Stroke outcome binary classification

Authors:
Anuar Nauryzbayev, Amangeldy Sarsembay, Zhannur Dosmailov, Zhanarys Khorshat, Alim Naizabek

Abstract:
In this project, it was necessary to train an algorithm so that it could identify people at risk of death in the event of a stroke. Tables were given as initial data containing information on age, gender, various past diseases, and several health conditions that led and did not lead patients to death after a stroke. Thus, people could be divided into two classes, one class fatal, the other without it. This means that the algorithm had to learn how to determine according to the signs described above, whether a person has a risk of death or not. To do this, we prepared a dataset by cleaning, filtering, and imputing features. We applied both traditional machine learning algorithms and AutoML technology as well. The set of traditional machine learning algorithms included: Naive Bayes, Logistic Regression, Decision trees, and SVM. We used balanced accuracy, confusion matrix, and f1-score. The highest balanced accuracy for the following methods was 0.8. AutoML contains facilities for handling Tabular data. AutoML with a focus on automated stack ensembling allowed the use of several ML algorithms, which give results of learning with accuracy from 0.5 to 0.85. The weighted ensemble achieves 85.0304% balanced accuracy on test data. Therefore, the objectives of this project were achieved.

Introduction

Stroke is a worldwide disease, which can cause the death of the patient [1]. The main issue of this paper is to predict the death or survival of patients diagnosed with stroke using binary classification. Currently, there are huge developments in the stroke prediction domain. Specifically, with the advancement of technology in the healthcare domain and the development of machine learning algorithms, it becomes possible to produce models that can make predictions based on patients' datasets [2]. Similarly, our report is using a patient information-based dataset. This dataset contains refined information from the data collected and anonymized by medical workers of UMC at NU. All data was used with the acknowledgment of the data owner. This dataset was used in binary classification algorithms to train a model that can predict the outcome of a patient with stroke.

Balanced accuracy and f1-score were used as the main metrics of evaluation. AutoML showed the highest performance with the following values 0.85 and 0.84 respectively. Feature selection increased the model's performance. From traditional methods, Decision trees showed values of 0.8 and 0.722 respectively.

The report contains several sections starting with an analysis made by researchers in stroke detection. Then, the methodology of the project was discussed. Then, the dataset and results were discussed.

Literature Review

Today early diagnosis of stroke is an open issue in the healthcare domain. However, as a result of the development of machine learning algorithms and healthcare technologies, it has become possible to predict stroke occurrence in the mass population. This approach is based on the large electronic medical claims database. Electronic medical claim is digital information about patients filed by a doctor. Using big databases of patient information it is possible to build models that can make predictions accurately [2].

Performance metric (95% CI)	Models		
	SVM	SGB	PLR
Accuracy	0.9789 (0.9740–0.9942)	0.9737 (0.9397–0.9914)	0.8947 (0.8421–0.9345)
AUC	0.9783 (0.9569–0.9997)	0.9757 (0.9543–0.9970)	0.8953 (0.8510–0.9396)
Sensitivity	0.9747 (0.9115–0.9969)	0.9512 (0.8797–0.9865)	0.8554 (0.7610–0.9230)
Specificity	0.9820 (0.9364–0.9978)	0.9907 (0.9494–0.9997)	0.9252 (0.8579–0.9671)
Positive predictive value	0.9747 (0.9115–0.9969)	0.9873 (0.9314–0.9996)	0.8987 (0.8101–0.9552)
Negative predictive value	0.9820 (0.9364–0.9978)	0.9640 (0.9103–0.9900)	0.8919 (0.8187–0.9428)

Table 1. [3]

According to table 1, the SGB and SVM have an accuracy of approximately 97% and PLR showed 89% accuracy which is also high. SGB is stochastic gradient boosting, PLR is penalized logistic regression, and SVM is a support vector machine.

KERNEL	Accuracy	Precision	Sensitivity	Specificity	F1 Score
Linear	91 %	84.7%	100 %	78.75 %	91.7
Quadratic	81 %	79.6%	87 %	73.4%	83.3%
RBF	59%	89%	27%	96%	41%
Polynomial	87.9%	84.8%	94.7%	80%	89 %

Table 2. [1]

According to the first source, the Support Vector Machine approach is a successful method of stroke prediction and can achieve 90% accuracy. According to table 2, the highest accuracy is achieved using the linear kernel function in the Support Vector machine approach. Among other kernel functions, the best is the linear function, since it has 91% accuracy and 84.7% precision [1]. SVM that uses polynomial kernel function has 0.01 higher precision than linear kernel function but has 3.1 less accuracy. Other functions have both less accuracy and precision.

This SVM model was trained with different kernel functions. Therefore, the SVM model achieved different performances depending on kernel function [1].

