Stroke Prediction Using Naïve Bayes (NB) & Random Forest (RF)

Ajai Suresh (170052073)

Description and Motivation

In today's world stroke is one of the most known reason for death; according to WHO stroke is the 2nd leading cause for death. The aim of this project is to use two different ML (machine learning) models to predict if a person has stroke or not. The two machine learning models used for this project are ***Naïve Bayes*** and ***Random Forest***.

Exploratory Data Analysis

- This dataset has been taken from Kaggle; Stroke Prediction Dataset (26.jan 2021 updated)
- Initial dataset
  - **5110** rows
  - **12** columns
    - **11** predictors
    - **1** target variable
      - The target variable contains 1 if stroke is predicted and 0 if not.
  - **201** missing values (null values) on the BMI column
  - There were outliers on BMI and glucose level
  - Work_type, smoking_status, gender, Residence_type, ever_married → these were strings
- Cleaned dataset
  - ID was not necessary as it was just a random number, hence removed.
  - The null values for the BMI were replaced with the mean of the BMI.
  - The outliers were not removed as they both have an impact on stroke prediction.
    - The outliers can be seen on figure 1.
  - Using label encoding, to convert the labels into numeric forms.
- A class imbalance problem does exist as only 249 cases is tested positive for stroke from 5110 cases.

Naïve Bayes

- The Naive Bayes algorithm is a tool for classifying data.
- Classifiers are models that categorize issue occurrences and assign class labels to them using vectors for predictors or function values.
- Its foundation is Bayes' theorem.
- It's termed naive Bayes because it assumes that the value of one function is independent of the value of the other, i.e. that altering one function's value has no effect on the other's value. For the same reason, it's also known as stupid Bayes.
- This technique is best suited for real-time predictions since it performs well with huge datasets.
- It helps to calculate the posterior probability P (c | x) using the previous probability for class P (c), the previous probability for predictor P (x) and the probability for predictor given class, also called probability P (x) | c).

Advantages

- Easy to implement.
- Fast If the assumption of independence holds, it works more efficiently than other algorithms.
- It requires less training data.
- It can work easily with missing values.
- Handles both continuous and discrete data.

Disadvantages

- The strong presumption that the functions should be independent, which is rarely the case in practice.
- Data scarcity is a problem.
- The possibility of losing precision.
- If a categorical variable's category is not present in the training dataset, the model assigns 0 probability to that category, and hence no prediction can be produced.

Random Forest

- Guided learning is used to create the random forest algorithm.
- It may be used to solve problems involving regression and classification.
- Random Forest is a collection of various decision-tree algorithms with random sampling, as the name indicates.
- The goal of this algorithm is to eliminate the flaws in the decision-making algorithm.
- The objective is to make the forecast more exact by averaging or averaging the results of many decision trees.
  - The more decision trees there are, the more accurate the outcome.

Advantages

- The random forest algorithm, as previously stated, may be implemented to overcome both regression and classification problems.
- It's simple to use.
- The random forest algorithm does not have a problem with the dataset being overfit.
- It may be used to determine which function is the most significant among those provided. When utilizing hyperparameters, good predictions are frequently made, and they are simple to comprehend.
- The random forest has a great degree of precision, flexibility, and variability.

Disadvantages

- When dealing with real-time applications, the method becomes slow and inefficient as the number of trees grows.
- When opposed to decision making, random forest takes longer.
- It also requires more resources for calculation.

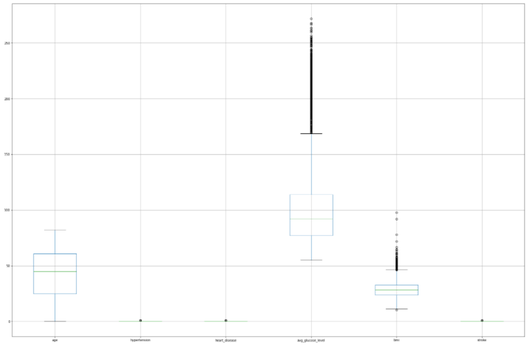

Figure 1: *the outliers of the stroke prediction data set*

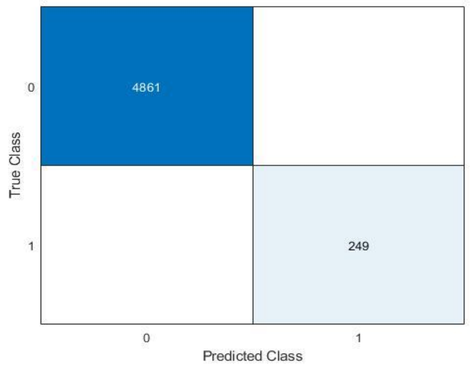

Figure 2: *Confusion matrix of Random Forest*

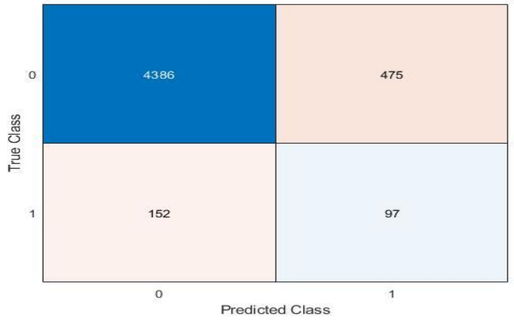

Figure 3: *Confusion matrix of Naïve Bayes*

	NB	RF
Accuracy Score	0.8773	1
Precision	0.9023	1
Recall	0.9665	1
Specifity	0.1696	1
F1 Score	0.9333	1

