Stroke Prediction and Contributing Factors Using Machine Learning

Thesis · May 2024

DOI: 10.13140/RG.2.2.27861.03045

Stella Okoye
State University of New York at Oswego

Stroke Prediction and Contributing Factors Using Machine Learning.

Department of Biomedical and Health Informatics

State University of New York at Oswego

By: Stella Okoye

sokoye@oswego.edu

ABSTRACT

Stroke ranks as the world's second-leading cause of death, with significant morbidity and financial implications. Early detection is critical, as up to 80% of strokes are preventable. In the United States, stroke is the fifth-leading cause of death, affecting over 795,000 individuals annually. This study explores the role of data mining and machine learning in stroke prediction. Leveraging patient data, machine learning models can forecast stroke occurrence by analyzing key clinical parameters such as age, blood pressure, and glucose levels. Through predictive analytics, healthcare professionals can enhance intervention strategies and improve patient outcomes. By integrating artificial intelligence in medicine, this project aims to develop a robust framework for stroke prediction, ultimately reducing the burden of stroke on individuals and healthcare systems.

Keywords: Stroke, Machine Learning, Prediction Analysis,

INTRODUCTION

Stroke is the world's second-biggest cause of death and one of the most life-threatening diseases for people over 65. It causes brain damage, like how a "heart attack" causes heart damage. It damages the brain in the same way that a "heart attack" damages the heart. When a stroke strikes, it not only costs much money in hospital bills and causes lasting disability, but it can also lead to death. Every 4 minutes, someone dies from a stroke; however, up to 80% of strokes can be avoided if we can detect or forecast stroke onset early on (*Stroke Facts* | *Cdc.Gov*, 2021). Stroke is the fifth-leading cause of death in the United States. According to the Centers for Disease Control and Prevention, stroke is a non-communicable disease that accounts for roughly 11% of deaths. Over 795,000 people in the United States suffer from a stroke daily. (*Stroke Facts* | *Cdc.Gov*, 2021). Many strokes are classified as ischemic embolic and hemorrhagic. An ischemic embolic stroke happens when a blood clot forms away from the patient brain, usually in the patient heart, and travels through the patient's bloodstream to lodge in narrower brain arteries. Hemorrhagic stroke is considered another type of brain stroke, as it happens when an artery in the brain leaks blood or ruptures. (Tazin et al., 2021)

Data Mining assumes an imperative part in the forecast of illnesses in the medical care industry. A significant subject of A.I. in medication is utilized in this project. A machine learning model would take the patient's information and propose many reasonable expectations. The framework can remove concealed information from a chronicled clinical data set and anticipate patients with infection and utilize clinical profiles like age, blood pressure, Glucose, and so forth; it can foresee the probability of patients getting an illness. (Harshitha et al., 2021) Machine Learning can predict the occurrence of a stroke due to advances in medical technology.

OBJECTIVE

This capstone project aims to find strokes using machine learning techniques and various algorithms, such as RFA, DTA, KNN, and SVM, analyzing the accuracy level and evaluating the model performance to find the most efficient ML algorithm for stroke detection.

By applying principles of machine learning over large existing data sets to effectively predict the stroke based on potentially modifiable risk factors such as gender, age, hypertension, heart disease, BMI: body mass index, and smoking status

METHODOLOGY

This section is divided into two parts.

- Data description
- Machine learning classifiers

Data description

The dataset is chosen from Kaggle with various physiological traits to proceed with this project (Amal, 2020). This dataset has 43400 rows and 12 columns. The output column 'stroke' has the value of either '1' or '0'. The value '0' indicates no stroke risk detected, whereas the value '1' indicates a possible risk of stroke. This dataset is highly imbalanced as the possibility of '0' in the output column ('stroke') outweighs that of '1' in the same column. Only 783 rows have the value '1', whereas 42,617 rows with the value '0' in the stroke column. There are seven categorical variables and three numerical variables in the dataset. The dataset discussed above is summarized in Table 1.

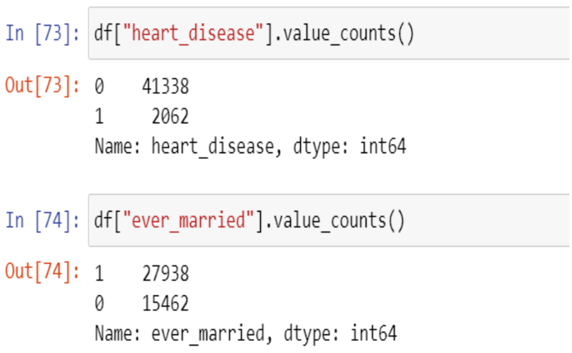
Each patient has the following information:

Attribute Name	**Type (Values)**	**Description**
Id	Integer	A unique integer value for patients
Gender	String literal (Male, Female, Other)	Tells the gender of the patient
Age	Integer	Age of the patient
Hypertension	Integer (1, 0)	Tells whether the patient has hypertension or not

Heart disease	Integer (1, 0)	Tells whether the patient has heart disease or not
Ever married	String literal (Yes, No)	Represents the patient's marital status
Work type	String literal (children, Govt_job, never worked, Private, Self-employed	Represents the patient's working scenario
Residence type	String literal (Urban, Rural)	Represent the patient's living scenario
Average glucose level	Floating point number	It gives the level of the patient's glucose condition.
Body mass index	Floating point number	Gives the value of the patient's body mass index
Smoking status	String literal (formerly smoked, never smoked, smokes, unknown)	It gives the smoking status of the patient
Stroke	Integer (1, 0)	Output column that gives the stroke status

Table 1. Stroke dataset.

The dataset contained 25665 female patients, 17724 male patients, and 11 patients were not specified. 2062 patients are suffering from heart disease, and 41338 patients do not have a record of heart disease. 39339 patients don't suffer from hypertension, while 4061 patients show a history of hypertension.

