CS6140 Final Report Group #14
Group member: Xichen Liu, Keming Xu, Wenshu Yang, Rodolph Mesadieu, Duo Zhao

Predictions on Stroke Dataset

Abstract
Stroke comes with a high morbidity rate, mortality rate, disability rate, recurrence rate and many complications. Stroke is as hazardous as cancer and coronary heart disease. In this case, the project is to predict if a person has a risk of getting a stroke based on physical conditions such as whether the person ever had heart disease, etc. The project will clean a massive dataset which includes many features not correlated to the prediction targets. And then, by configuring multiple models and determining the model and hyperparameters that perform the best, we will use the well-configured model to make the actual predictions and make conclusions on the predicted results.

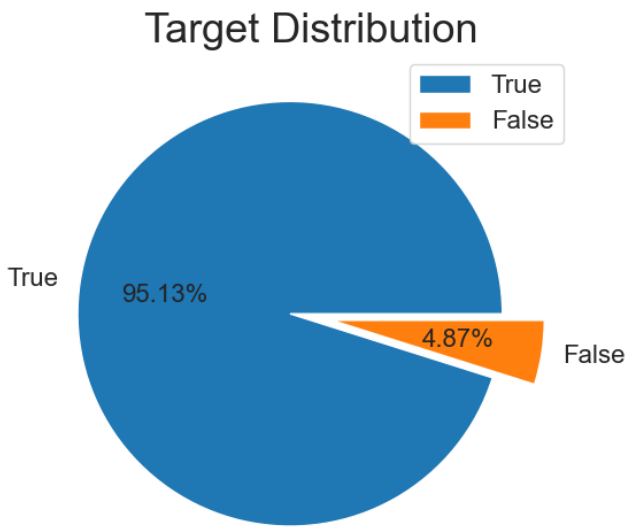
1 Introduction

We obtained the dataset from Kaggle. As for the original dataset, it contains features: patient id, gender, age, ever had hypertension, ever had heart disease, ever married, working type, residence type, average glucose level, bmi index, and smoking frequency. people subconsciously feel that the stroke may be caused by many other diseases. Here, by modeling the data of stroke patients, we may be able to clearly demonstrate how the diseases, habits and customs affect the risk of getting strokes.

2 Our Approach

2.1 Implement EDA(exploratory data analysis) and Cleaning Datasets

The original dataset contains many unhelpful or redundant features that may affect the modeling and predicting part of the project. In this case, we implemented the EDA first to check the correlations between the features and target, or in between the features. And, we also filled out the missing values in the dataset accordingly, which some missing values can be deduced from other features.

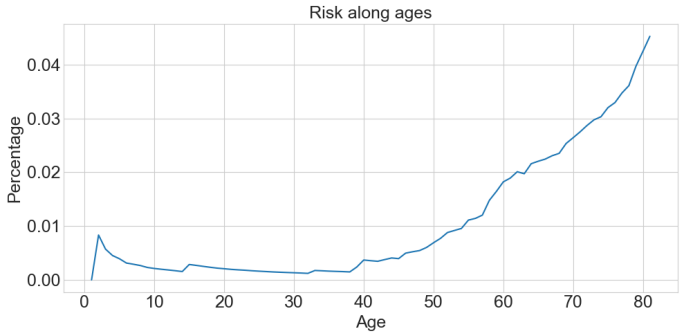


As for the original dataset, some bmi values are missing. We made some predictions based on patients' ages and gender in a simple decision tree regressor. After filling out the missing values, the dataset is good with the contents, which contains no missing values or duplicates.

The next step of our project is implementing EDA to dig more about the correlations to find if we can add more features or shrink the size of features by dropping the redundant.

The target distribution shows that the positive samples and negative samples are extremely imbalanced, which means oversampling is needed here.

