now, use Bayes theorem to calculate the posterior probability.
- Naïve Bayes is one of the fast and easy ML algorithms to predict a class of datasets.
- It can be used for Binary as well as Multi-class Classifications.
- It performs well in Multi-class predictions as compared to the other Algorithms.
- It is the most popular choice for text classification problems.
- Naive Bayes assumes that all features are independent or unrelated, so it cannot learn the relationship between features.

- **XG BOOST:**

- XGBoost is a popular and efficient open-source implementation of the gradient boosted trees algorithm. Gradient boosting is a supervised learning algorithm, which attempts to accurately predict a target variable by combining the estimates of a set of simpler, weaker models.

- When using gradient boosting for regression, the weak learners are regression trees, and each regression tree maps an input data point to one of its leafs that contains a continuous score. XGBoost minimizes a regularized (L1 and L2) objective function that combines a convex loss function (based on the difference between the predicted and target outputs) and a penalty term for model complexity (in other words, the regression tree functions). The training proceeds iteratively, adding new trees that predict the residuals or errors of prior trees that are then combined with previous trees to make the final prediction. It's called gradient boosting because it uses a gradient descent algorithm to minimize the loss when adding new models.

- XGBoost is able to solve real world scale problems using a minimal amount of resources.

5. HYPER PARAMETER TUNING:

A Machine Learning model is defined as a mathematical model with a number of parameters that need to be learned from the data. By training a model with existing data, we are able to fit the model parameters.

However, there is another kind of parameters, known as *Hyperparameters*, that cannot be directly learned from the regular training process. They are usually fixed before the actual training process begins. These parameters express important properties of the model such as its complexity or how fast it should learn.

Some examples of model hyperparameters include:

1. The penalty in Logistic Regression Classifier i.e. L1 or L2 regularization
2. The learning rate for training a neural network.
3. The C and sigma hyperparameters for support vector machines.
4. The k in k-nearest neighbors.

The aim of this article is to explore various strategies to tune hyperparameter for Machine learning model.

Models can have many hyperparameters and finding the best combination of parameters can be treated as a search problem. Two best strategies for Hyperparameter tuning are:

- GridSearchCV:

The traditional way of performing hyperparameter optimization has been *grid search*, or a *parameter sweep*, which is simply an exhaustive searching through a manually specified subset of the hyperparameter space of a learning algorithm. A grid search algorithm must be guided by some performance metric, typically measured by cross-validation on the training set or evaluation on a held-out validation set.

Since the parameter space of a machine learner may include real-valued or unbounded value spaces for certain parameters, manually set bounds and discretization may be necessary before applying grid search.

For example, a typical soft-margin SVM classifier equipped with an RBF kernel has at least two hyperparameters that need to be tuned for good performance on unseen data: a regularization constant *C* and a kernel hyperparameter γ. Both parameters are continuous, so to perform grid search, one selects a finite set of "reasonable" values for each.

Grid search then trains an SVM with each pair (*C*, γ) in the Cartesian product of these two sets and evaluates their performance on a held-out validation set (or by internal cross-validation on the training set, in which case multiple SVMs are trained per pair). Finally, the grid search algorithm outputs the settings that achieved the highest score in the validation procedure.

Grid search suffers from the curse of dimensionality, but is often embarrassingly parallel because the hyperparameter settings it evaluates are typically independent of each other

- RandomizedSearchCV:

Random Search replaces the exhaustive enumeration of all combinations by selecting them randomly. This can be simply applied to the discrete setting described above, but also generalizes to continuous and mixed spaces. It can outperform Grid search, especially when only a small number of hyperparameters affects the final performance of the machine learning algorithm. In this case, the optimization problem is said to have a low intrinsic dimensionality. Random Search is also embarrassingly parallel, and additionally allows the inclusion of prior knowledge by specifying the distribution from which to sample. RandomizedSearchCV implements a “fit” and a “score” method. It also implements “score_samples”, “predict”, “predict_proba”, “decision_function”, “transform” and “inverse_transform” if they are implemented in the estimator used.

The parameters of the estimator used to apply these methods are optimized by cross-validated search over parameter settings.

In contrast to GridSearchCV, not all parameter values are tried out, but rather a fixed number of parameter settings is sampled from the specified distributions. The number of parameter settings that are tried is given by n_iter.