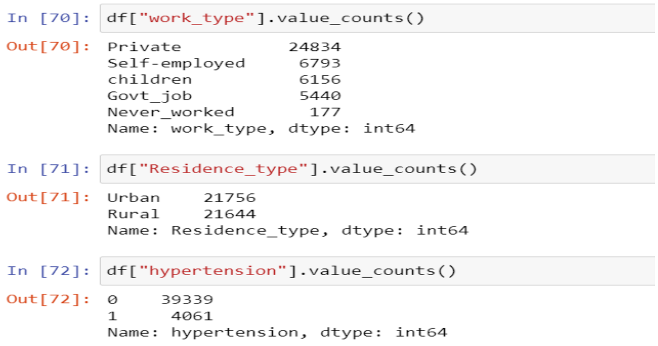


```
In [73]: df["heart_disease"].value_counts()

Out[73]: 0    41338
         1     2062
         Name: heart_disease, dtype: int64

In [74]: df["ever_married"].value_counts()

Out[74]: 1    27938
         0    15462
         Name: ever_married, dtype: int64
```



```
In [70]: df["work_type"].value_counts()
Out[70]: Private          24834
         Self-employed     6793
         children          6156
         Govt_job          5440
         Never_worked       177
         Name: work_type, dtype: int64

In [71]: df["Residence_type"].value_counts()
Out[71]: Urban    21756
         Rural    21644
         Name: Residence_type, dtype: int64

In [72]: df["hypertension"].value_counts()
Out[72]: 0    39339
         1     4061
         Name: hypertension, dtype: int64
```

RESULT

Aim1: summarizing stroke characteristics to discover risk factors.

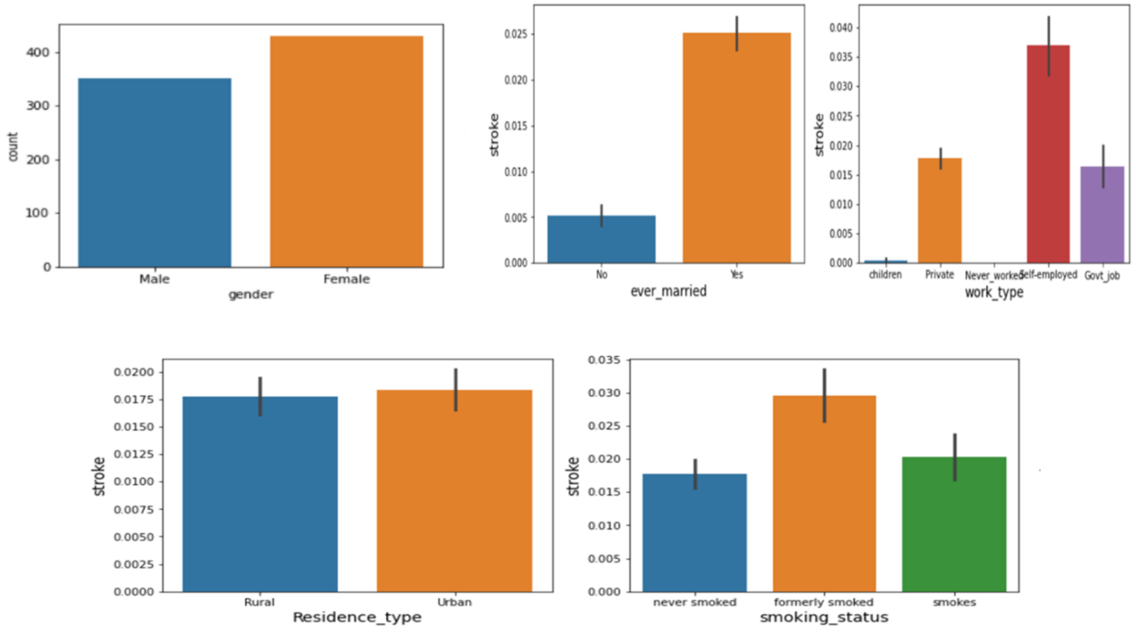
Exploratory Data Analysis

In Exploratory Data Analysis (EDA), data sets are understood by summarizing their essential properties and visualizing them visually, which is referred to as visual data analysis. This stage is crucial, especially when it comes to modeling the data to apply machine learning techniques. Histograms, Box plots, Scatter plots, and other types of plots are available in EDA, among other things. It is common for data exploration to require a significant amount of time.

There are seven categorical variables and three numerical variables.

Categorical variable

From the plots of each categorical variable vs. the target variable "stroke," we see that males and females seem to have about the same chance of getting a stroke, but females had more stroke occurrence than males for this dataset. If a person is ever married, the chance of having a stroke seems to be much higher. The residence type does not seem to make a difference. If a person smokes or formerly smokes, have a higher chance of getting a stroke. Self-employed people have a much higher chance of getting a stroke. One possible reason maybe they endue higher stress. If a person has hypertension or heart disease, the chance of having a stroke seems to grow triple. The plots below show each of the categorical variables vs. the target variable



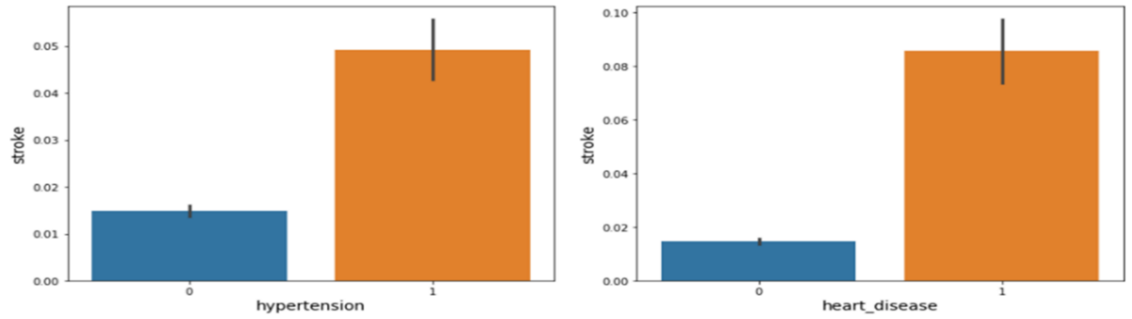


Figure 1.5. Plots of categorical variables vs. target variable

Numerical variables

From the plots of the distributions of the three numerical variables, we see that stroke can happen in all ages, and the risk of having stroke increases by age after the age of 40. If a person has pre-diabetes or diabetes (average glucose level above 150 mg/dL), the risk will increase. Those with a higher glucose level and a BMI between 30 and 35 are also observed to have a higher probability of getting a stroke.