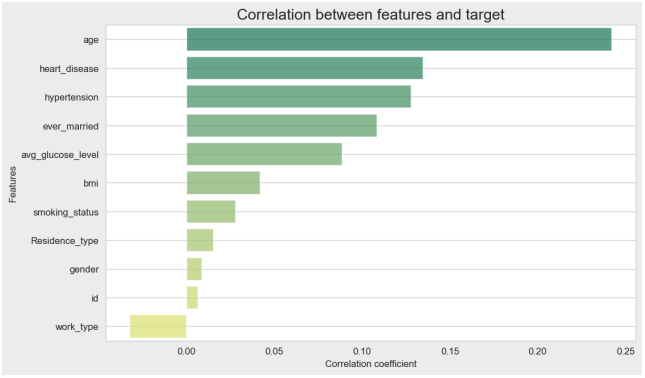
Also, we can plot the risk of getting strokes over the ages. According to the plot of the

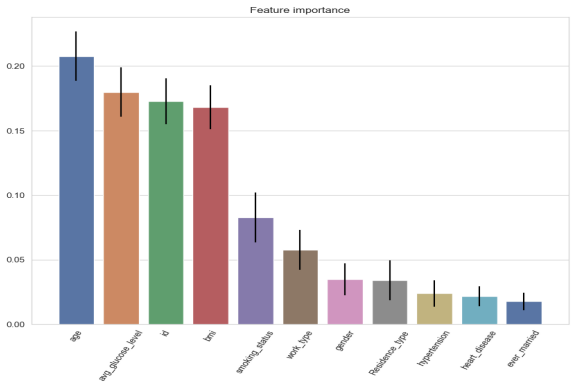


relationship between ages and the risk of getting strokes, we can find that age has a significant influence on the chance that a patient has a stroke. In this case, the age must be kept to train the models and make predictions.

Also, we discovered some correlation between the bmi index, average glucose level and the risk of getting strokes.

From the histogram between bmi index





and stroke risks, and the histogram between average glucose level and stroke risks, we found that the general distribution of whether a patient has stroke or not relates to average glucose level and bmi are similar. In other words, they can be combined into one feature. After that, we dropped the redundant features according to the importance of the features and the correlation between each feature and target.

According to the correlation graph and feature importance graph, it's not hard to find that the target, stroke status, does not depend on gender, residence_type, and work_type that much, so we are good to drop these 3 features.

In other cases, we also applied PCA to have essential components prepared for training and predicting.

2.2 Model Selection

- **K-Nearest Neighbors Classification**
- **Naive bayes classifier**
- **Support Vector Machine**
- **Random Forest Classifier**
- **Decision Tree Classifier**
- **Linear Regression**

2.3 Modeling

2.3.1 K-Nearest Neighbors Classification

K-Nearest Neighbors (KNN) is a supervised classification method for estimating the likelihood taht a data point will become a member of one group or another based on what group the data pointes nearest to it belong to. It is a lazy algorithm which will not train the model at the beginning, but will stores all the datasets and act on at the time of classification.

2.3.2 Naive bayes classifier

For the classifier we implemented in this project assumes a Gaussian distribution for data. For hyperparameter tuning in this model, there is only 1 parameter to tune, which is var_smoothing. This variable is used to add a user-defined value to the distribution's variance. After using the GridSearchCV by setting the range for var_smoothing np.logspace(0,-9, num=100), the value for best result is 1.519911082952933e-07.

2.3.3 Support Vector Machine

Support vector machine is a general linear classifier that conducts dual classification of data according to supervision learning methods. The decision -making boundary is the maximum marginal spacing ultra -plane for learning sample solutions

2.3.4 Random Forest Classifier

In machine learning, the random forest is a classifier containing multiple decision trees, and the category of its output is determined by the number of categories output by individual trees. For many types of materials, it can produce high accuracy classifiers; Its learning process is very fast as well.

2.3.5 Decision Tree Classifier

In machine learning, decision trees are used to represent decisions when given a certain set of data, these trees can be visualized by modeling the decisions via leaf nodes marked with decisions at each split.

