STROKE PREDICTION

1. **ABSTRACT:**

According to the World Health Organization (WHO) stroke is the 2nd leading cause of death globally, responsible for approximately 11% of total deaths. So, in this project we are predicting whether the patient is likely to get stroke or not depending on the input parameters. We adopt numerous machine learning concepts and algorithms in this project to predict the chances of stroke for a given patient. Machine Learning has solid connections to numerical improvement, which conveys strategies, hypothesis and application areas to the field. Machine learning is some of the time conflated with data mining where the latter subfield concentrates more on exploratory information analysis and is known as supervised learning. Use of machine learning and data science makes life easier in every aspect, using machine learning and predicting the outcomes whether the patient will get stroke or not, will save many lives of the humans. And machine learning is booming, and machine learning is firmly identified with (and frequently covers with) computational insights, which also focuses on prediction making through the use of technology.

The main idea of this project is: -

-In this project our aim is to predict whether the patient will get stroke or not.

-We are implementing different models like Naïve Bayes, Logistic regression and K- nearest neighbour for predicting the stroke.

-We are calculating precision, recall, F1 score and accuracy to validate the model.

- We are also calculating average precision recall score and precision recall curve in this project, which would be used as the evaluating metrics.

1. **BACKGROUND:**

**Data sources and data preparation:**

* We are using the Healthcare-Dataset-Stroke-Date available on the Kaggle.
* There is a total of 12 attributes in a data with 5110 data instances, so the dataset is enough for our analysis and generating predictions.
* The dataset used were in csv format, so we didn’t have perform any additional data preparation tasks in order to use the data.

**Data exploration, visualization, cleansing and transformation**

The dataset is available on Kaggle, which is been used in this project is already preprocessed and only one column “bmi” has 201 missing values which we filled it with 0. The dataset includes variables such as gender, age, hypertension, heart disease, work type, bmi, smoking status and many more other parameters, which contributes towards the prediction.

After some initial exploration of the datasets, we came up with the following dataset with which we conducted our analysis on to achieve our goals.

Table

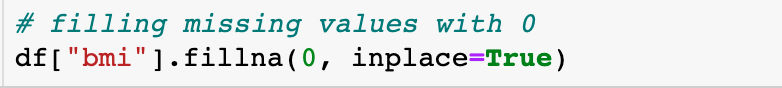
Description automatically generated

Checking the Null values: -

Table

Description automatically generated

Filling the missing values in bmi column with 0: -



Displaying the basic details about the dataset: -

Table

Description automatically generated

For the modeling we have to create our dataset according to our problem, our main aim to use supervised learning modelling is to predict weather the patient will get a stroke or not, for that we have to pre- process the data accordingly.

1. **RELATED WORK AND METHODOLOGY: -**

The main task of this project after loading the data is to eliminate the variables which don’t have impact towards the prediction and deleting the columns which are not contributing towards the prediction and create the correlation matrix between the attributes.

Table

Description automatically generated

We used various supervised learning algorithms to model our data. The models we used in our project are Naïve Bayes, Logistic regression and K- nearest neighbour. Then we will be validating each model to check whether the model is best for our prediction or not by using accuracy, precision, recall and F1 score and also by calculating confusion matrix.

**MODELS: -**

Logistic regression: Logistic regression is a classification model. The model builds a classification model to predict the probability that a given data entry belongs to the category numbered as “1” or “0”. Just like Linear regression assumes that the data follows a linear function, Logistic regression models the data using the sigmoid function.

K- nearest neighbour: K-Nearest Neighbors (KNN) is one of the simplest algorithms used in [Machine Learning for regression](https://quantra.quantinsti.com/course/trading-with-machine-learning-regression) and classification problem. KNN algorithms use data and classify new data points based on similarity measures (e.g. distance function). Classification is done by a majority vote to its neighbors. The data is assigned to the class which has the nearest neighbors. As you increase the number of nearest neighbors, the value of k, accuracy might increase.

Naïve Bayes: The naive Bayes classifiers are a simple family of "probabilistic classifiers" based on Bayes' theorem and strong (naive) independence assumptions between the features. They are one of the most basic Bayesian network models, but when combined with kernel density estimation, they can achieve higher levels of accuracy. The number of parameters required by Nave Bayes classifiers is linear in the number of variables (features/predictors) in a learning problem.

1. **EXPERIMENTS AND RESULTS: -**

We are predicting the patient will get the stroke or not by considering many variables and then distributing our data into training and testing.

As we are doing modelling, we have several techniques for modelling, such as accuracy, precision, recall and F1 score. We calculate the accuracy, precision, recall and F1 score values using the statistical formulas. The highest the accuracy is, better the model is.

**Accuracy:** Accuracy is the most intuitive performance measure and it is simply a ratio of correctly predicted observation to the total observations.

Accuracy = TP + TN / TP+FP+FN+TN

**Precision:** Precision is the ratio of correctly predicted positive observations to the total predicted positive observations.

Precision = TP/TP+FP

**Recall:** Recall is the ratio of correctly predicted positive observations to the all observations in actual class

Recall = TP/TP+FN

**F1 Score:** F1 Score is the weighted average of Precision and Recall. Therefore, this score takes both false positives and false negatives into account.