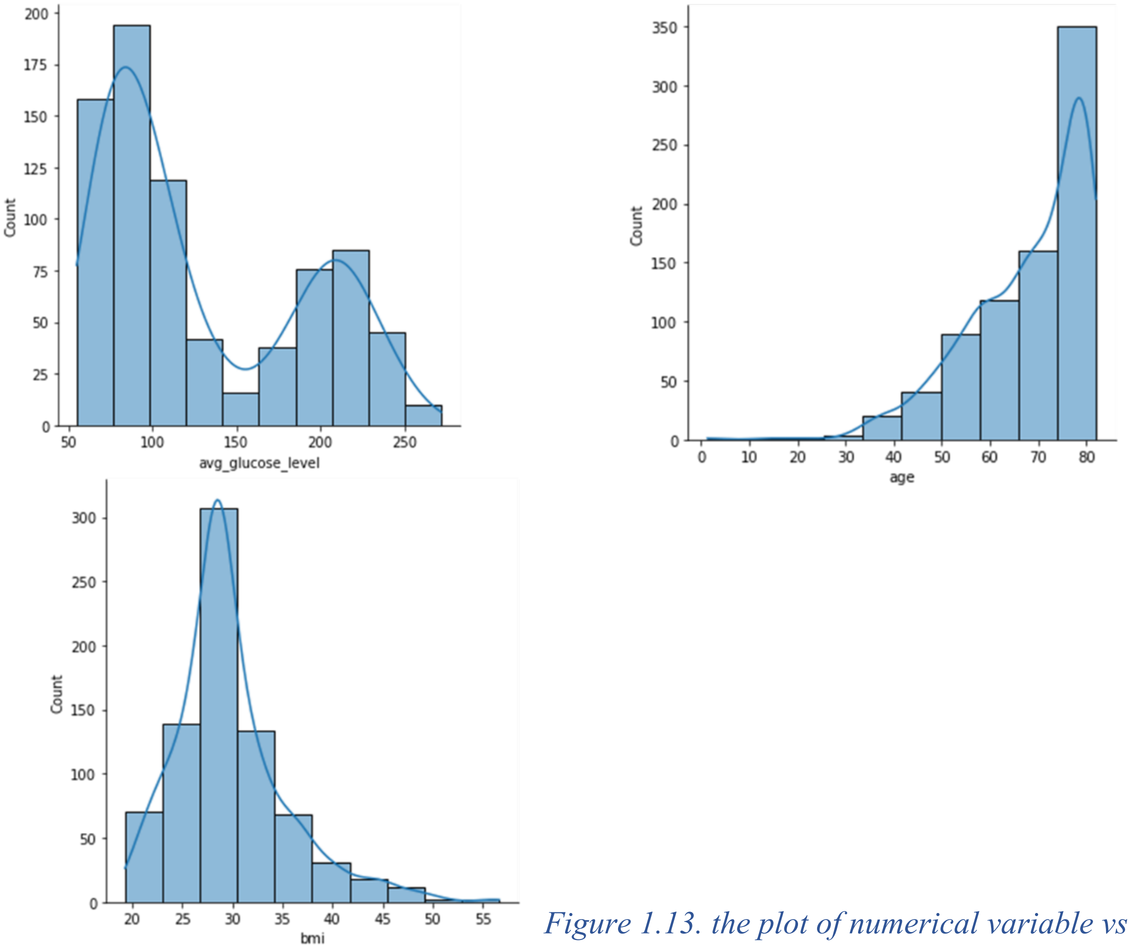
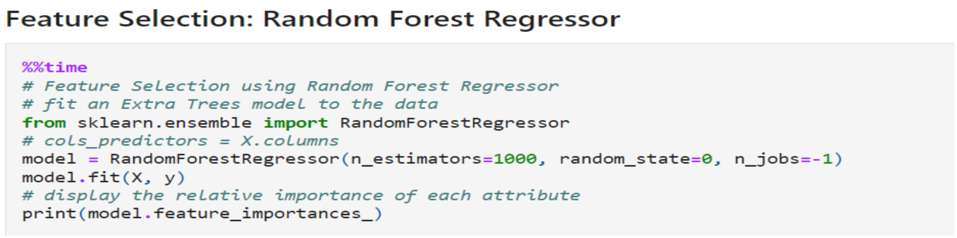


Figure 1.13. the plot of numerical variable vs. target variable

Aim 2: Identify the risk factors that are most predictive of stroke based on available data.

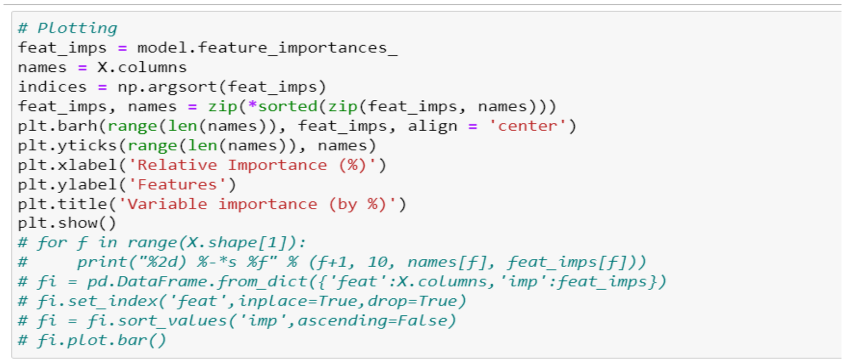
Feature selection

When working with a dataset, feature selection is the process of picking out the most significant features from the dataset. Feature Selection can also improve the performance of a machine learning model in several situations.