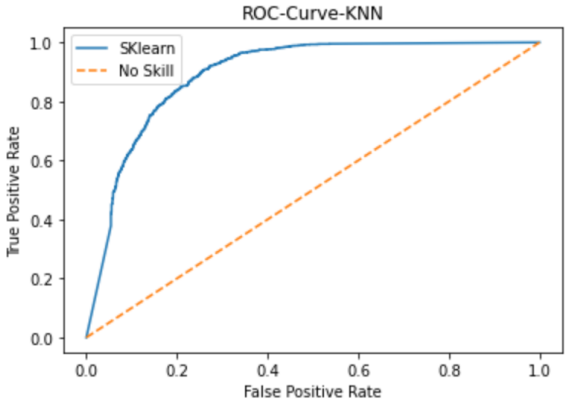
2.3.6 Linear Regression

Linear Regression in machine learning is a learning technique that is used for prediction based on the relationship of variables to one another through regression.

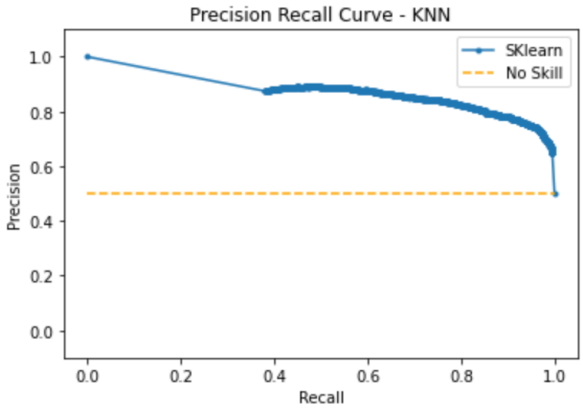
2.4 Results

2.4.1 K-Nearest Neighbors Classification

The precision and recall scores is 83.28% and 82.16%. The F1 score of KNN classification result is 82.72%. The Receiver operating characteristic (ROC) curve for KNN is 82.16%



ROC Curve for KNN

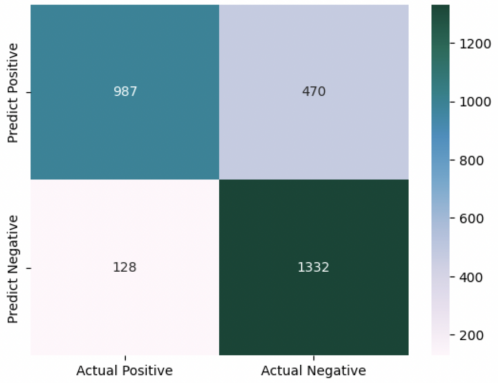


Precision Recall Curve for KNN

2.4.2 Naive bayes classifier

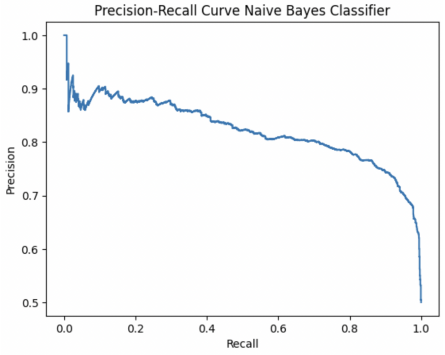
Here is the some statistics for the prediction:

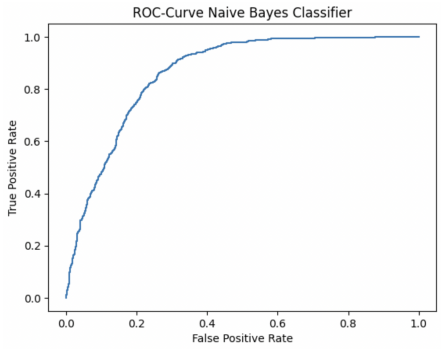
```
Null accuracy score: 0.5005

Model accuracy score: 0.7950
Model precision score: 0.7392
Model recall score: 0.9123
Model f1 score: 0.8167
Training set score: 0.7965
Test set score: 0.7950

Classification report
              precision    recall  f1-score   support

           0       0.89      0.68      0.77      1457
           1       0.74      0.91      0.82      1460

    accuracy                           0.79      2917
   macro avg       0.81      0.79      0.79      2917
weighted avg       0.81      0.79      0.79      2917
```



The model accuracy is 0.795, which is pretty good. Some other factors also need to be considered.
We need to compare the model accuracy with the null accuracy, which means the occurrences of the most frequent class. From the above data we can see, the null accuracy is 0.5, which is much smaller than the model accuracy. We can say the classifier is doing a good job. Also, as we can see, the training set score is 0.7965 and test set score is 0.7950. Two values are comparable. There is no sign of overfitting.