If all parameters are presented as a list, sampling without replacement is performed. If at least one parameter is given as a distribution, sampling with replacement is used. It is highly recommended to use continuous distributions for continuous parameters.

- ## Classification Report

The classification report visualizer displays the precision, recall, F1, and support scores for the model. In order to support easier interpretation and problem detection, the report integrates numerical scores with a color-coded heatmap. All heatmaps are in the range (0.0, 1.0) to facilitate easy comparison of classification models across different classification reports.

The classification report shows a representation of the main classification metrics on a per-class basis. This gives a deeper intuition of the classifier behavior over global accuracy which can mask functional weaknesses in one class of a multiclass problem. Visual classification reports are used to compare classification models to select models that are “redder”, e.g. have stronger classification metrics or that are more balanced.

The metrics are defined in terms of true and false positives, and true and false negatives. Positive and negative in this case are generic names for the classes of a binary classification problem. In the example above, we would consider true and false occupied and true and false unoccupied. Therefore a true positive is when the actual class is positive as is the estimated class. A false positive is when the actual class is negative but the estimated class is positive. Using this terminology the metrics are defined as follows:

precision

Precision can be seen as a measure of a classifier’s exactness. For each class, it is defined as the ratio of true positives to the sum of true and false positives. Said another way, “for all instances classified positive, what percent was correct?”

recall

Recall is a measure of the classifier's completeness; the ability of a classifier to correctly find all positive instances. For each class, it is defined as the ratio of true positives to the sum of true positives and false negatives. Said another way, "for all instances that were actually positive, what percent was classified correctly?"

f1 score

The F_1 score is a weighted harmonic mean of precision and recall such that the best score is 1.0 and the worst is 0.0. Generally speaking, F_1 scores are lower than accuracy measures as they embed precision and recall into their computation. As a rule of thumb, the weighted average of F_1 should be used to compare classifier models, not global accuracy.

support

Support is the number of actual occurrences of the class in the specified dataset. Imbalanced support in the training data may indicate structural weaknesses in the reported scores of the classifier and could indicate the need for stratified sampling or rebalancing. Support doesn't change between models but instead diagnoses the evaluation process.

S. NO.	MODEL NAME	ACCURACY (%)
1.	SUPPORT VECTOR MACHINE	79.2208
2.	RANDOM FOREST	77.2727
3.	LOGISTIC REGRESSION	76.5507
4.	K NEAREST NEIGHBORS	75.5737
5.	GAUSSIAN NAÏVE BAYERS	74.6753
6.	DECISION TREE	77.5304
7.	XG BOOST	78.1833

6. CONCLUSION:

After performing the hyper parameter tuning on several models, and testing for the accuracy of each model, we conclude that the SUPPORT VECTOR MACHINE algorithm showed the best performance with an accuracy of 79.22 %.

The original labels are as follows:

```
[0 0 0 0 0 0 0 0 0 0 1 0 1 0 0 1 0 0 1 1 0 0 0 0 0 1 0 0 1 0 1 1 1 1 0 1 1
 1 0 1 0 0 0 1 0 1 1 0 0 0 0 1 1 1 0 0 0 0 0 1 1 0 0 1 0 0 0 1 0 1 0 0 0 1
 0 0 0 0 0 0 1 1 0 0 0 1 1 1 0 0 0 0 0 1 0 0 0 0 1 0 0 0 0 1 0 0 0 1 0 1 0 0 1 1 1 0
 0 0 1 0 0 1 0 0 0 0 0 1 0 0 0 1 1 1 0 1 1 0 1 1 0 1 1 1 0 0 0 0 0 0 0 0 1 0
 0 1 0 0 1 0]
```

The predicted labels by the model are as follows:

```
[0 0 0 0 0 0 0 0 0 0 1 0 1 0 0 0 0 0 0 1 0 0 0 0 0 0 1 0 0 0 0 1 1 1 1 1 1 1
 0 0 0 0 0 0 0 0 1 1 0 0 1 0 1 1 0 0 0 1 0 0 1 1 0 0 0 0 1 0 1 0 1 1 0 0 0
 0 0 0 0 0 0 1 0 0 0 0 1 1 0 0 0 0 0 0 0 0 1 0 0 1 0 1 0 1 0 0 1 0 0 0
 0 0 1 0 0 1 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 0 0 1 0 0 1 1 0 0 0 0 0 0 0 0 0
 0 1 0 0 0 0]
```