Feature Selection: Random Forest Regressor

```
%%time
# Feature Selection using Random Forest Regressor
# fit an Extra Trees model to the data
from sklearn.ensemble import RandomForestRegressor
# cols_predictors = X.columns
model = RandomForestRegressor(n_estimators=1000, random_state=0, n_jobs=-1)
model.fit(X, y)
# display the relative importance of each attribute
print(model.feature_importances_)
```



```
# Plotting
feat_imps = model.feature_importances_
names = X.columns
indices = np.argsort(feat_imps)
feat_imps, names = zip(*sorted(zip(feat_imps, names)))
plt.barh(range(len(names)), feat_imps, align = 'center')
plt.yticks(range(len(names)), names)
plt.xlabel('Relative Importance (%)')
plt.ylabel('Features')
plt.title('Variable importance (by %)')
plt.show()
# for f in range(X.shape[1]):
#     print("%2d) %-*s %f" % (f+1, 10, names[f], feat_imps[f]))
# fi = pd.DataFrame.from_dict({'feat':X.columns,'imp':feat_imps})
# fi.set_index('feat',inplace=True,drop=True)
# fi = fi.sort_values('imp',ascending=False)
# fi.plot.bar()
```

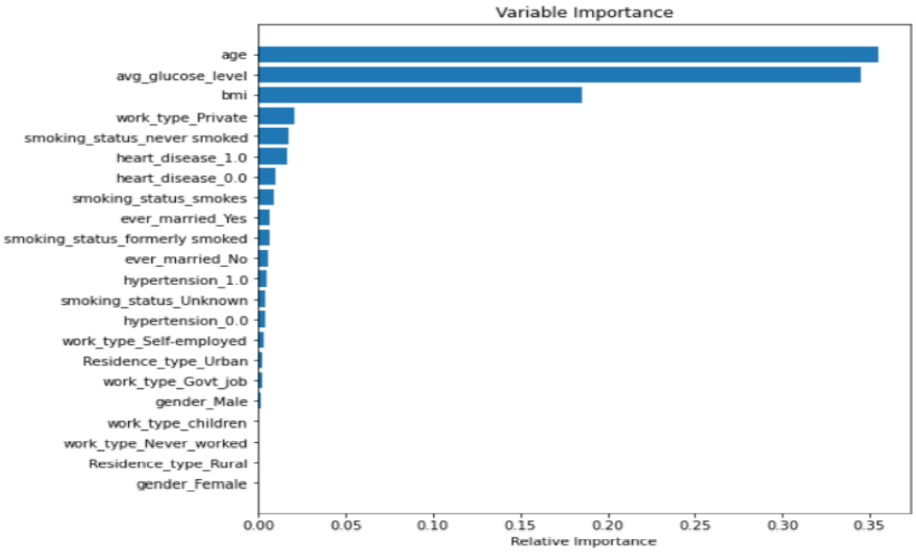
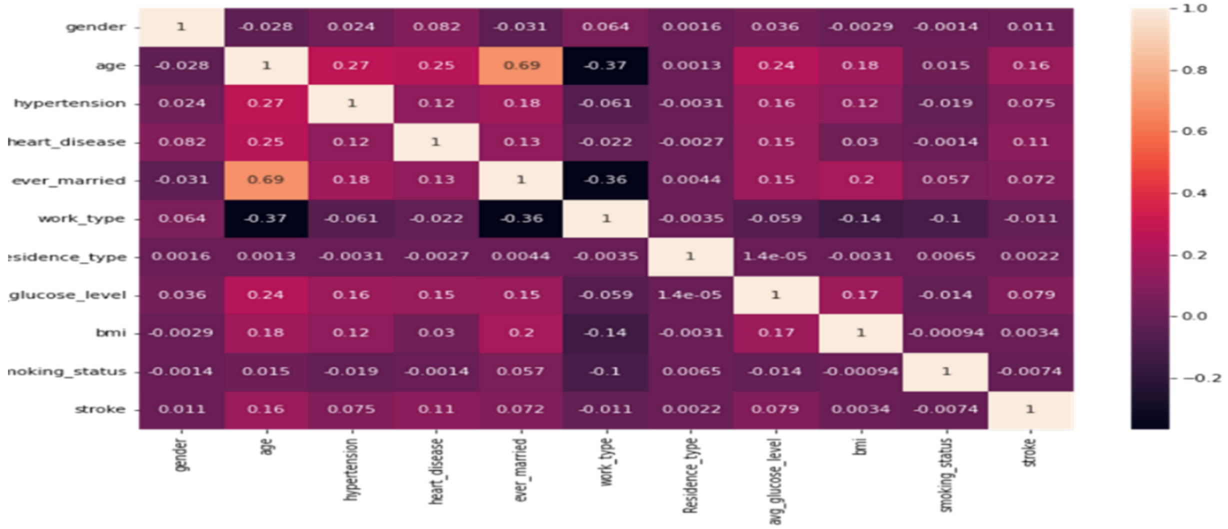


Figure 15. Feature selection

From the figure above, among all the features, we found that age, average glucose level, and BMI are the top three most important features. Just the age and the average glucose level contribute almost 70% of the importance.

Heatmap correlation

Heat Maps are a form of visualization required when we need to identify the dependent variables in a dataset. The use of heat maps is one of the most effective methods of determining the relationship between the features. From the below figure, we can verify the presence of multicollinearity between some of the variables. We can see that- work_type and ever_married negatively correlate with stroke, and age positively correlates with stroke.