The recall score and precision score are 0.913 and 0.7392. The F1 score is 0.8176. The Precision-Recall Curve and Roc Curve look quite promising.



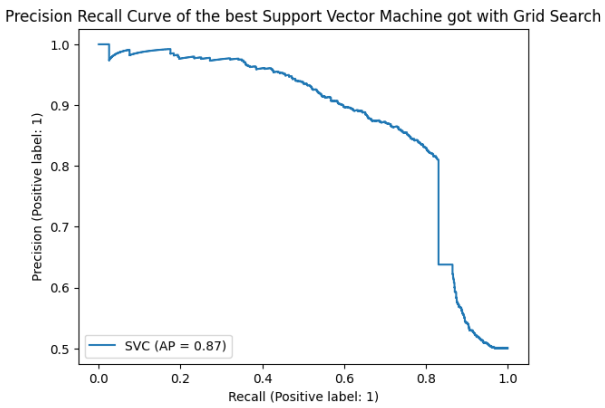


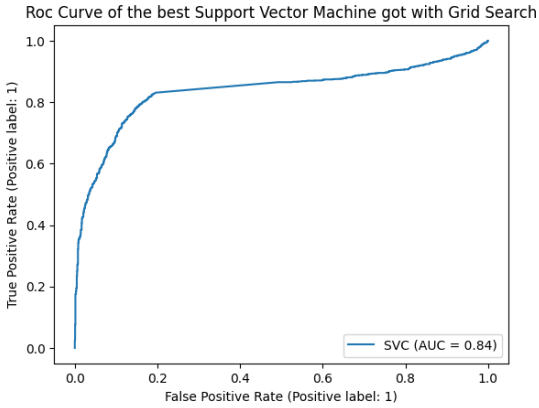
2.4.3 Support Vector Machine

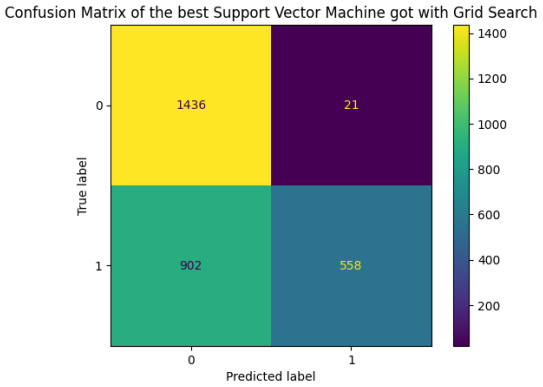
The best hyperparameter of this model is c=100, gamma=0.1. Both Grid Search and Randomized Search support this hyperparameter combination.

The accuracy is ***68.36%***, recall score is 0.382, precision score is 0.9637, and f1 score is 0.547.

These metrics do not look good.