Model	UAR	Sensitivity	Specificity	Accuracy
DNN	0.858	0.845	0.871	0.873
GBDT	0.860	0.856	0.865	0.868
LR	0.841	0.820	0.864	0.866
SVM	0.824	0.813	0.837	0.839

Table 3. [2]

These two methods namely GBDT and DNN according to table 3 showed the best performance among other algorithms. The second source shows that deep neural networks and gradient boosting decision trees are more accurate than logistic regression models and support vector machine methods [2]. However, DNN, LR, GBDT, and SVM have accuracy higher than 80%. Consequently, DNN, LR, GBDT, and SVM are algorithms capable of predicting stroke in patients. Furthermore, in the third source, stochastic gradient boosting, penalized logistic regression, and SVM showed high performance in ischemic stroke prediction. As a result, it is observable that according to previous works it is possible to predict stroke with high accuracy using SVM, DNN, GBDT, LR, SGB, and PLR since they all have accuracy higher than 80%.

According to the information provided in previous works, it is possible to develop models that can successfully predict stroke. Therefore, preventing stroke before patients get this disease. Consequently, if patients get a stroke, it is possible to predict the outcome of the stroke. That means, if available machine models are able to predict stroke, they also can predict the outcome of the stroke. For example, the fourth source discusses the hemorrhagic stroke mortality prediction problem. In this study Jrip, MLP, and J48 machine learning algorithms. Table 4 shows the performance of these machine learning algorithms. J48 is a decision tree machine learning algorithm, Jrip is a RIPPER algorithm and MLP is a multi-layer perceptron algorithm [4].

Data	J48	Jrip	MLP
Accuracy	88.9%	88.1%	88.4%
Precision	88.6%	87.8%	88.2%
Recall	88.9%	88.1%	88.4%
F-measure	88.7%	87.9%	88.3%

Table 4. [4]

In table 4 J48, Jrip, and MLP performance of these algorithms with attribute selection were shown. As a result, using these machine learning methods it is possible to predict the survival of the patients who suffered a hemorrhagic stroke [4]. In this study, our machine learning model will predict the survival of patients with ischemic and hemorrhagic stroke.

Methodology

It was essential to pick the right method to run on our dataset. There are several important aspects of our dataset. It has a relatively small number of samples, most data are binary and have a large ratio of dimension over samples(1/3). Our task was to predict stroke using certain information so we looked up for algorithms that can be used for binary classification.

As the number of samples was small it is better to use simple classification methods to prevent overfitting such as Logistic Regression and Naive Bayes.

As the dimensionality of the dataset is high, the application of Support Vector Machine(SVM) could lead to good classification. Also, researchers on stroke prediction mostly relied on using SVM as it obtained high accuracy in classification.

Moreover, there is a good classification method called Decision Trees. As our dataset contains mostly yes/no(1/0) data, this method can be applied to this classification task.

All chosen classification methods are supervised learning algorithms. You can better familiarize yourself with the methods used in the next sections.

- Logistic Regression

Logistic regression is a supervised learning method to predict some outcome using independent input data. The output of the method is a probability value between 0 and 1. It uses a sigmoid activation function to predict the output[5]:

$$\phi(z) = \frac{1}{1+e^{-z}}$$

- Naive Bayes

Naive Bayes is a probabilistic classification method. This method uses the Bayes theorem to create a generative model. And then this model classifies the dataset[5].

$$y^{new} = argmax\, p(y == c) \prod_{d} P(x_d^{new} | y == c\ ) \text{ [5]}$$

- Support Vector Machine

SVM is a linear algorithm based on a hyperplane and it finds a normal vector w with the smallest norm to the canonical hyperplane. Also, SVM is able to work with non-linear data using kernel trick [5].

- Decision Trees

Decision trees are structures where each node besides leaf nodes makes a binary choice on one feature. There are many algorithms to build such a tree. The most common way to build a tree is to create nodes by splitting based on benefit scores, such as gini. Further development of decision trees is gradient boosting models, such as XGBoost and LightGBM.

$$gini = 1 - \sum(p(x = k))^2$$

Also, we used Gluon AutoML as an alternative approach. It is a highly automated framework for machine learning problems. Its' main advantage is the ease of use. API is very simple and common tasks require less than 15 lines of code. It helps with every step of the model training process. Models are automatically trained with hyperparameter tuning and ensembles are built where appropriate. For binary classification purposes following algorithms are used: KNN, random forest, extra trees, XGBoost, LightGBM, torch, and fastai neural networks, and catboost models. Also, it automates some steps of data preparation, such as filtering categorical unique values, encoding categorical features,

converting text features into trigram count vectors, and doing imputation for certain features types [6].

Automl was primarily used to experiment with dataset variations (features deleted and synthesized). Also, various settings were tested. Every preset was tested and 'best_quality' was selected because model size, training, and inference speed remained in an acceptable range. 60, 120, and 300 seconds training time limits were tested and 120 seconds was selected because further increases did not yield any improvement. Tweaking other settings did not result in significant changes regarding model training or predictive performance [6].