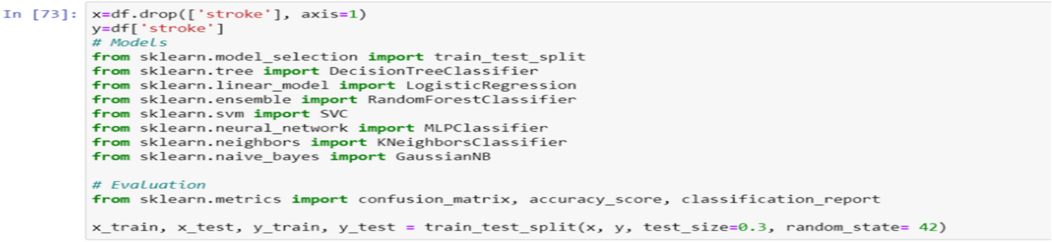
	gender	age	hypertension	heart_disease	ever_married	work_type	Residence_type	avg_glucose_level	bmi	smoking_status	stroke
gender	1	-0.028	0.024	0.082	-0.031	0.064	0.0016	0.036	-0.0029	-0.0014	0.011
age	-0.028	1	0.27	0.25	0.69	-0.37	0.0013	0.24	0.18	0.015	0.16
hypertension	0.024	0.27	1	0.12	0.18	-0.061	-0.0031	0.16	0.12	-0.019	0.075
heart_disease	0.082	0.25	0.12	1	0.13	-0.022	-0.0027	0.15	0.03	-0.0014	0.11
ever_married	-0.031	0.69	0.18	0.13	1	-0.36	0.0044	0.15	0.2	0.057	0.072
work_type	0.064	-0.37	-0.061	-0.022	-0.36	1	-0.0035	-0.059	-0.14	-0.1	-0.011
esidence_type	0.0016	0.0013	-0.0031	-0.0027	0.0044	-0.0035	1	1.4e-05	-0.0031	0.0065	0.0022
glucose_level	0.036	0.24	0.16	0.15	0.15	-0.059	1.4e-05	1	0.17	-0.014	0.079
bmi	-0.0029	0.18	0.12	0.03	0.2	-0.14	-0.0031	0.17	1	-0.00094	0.0034
noking_status	-0.0014	0.015	-0.019	-0.0014	0.057	-0.1	0.0065	-0.014	-0.00094	1	-0.0074
stroke	0.011	0.16	0.075	0.11	0.072	-0.011	0.0022	0.079	0.0034	-0.0074	1

Figure 16. Heatmap correlation

Aim 3: Build predictive models of stroke based on available data.

Data analysis

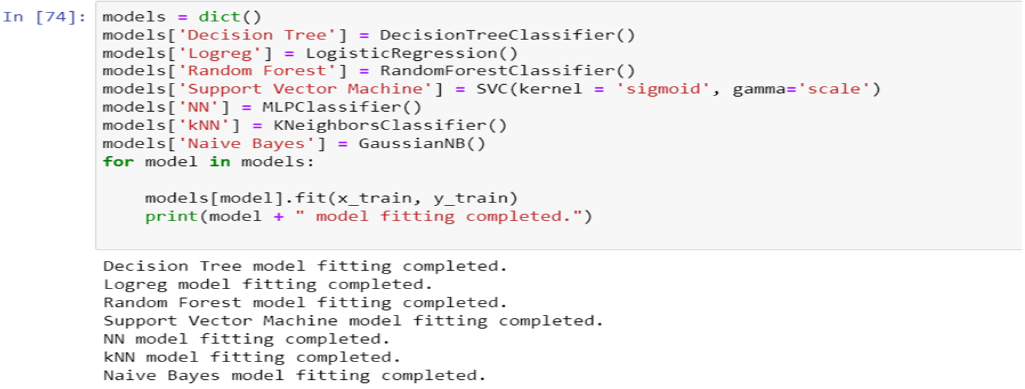
The results obtained by applying the machine learning classifiers are shown in the section. Seven machine learning algorithms will be explored for this dataset to see which produces reliable and repeatable results. All the predictor variables will be mapped to an array x and the target variable to an array y. The target variable is the 'stroke' column.



```
In [73]: x=df.drop(['stroke'], axis=1)
         y=df['stroke']
         # Models
         from sklearn.model_selection import train_test_split
         from sklearn.tree import DecisionTreeClassifier
         from sklearn.linear_model import LogisticRegression
         from sklearn.ensemble import RandomForestClassifier
         from sklearn.svm import SVC
         from sklearn.neural_network import MLPClassifier
         from sklearn.neighbors import KNeighborsClassifier
         from sklearn.naive_bayes import GaussianNB

         # Evaluation
         from sklearn.metrics import confusion_matrix, accuracy_score, classification_report

         x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.3, random_state= 42)
```



```
In [74]: models = dict()
         models['Decision Tree'] = DecisionTreeClassifier()
         models['Logreg'] = LogisticRegression()
         models['Random Forest'] = RandomForestClassifier()
         models['Support Vector Machine'] = SVC(kernel = 'sigmoid', gamma='scale')
         models['NN'] = MLPClassifier()
         models['kNN'] = KNeighborsClassifier()
         models['Naive Bayes'] = GaussianNB()
         for model in models:

             models[model].fit(x_train, y_train)
             print(model + " model fitting completed.")

         Decision Tree model fitting completed.
         Logreg model fitting completed.
         Random Forest model fitting completed.
         Support Vector Machine model fitting completed.
         NN model fitting completed.
         kNN model fitting completed.
         Naive Bayes model fitting completed.
```

Figure 17. Model building

Prediction-error metrics and model selection

A convenient tool for performance evaluation is the so-called confusion matrix, a square matrix consisting of columns and rows that list the number of instances as "actual class" vs. "predicted class" ratios. A confusion matrix for a simple "Stroke vs. No Stroke" classification could look like this:

	No Stroke	Stroke
No Stroke	True Positive	False Negative
Stroke	False Positive	True Negative

Table 2. Confusion matrix

Algorithms	**True positive**	**False-positive**	**False-negative**	**True negative**
Logistic regression	12785	0	235	0
Decision tree	12522	213	263	22
Random forest	12783	2	235	0
Support vector machine	12582	223	203	12
Neural network	12785	0	235	0
K-nearest neighbor	12781	4	235	0
Navies Bayes	11925	860	159	76

Table 3. Values obtained for confusion matrix using different algorithms

The terms "accuracy" and "error" are frequently used to describe categorization performance. Accuracy is the percentage of correct classifications out of a total number of samples; it is sometimes used interchangeably with specificity/precision, even though they are calculated differently. Accuracy is calculated as:

$$\text{Accuracy} = \frac{\boldsymbol{TP+TN}}{\boldsymbol{TP+TN+FP+FN}}$$

- T.P. True positive: the patient has the disease, and the test is positive.
- F.P. False positive: the patient does not have the disease, but the test is positive.
- T.N. True negative: the patient does not have the disease, and the test is negative.
- F.N. False negative: the patient has the disease, but the test is negative.