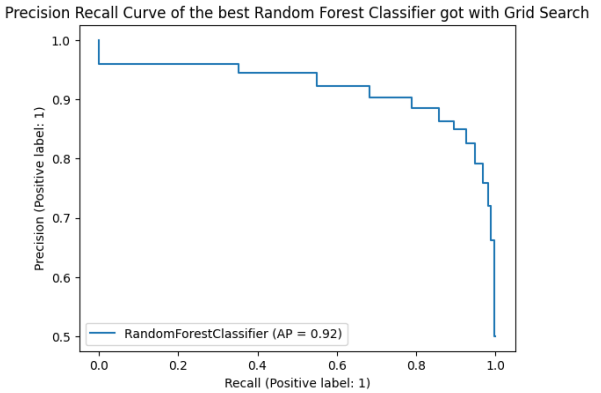


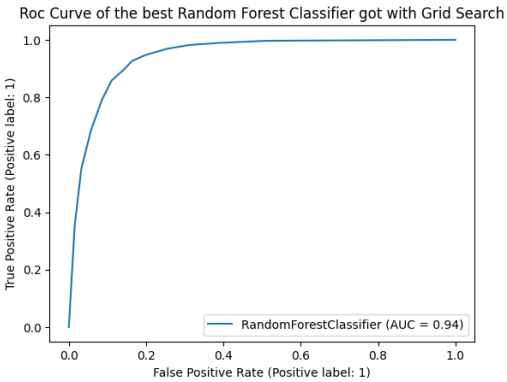
2.4.4 Random Forest Classifier

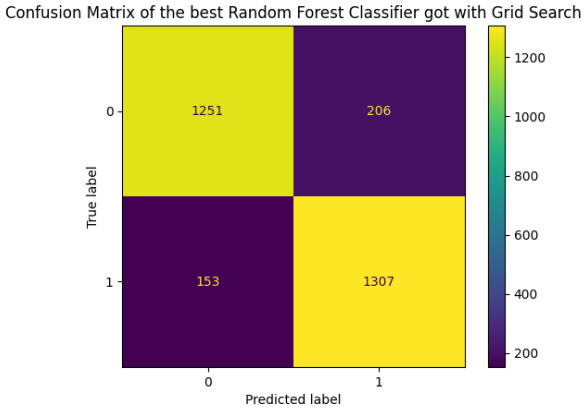
The best hyperparameter of this model is max_depth=1000, max_features="sqrt", min_samples_leaf=1, min_samples_split=2, and n_estimators=12. Both Grid Search got this combination and ite performance is better than the best model got with Randomized Search.

The accuracy is ***87.7%***, recall score is 0.895, precision score is 0.8638, and f1 score is 0.879.

These metric look good.