F1 Score = 2 \*(Precision \* Recall) / (Recall+ Precision)

**Average- Precision Score: -**

AP summarizes a precision-recall curve as the weighted mean of precisions achieved at each threshold, with the increase in recall from the previous threshold used as the weight:

Text

Description automatically generated

where Pn and Rn are the precision and recall at the nth threshold. This implementation is not interpolated and is different from computing the area under the precision-recall curve with the trapezoidal rule, which uses linear interpolation and can be too optimistic.

**Precision- Recall Curve: -**

The precision-recall curve is made by calculating and plotting the precision versus recall for a single classifier at various thresholds. It is the graph between Precision and Recall where recall is on x-axis and precision is on y-axis. A precision-recall curve helps to visualize how the choice of threshold affects classifier performance and can even help us select the best threshold for a specific problem.

**First, we implemented logistic regression: -**

After implementing the logistic regression, we calculated accuracy, precision, recall and f1 score and got the following values:

Accuracy of Logistic Regression: 0.9422700587084148

Precision score of Logistic Regression: 0.3076923076923077

recall score of Logistic Regression: 0.07407407407407407

F1 score of Logistic Regression: 0.11940298507462686

Graphical user interface, text, application, email

Description automatically generated

And after calculating this we created confusion matrix and calculated mean average precision and made precision recall curve.

Text

Description automatically generated

The values of confusion matrix are: -

accuracy = 0.9422700587084148

precision = 0.3076923076923077

recall = 0.07407407407407407

F1 = 0.11940298507462686

TRUE POSITIVE = 4

TRUE NEGATIVE = 959

FALSE NEGATIVE = 50

FALSE POSITIVE = 9

Mean Average precision and Precision recall score are: -

Graphical user interface

Description automatically generated

**Second, we implemented K-nearest neighbour: -**

We followed the same steps as Logistics regression, after implementing the KNN we calculated accuracy, precision, recall and f1 score and got the following values:

Cross Validation score: 0.9530362826375292

Accuracy of K-nearest Neighbor: 0.9481409001956947

Precision score of K-nearest Neighbor: 0.6666666666666666

recall score of K-nearest Neighbor: 0.037037037037037035

F1 score of K-nearest Neighbor: 0.07017543859649122

Graphical user interface, text, application, email

Description automatically generated

And after calculating this we created confusion matrix and calculated mean average precision and made precision recall curve.

Table

Description automatically generated with medium confidence

The values of confusion matrix are: -

accuracy = 0.9481409001956947

precision = 0.6666666666666666

recall = 0.037037037037037035

F1 = 0.07017543859649122

TRUE POSITIVE = 2

TRUE NEGATIVE = 967

FALSE NEGATIVE = 52

FALSE POSITIVE = 1

Mean Average precision and Precision recall score are: -

Chart

Description automatically generated

**Third, we implemented Naïve Bayes: -**

We followed the same steps as Logistics regression, after implementing the Naïve Bayes we calculated accuracy, precision, recall and f1 score and got the following values:

Cross Validation score: 0.9425184817408878

Accuracy of Naive Bayes: 0.9393346379647749

Precision score of Naive Bayes: 0.1

recall score of Naive Bayes: 0.018518518518518517

F1 score of Naive Bayes: 0.03125

Text

Description automatically generated

And after calculating this we created confusion matrix and calculated mean average precision and made precision recall curve.

Table

Description automatically generated with medium confidence

The values of confusion matrix are: -

accuracy = 0.9393346379647749

precision = 0.1

recall = 0.018518518518518517

F1 = 0.03125

TRUE POSITIVE = 1

TRUE NEGATIVE = 959

FALSE NEGATIVE = 53

FALSE POSITIVE = 9

Mean Average precision and Precision recall score are: -

Graphical user interface, application

Description automatically generated

1. **CONCLUSION: -**

By implementing 3 different types of models, which are Logistic regression, K-nearest neighbour and Naïve Bayes to figure out the best model to predict whether the patient will get the stroke or not and after evaluating we concluded that for our dataset K-nearest neighbour is working good as compared to other models as it got the highest accuracy which is equal to 94.8% and even the precision value for K- nearest neighbour is highest which is equal to 0.66.

The models are evaluated on the above data set and the evaluation metrics for all of the models are been compared below: -

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Models | Accuracy | Precision | Recall | F1 |
| Logistic Regression | 0.942 | 0.30 | 0.07 | 0.1 |
| K-nearest Neighbour | 0.948 | 0.66 | 0.03 | 0.07 |
| Naïve Bayes | 0.93 | 0.1 | 0.01 | 0.03 |

**References: -**

* 1. <https://scikit-learn.org/stable/modules/generated/sklearn.metrics.average_precision_score.html>
  2. <https://www.kaggle.com/fedesoriano/stroke-prediction-dataset>
  3. <https://en.wikipedia.org/wiki/Naive_Bayes_classifier>
  4. <https://blog.quantinsti.com/machine-learning-k-nearest-neighbors-knn-algorithm-python/>
  5. <https://medium.com/@douglaspsteen/precision-recall-curves-d32e5b290248>