Other indicators for classification performances are Sensitivity, Specificity, Recall, and Precision.

Sensitivity (synonymous to recall) and precision are assessing the "True Positive Rate" for a binary classification problem: The probability of making a correct prediction for a "positive/true" case (e.g., to predict disease, the disease is correctly predicted for a patient who truly has this disease).

$$\text{Sensitivity}= \frac{\boldsymbol{TP}}{\boldsymbol{TP+TN}}$$

Precision= $\frac{TP}{TP+FP}$

Specificity describes the "True Negative Rate" for a binary classification problem: The probability of making a correct prediction for a "false/negative" case (e.g., to predict disease, no disease is predicted for a healthy patient).

Specificity= $\frac{TN}{TN+FP}$

Algorithms	**Precision**	**Recall**	**F1- Score**	**Accuracy**
Logistic regression	0.98	1.00	0.99	98%
Decision tree	0.98	0.98	0.98	96%
Random forest	0.98	1.00	0.99	98%
Support vector machine	0.98	0.98	0.98	97%
Neural network	0.98	1.00	0.99	98%
K-nearest neighbor	0.98	1.00	0.99	98%
Navies Bayes	0.99	0.93	0.96	92%

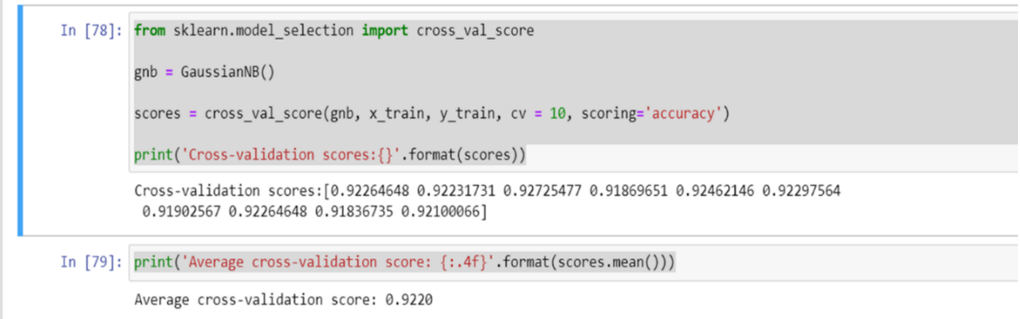
Table 4. Analysis of different machine learning algorithms

From the above accuracy summary, Logistic Regression, Random Forest, neural network, and KNN models all give high accuracy score of 98%. It is also crucial to examine each model's error and recall value. The confusion matrix shows that models with a 98% accuracy score have high false negatives. High false-negative indicates type 2 error. The aim is to prevent type 2 errors in the stroke prediction study since it means it fails to identify people who have had a stroke and instead declares them stroke-free. According to the classification report above, the Naive Bayes Model has met the goal, despite its accuracy of 92.17%.

Cross-validation

Cross-validation is a statistical method for evaluating and comparing learning algorithms that divide data into two segments: one for learning or training a model and the other for validating it. The training and validation sets must cross over in subsequent rounds in traditional cross-validation to validate each data point.

Cross-Validation is a very powerful tool. It helps us better use our data and gives us much more information about our algorithm performance.



```
In [78]: from sklearn.model_selection import cross_val_score

         gnb = GaussianNB()

         scores = cross_val_score(gnb, x_train, y_train, cv = 10, scoring='accuracy')

         print('Cross-validation scores:{}'.format(scores))

         Cross-validation scores:[0.92264648 0.92231731 0.92725477 0.91869651 0.92462146 0.92297564
          0.91902567 0.92264648 0.91836735 0.92100066]

In [79]: print('Average cross-validation score: {:.4f}'.format(scores.mean()))

         Average cross-validation score: 0.9220
```

Figure 18. Cross-validation

Using the mean cross-validation, we can conclude that we expect the model to be around 92.20% accurate on average. The original model accuracy is 92.17%, but the mean cross-validation accuracy is 92.20%. So, the 10-fold cross-validation accuracy results in performance improvement for this model.

AUC – ROC

The AUC ROC curve (Area Under the Curve - Receiver Operating Characteristics) is a performance evaluation for classification problems at various threshold levels. AUC represents the degree or measure of separability, whereas ROC is a probability curve. It indicates how well the model can distinguish between classes. The AUC indicates how well the model predicts 0s as 0s and 1s as 1s. The higher the AUC, the better the model is distinguishing between diseased and unaffected people.