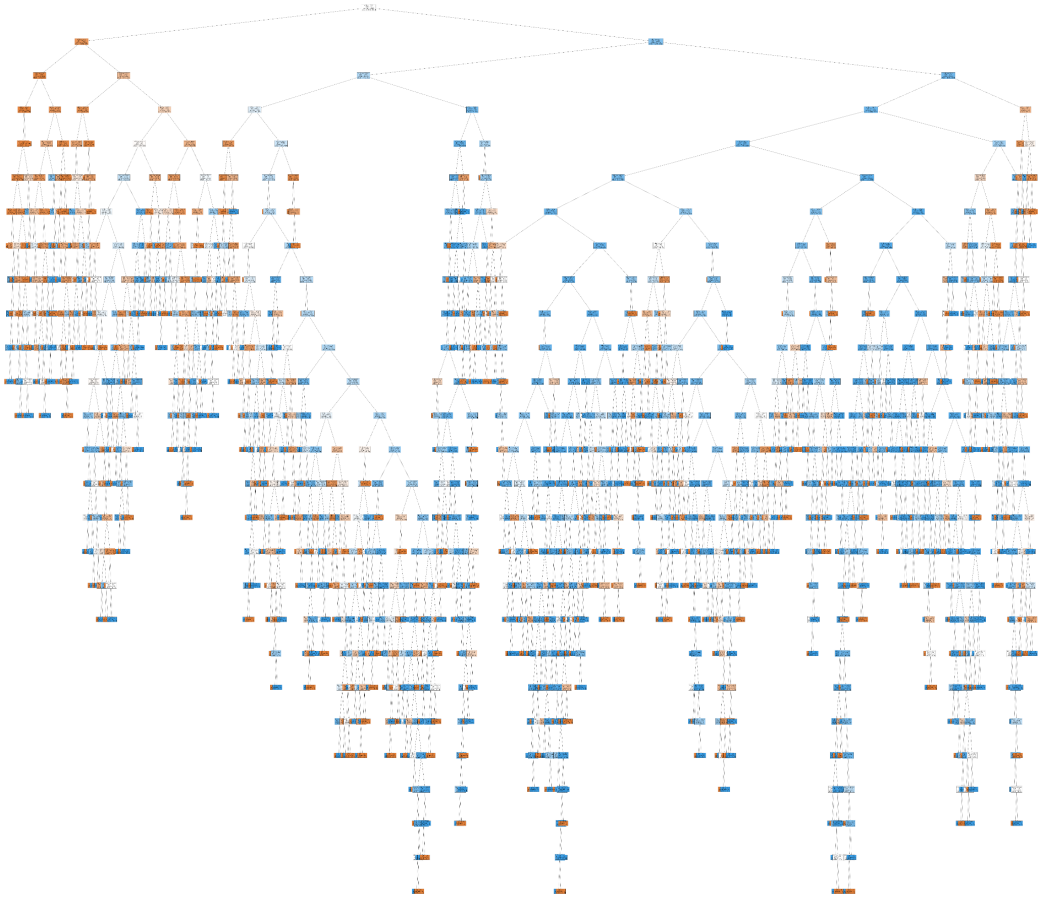


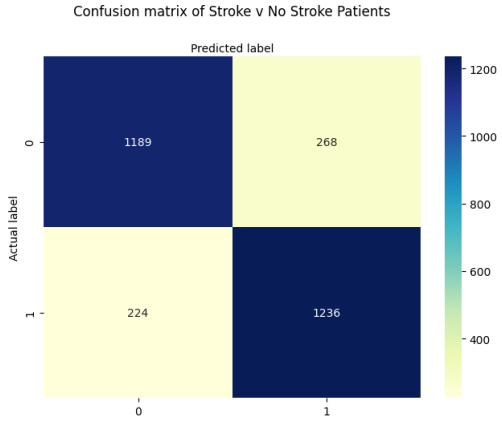


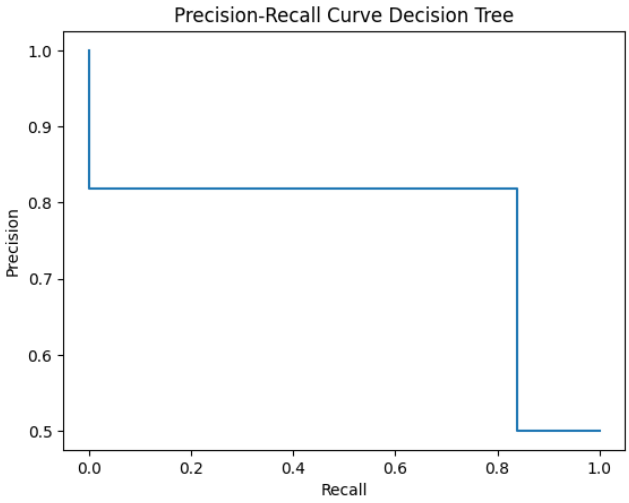
2.4.5 Decision Tree Classifier

The accuracy of this model is **83.1%**, a precision score of **82.18%,** a recall score of **84.66%** and an f1-score of **82.83**%. The metrics are good.

Although the metrics are good, the depth of the decision tree was shown to have over 21 levels when generation was complete. The depth of the tree being so high may point to signs of overfitting within the model. Optimizing the parameters via a grid search would aid in determining an optimal max_depth, but a take away would be in running the decision tree n-teenth times which would be extremely time costly.