Figure 4: *Scores for both NB and RF*

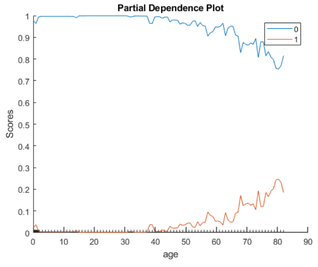

Figure 5: *How likely to get stroke as you get older*

Methodology

- The dataset was split as 75% training data and 25% testing data, where the testing data remains as it is till the end.
- This ended up giving 3832 training observations and 1278 testing observations.
- After obtaining the optimum hyperparameters, evaluating the results on the training set using training, validation, and confusion matrices.
- On the testing data, evaluating the two chosen models with their optimal parameters and comparing the results.
- Not only accuracy but recall and precision were used as parameters to evaluate the resulting model.

Hypothesis

- After reading some research papers, the results within the different ML models will return similar accuracy, but it will be interesting to see which one that will give a more accuracy.
- According to one of the papers that I read; I can assume that the Random Forest model should give me a better accuracy when predicting stroke.

Choice of Parameters and Experimental Results

Naïve Bayes

- For the Naïve Bayes model, the set of predictors to include were selected on the basis of evaluating the partialDependencePlots for each parameter. This concluded with just the predictors *age, hypertension, heart_disease, avg_glucose_level* and *bmi.*
- Further removal of predictors proved to lead to degrading of the accuracy of the model.

Random Forest

- The initial set of predictors chosen for the Random Forest model was also chosen on the basis of partialDependencePlots; predictors were successively removed until the data was left with a set of just 3: *age, avg_glucose_level and bmi.*
- The min leaf size was initially set to 1, but when I changed it to 2, I had a less accurate answer as shown below.

  4858 3
  33 216

- The number of trees used in this project was 500, hence the accuracy of 100%.
  - This can be avoided, and more reasonable and realistic results can be obtained if the random forest had less than 100 trees.
- As part of the data exploration, a logistic regression model was implemented. This model did not predict any stroke at all, which led to realise that the incidence of stroke likely depends on the exploratory variables in a complex way.

Analysis and Critical Evaluation of Results

- The testing accuracy of the cross-validated RF with just three predictors is very good, with a 100 % success score.
- This raises some concern about overfitting; we remark that the *mean* out-of-bag-error is 0.06 for the trees in the random forest.
- Since each tree may be regarded as a realization of a cross-validation step, this suggests that the model is good, though the 100%-success may indicate some overfitting.
- On the other hand, the accuracy score of Naïve Bayes is 87.7 %,.
- Looking at figure 5, it can be seen that a human being is more likely to get stroke as that person gets older; till the age of 40, the probability of getting stroke is very small.
- As I kept getting 100% accuracy in MATLAB, I decided to run these two models on Python as I already did the pre-processing in Python.
  - This showed that RF was a better choice for stroke prediction compared to NB resulting 0.95.

Lessons Learnt

- Due to the imbalance data at the target value the model performance will not be the best considering in the terms precision and recall.
- Looking at the RF, the parameters chosen should be done wisely due to overfitting.

Future Work

- Try using other ML models to train the data.
- Use more of SMOTE to sort out the imbalanced data.

References

- Dave, Parth. "Stroke Prediction Model |How to Create a Stroke Prediction Model?" *Analytics Vidhya*, 24 May 2021, www.analyticsvidhya.com/blog/2021/05/how-to-create-a-stroke-prediction-model/ [Accessed 15 Dec. 2021]
- Glen, S., 2021. *Stroke Prediction using Data Analytics and Machine Learning*. [online] Datasciencecentral.com. Available at: https://www.datasciencecentral.com/profiles/blogs/stroke-prediction-using-data-analytics-and-machine-learning [Accessed 15 December 2021].
- Khosla, A., Cao, Y., Chiung-Yu Lin, C., Chiu, H.-K., Hu, J. and Lee, H. (2010). *An integrated machine learning approach to stroke prediction*. [online] Available at: https://www.researchgate.net/publication/221654326_An_integrated_machine_learning_approach_to_stroke_prediction [Accessed 15 Dec. 2021].
- Sharma, Abhishek. "Decision Tree vs. Random Forest - Which Algorithm Should You Use?" *Analytics Vidhya*, 11 May 2020, www.analyticsvidhya.com/blog/2020/05/decision-tree-vs-random-forest-algorithm/ [Accessed 15 Dec. 2021]
- Ray, Sunil. "6 Easy Steps to Learn Naive Bayes Algorithm (with Code in Python)." *Analytics Vidhya*, 3 Sept. 2019, www.analyticsvidhya.com/blog/2017/09/naive-bayes-explained/.[Accessed 15 Dec 2021]
- Tavares, J.A. (2021). *Stroke prediction through Data Science and Machine Learning Algorithms*. [online] Research Gate. Available at: https://www.researchgate.net/publication/352261064_Stroke_prediction_through_Data_Science_and_Machine_Learning_Algorithms [Accessed 15 Dec. 2021].