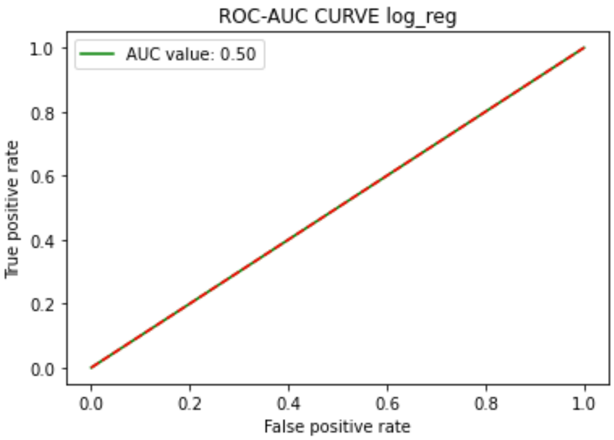


Figure 19. ROC of logistic regression

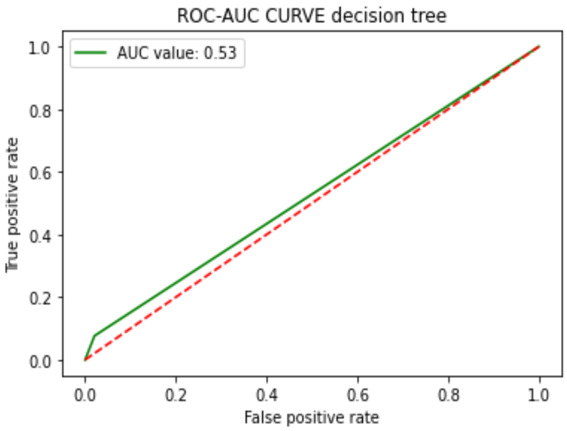


Figure 20. ROC of decision tree

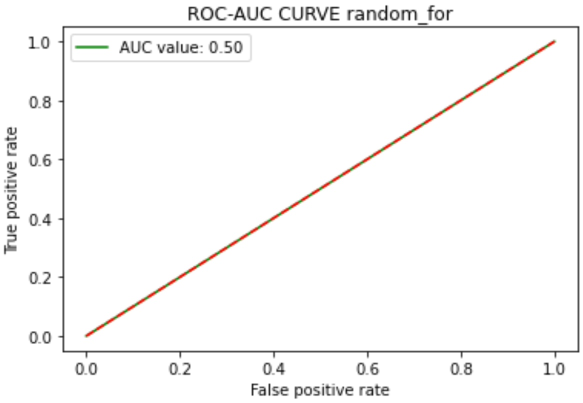


Figure 21. ROC for random forest

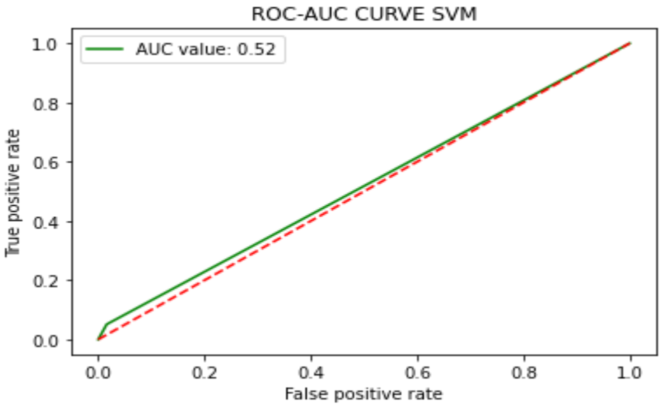


Figure 22. ROC for SVM

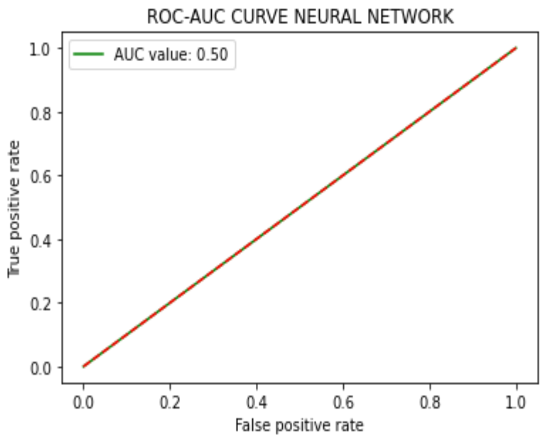


Figure 23. ROC of neural network

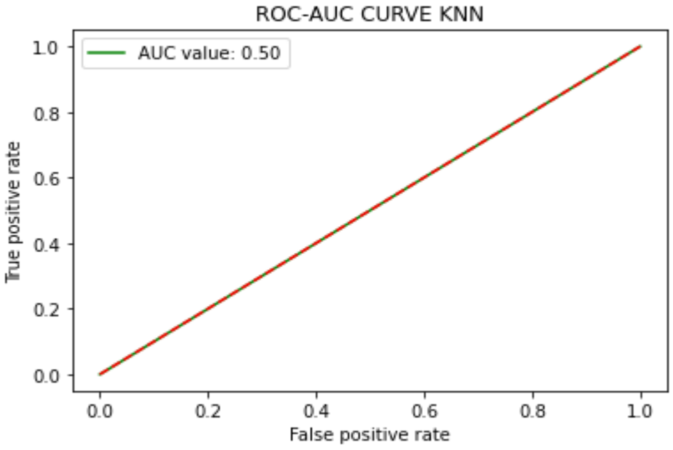


Figure 24. ROC of KNN

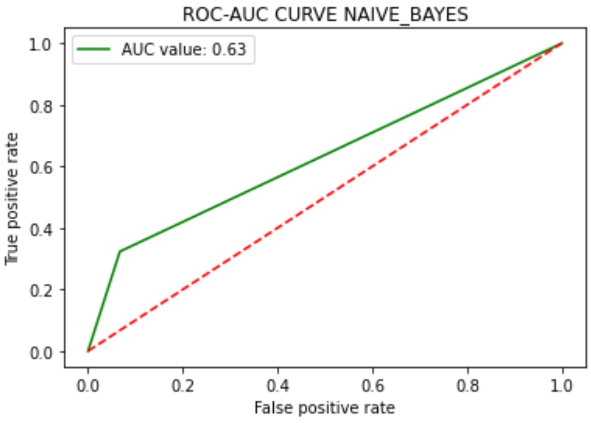


Figure 25. ROC of naive Bayes

From the above figure, the AUC ROC curve indicates how well the naïve Bayes algorithm is a better predictor of stroke for the dataset used in this project study.