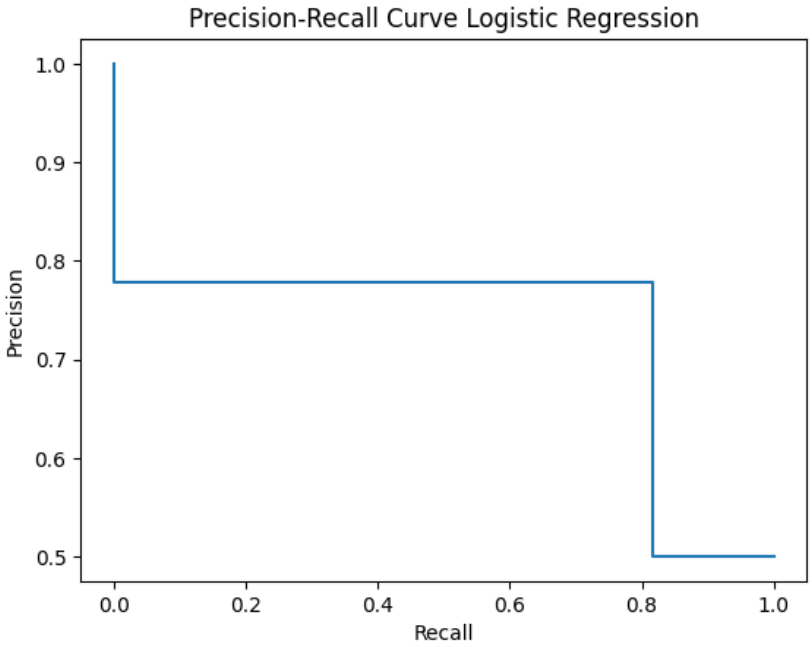






2.4.6 Linear Regression

The Linear Regression metrics were as follows:
Accuracy: 79.22% Precision: 77.87% Recall: 81.71% F1_Score: 79.74%
These are good metrics. Additionally the coefficient of determination was 0.79 with an intercept of -6.519. The .79 coefficient of determination shows that the fit is a good one, with a perfect fit being 1.0.



3 Conclusion

Upon analysis of the data, 6 different classifiers were used and the evaluation matrix of these classifiers shown as below:

Models	**Accuracy**	**Precision**	**Recall**	**F1-Score**
KNN	81.04%	0.8218	0.8103	0.8161
Naive-Bayes	79.5%	**0.913**	0.7392	0.8176
SVM	68.36%	0.382	**0.9637**	0.547
Random Forest	**87.7%**	0.8638	0.895	**0.879**
Decision Tree	83.1%	0.8218	0.8466	0.828
Linear Regression	79.22%	0.7787	0.8171	0.797
Average	79.57%	0.7017	0.8472	0.7826

After the implementation, we found that the Random Forest Classifier reached the highest accuracy which is 87.7% over the six models we chose.

As the only ensemble model among the six models, Random Forest got the best accuracy. It means that ensemble models do have their advantages. In the future, we can explore more ways to ensemble primitive models that can gather the advantage of each of them.

Based on the result of all six models, we found it is possible to predict the stroke using historical data from patients. As society develops at a faster and faster pace, strokes are appearing more frequently. Through the analysis of machine learning, stroke becomes predictable and also provides a reliable basis for scientists to find ways to prevent and treat strokes.

4. Future Work

If we had more time, we would explore more models to get better predictions, like XGBoost classifier, LightBGM classifier, Catboost classifier.

We can also explore more ways to ensemble primitive models that can gather the advantage of each of them.

Reference

Rachidyz. (2021, March 23). EDA and modeling for predicting stroke 🩺 💊 💉. Kaggle. Retrieved from
https://www.kaggle.com/code/rachidyz/eda-and-modeling-for-predicting-stroke

“Predicting a Stroke [SHAP, LIME Explainer & ELI5].” Kaggle.com,
www.kaggle.com/code/joshuaswords/predicting-a-stroke-shap-lime-explainer-eli5.

“Early Stroke Prediction Using Machine Learning” IEEE 2022, Chetan.S, Shamneesh.S
https://ieeexplore.ieee.org/document/9765307.

“Classification And Regression Trees for Machine Learning” 2016, Brownlee.J
https://machinelearningmastery.com/classification-and-regression-trees-for-machine-learning/

“Decision Trees in Machine Learning” 2017, Gupta.P
https://towardsdatascience.com/decision-trees-in-machine-learning-641b9c4e8052

sklearn.ensemble.RandomForestClassifier

https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html

sklearn.svm.SVC — scikit-learn 1.1.3 documentation

https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVC.html