CONCLUSION

With the increasing number of deaths due to heart stroke, developing a system to predict heart stroke effectively and accurately has become required. The motivation for the study was to find the most efficient ML algorithm for the detection of heart stroke. This study compares the accuracy score of logistic regression, support vector machine, Random Forest, Decision Tree, neural network, naïve Bayes theorem, and KNN algorithms for predicting heart stroke using the Kaggle dataset. From the above accuracy summary, Logistic Regression, Random Forest, neural network, and KNN models all give high accuracy score of 98%. It is also crucial to examine each model's error and recall value. The confusion matrix shows that models with a 98% accuracy score have high false negatives. High false-negative indicates type 2 error. The aim is to prevent type 2 errors in the stroke prediction study since it means it fails to identify people who have had a stroke and instead declares them stroke-free. This study indicates that the Naïve Bayes algorithm is the most efficient algorithm, with an accuracy score of 92.17% for the prediction of heart stroke.

ACKNOWLEDGEMENT

Supervisor: Isabelle Bichindaritz

REFERENCE

English, C. (Ed.). (2013). *Stroke risk factors and prevention*. Stroke risk factors and prevention - Better Health Channel. Retrieved April 25, 2022, from https://www.betterhealth.vic.gov.au/health/conditionsandtreatments/stroke-risk-factors-and-prevention#risk-factors-of-stroke

McFarland, J., Marti, C., Nedungadi, P., Patterson, K., Jackson, K., & Aron, J. (2015). *Let's talk about complications after stroke*. American Stroke Association. Retrieved April 25, 2022, from https://www.stroke.org/-/media/stroke-files/lets-talk-about-stroke/life-after-stroke/ltas_complications-after stroke.pdf?la=en&hash=98F3D2A38DA974D90F624B33FB036C27570D6D14

Amal, L. (2020, October 26). *Heart stroke*. Kaggle. Retrieved April 25, 2022, from https://www.kaggle.com/lirilkumaramal/heart-stroke

Centers for Disease Control and Prevention. (2021, August 2). *Stroke*. Centers for Disease Control and Prevention. Retrieved January 30, 2022, from https://www.cdc.gov/stroke/index.htm

Centers for Disease Control and Prevention. (2021, May 3). *Know the facts about stroke*. Centers for Disease Control and Prevention. Retrieved January 30, 2022, from https://www.cdc.gov/stroke/facts_stroke.htm

Danesi, M., Okubadejo, N., & Ojini, F. (2007). Prevalence of stroke in an urban, mixed-income community in Lagos, Nigeria. *Neuroepidemiology*, *28*(4), 216–223. https://doi.org/10.1159/000108114

Driscoll, M. D., Jablonski, J., & Stratis, K. (2021, February 27). *Jupyter Notebook: An introduction*. Real Python. Retrieved April 26, 2022, from https://realpython.com/jupyter-notebook-introduction/

Gaurav, H. (2021, May 24). *Five classification algorithms you should know - introductory guide!* Analytics Vidhya. Retrieved April 27, 2022, from https://www.analyticsvidhya.com/blog/2021/05/5-classification-algorithms-you-should-know-introductory-guide/

Harshitha, K. V., Gupta, G., Vaishak, P., & K BPrajna, K. B. (2021). Stroke prediction using machine learning algorithms. *International Journal of Innovative Research in Engineering & Management*, *8*(4), 6–9. https://doi.org/10.21276/ijirem.2021.8.4.2

Jarrell, B., Tadros, A., Whiteman, C., Crocco, T., & Davis, S. M. (2007). National Healthline responses to a stroke scenario. *Stroke*, *38*(8), 2376–2378. https://doi.org/10.1161/strokeaha.107.487710

Kleindorfer, D. O., Khoury, J., Moomaw, C. J., Alwell, K., Woo, D., Flaherty, M. L., ... & Kissela, B. M. (2010). Stroke incidence is decreasing in whites but not in blacks: a population-based estimate of temporal trends in stroke incidence from the Greater Cincinnati/Northern Kentucky Stroke Study. Stroke, 41(7), 1326-1331.

Kshirsagar, A., Goyal, H., Loya, S., & Khade, A. (2021). Brain Stroke Prediction Portal Using Machine Learning. *International Journal for Research in Engineering Application & Management*, *07*(03), 262–268. https://doi.org/ 10.35291/2454-9150.2021.0348

Meel, V. (2022, March 6). *Data preprocessing techniques for machine learning with python*. viso.ai. Retrieved April 29, 2022, from https://viso.ai/deep-learning/data-preprocessing-techniques-for-machine-learning-with-python/

Min, S. N., Park, S. J., Kim, D. J., Subramaniyam, M., & Lee, K.-S. (2018). Development of an algorithm for stroke prediction: A National Health Insurance Database Study in Korea. *European Neurology*, *79*(3-4), 214–220. https://doi.org/10.1159/000488366

Nicholson, C. (2020). *A beginner's Guide to Neural Networks and deep learning*. Pathmind. Retrieved April 28, 2022, from https://wiki.pathmind.com/neural-network

Osuntokun, B. O., Adeuja, A. O. G., Schoenberg, B. S., Bademosi, O., Nottidge, V. A., Olumide, A. O., Lge, O., Yaria, F., & Bolis, C. L. (1987). Neurological disorders in Nigerian Africans: A community-based study. *Acta Neurologica Scandinavica*, *75*(1), 13–21. https://doi.org/10.1111/j.1600-0404.1987.tb07883.x

Sailasya, G., & Kumari, G. L. (2020). Analyzing the performance of stroke prediction using ML classification algorithms. *International Journal of Advanced Computer Science and Applications*, *12*(6). https://doi.org/10.14569/ijacsa.2021.0120662

Tazin, T., Alam, M. N., Dola, N. N., Bari, M. S., Bourouis, S., & Monirujjaman Khan, M. (2021). Stroke disease detection and prediction using robust learning approaches. *Journal of Healthcare Engineering*, *2021*, 1–12. https://doi.org/10.1155/2021/7633381

Wahab, K. W. (2008). The burden of stroke in Nigeria. *International Journal of Stroke*, *3*(4), 290–292. https://doi.org/10.1111/j.1747-4949.2008.00217.x