Dataset

The unrefined data come from the University Health Center at Nazarbayev University. Raw data contains information about 150 real patients who were diagnosed with stroke. Data was provided in excel documents containing 5 tables namely Baseline Info, Blood press, Sheet5, Outcomes, and Glasgow CS. The combined number of features(about 70) in all tables was relatively high in comparison to the number of samples(150). In order to decrease the number of features, we used an analytical method by looking only at key factors that could affect stroke.

1. Baseline Info table

   In the Baseline Info table 38 columns with information about patients are given. To discover key factors we searched for the importance of every term. Firstly, we excluded columns that had too much data such as Primary Diagnosis and Comorbidities. Then, columns that are not filled to 50% are excluded. For example, columns containing information about the weight and height of patients have only around 50 samples vs 150 patients. Also, information regarding the date of admission and discharge is not included in the final dataset as they do not have any importance. Additionally, measured values of blood pressure, GCS, and heart rate on admission to the hospital were ignored. In looking at used medications we checked for their purpose as some medications are used only to cure consequences of stroke such as weakness, immune system, etc. The table contained types of used medications but we used only the facts of application as yes or no. One column containing the history of cardiovascular disease was split into several columns: Heart Rhythm disturbance, Arterial hypertension CVD,

Coronary heart disease, and CHF with yes or no labels. Finally, from this table, we used in our classification the following columns: HyperTension, IHD, Diabetes Mellitus, A-Fibrillation, Metabolic, Chronic Renal Failure, Chronic Liver Failure, Hemiparesis, Cerebral edema, Stroke Diagnosis, Arterial hypertension CVD, Coronary heart disease, RHD, Anticoagulants, Vasopressors, Ischemic heart disease, Heart Rhythm disturbance, Age.

2. Sheet5
   This table contained results of observation of patients up to 64 days on Disseminated intravascular coagulation(DIC) occurrence on a particular day. We only summed up the occurrence of DIC for each patient and included this in the dataset.
3. Blood Press
   In the table Blood press blood pressure of patients in the morning and the evening was provided in a time series up to 64 days. However, data on some days was missing and the number of days recorded was different from patient to patient. Therefore, we took the following data to our dataset: the slope of these time series, average systolic blood pressure in the morning, average diastolic blood pressure, average systolic blood pressure in the evening, the standard deviation of systolic blood pressure in the morning, the standard deviation of diastolic blood pressure, the standard deviation of systolic blood pressure in the evening.
4. Glasgow CS
   Following table concluded data of Glasgow Coma Scale(GCS) measurements for each day up to 80 days. Same as in the previous table, data was not presented for each participant daily as some patients were discharged after several days while others were in the hospital. From this table, average and standard deviation values of GCS were calculated and added to the dataset.
5. Outcomes
   This table contained information about some medical conditions of the client. We dropped all columns containing information about the date. Also, we removed text comments and other unique values. From these columns we took the following data into our data set: Shock, MI, Dysrhythmia, Date of dysrhythmia, Hyperglycemia, Heart Failure, Sepsis, Systemic Inflam Syndrome, Recurrent Stroke, Hemorrhagic transformation of stroke,

Pulmonary failure, Renal Failure, Liver failure, Ventilator-Associated Pneumonia, Gastrointestinal Bleeding, Urine Tract Infection, Neurosurgery Performed, Surgical Site Infection.

Our final dataset consisted of 150 samples and 46 features. List of features:

#	Column	Non-Null Count	Dtype
0	DIC Occurrence	150 non-null	int64
1	HyperTension	150 non-null	int64
2	IHD	150 non-null	int64
3	Diabete Mellitus	150 non-null	int64
4	A-Fibrillation	150 non-null	int64
5	Metabolic	150 non-null	int64
6	Chronic Renal Failure	150 non-null	int64
7	Chronic Liver Failure	150 non-null	int64
8	Hemiparesis	150 non-null	int64
9	Cerebral edema	150 non-null	int64
10	Stroke Diagnosis	150 non-null	int64
11	Arterial hypertension CVD	150 non-null	int64
12	Coronary heart disease	150 non-null	int64
13	RHD	150 non-null	int64
14	Anticoagulants	150 non-null	int64
15	Vasopressors	150 non-null	int64
16	Ishemic heart disease	150 non-null	int64
17	Heart Rhythm disturbance	150 non-null	int64
18	Age	150 non-null	int64
19	Shock	150 non-null	int64
20	MI	150 non-null	int64
21	Dysrhythmia	150 non-null	int64
22	Date of dysrhytmia	150 non-null	int64
23	Hyperglycemia	150 non-null	int64
24	Heart Failure	150 non-null	int64
25	Sepsis	150 non-null	int64
26	Systemic Inflam Syndrome	150 non-null	int64
27	Recurrent Stroke	150 non-null	int64
28	Hemmorhagic transformation of stroke	150 non-null	int64
29	Pulmonary failure	150 non-null	int64
30	Renal Failure	150 non-null	int64
31	Liver failure	150 non-null	int64
32	Ventilator Associated Pneumonia	150 non-null	int64
33	Gastr.Intestin. Bleeding	150 non-null	int64
34	Urine Tract Infection	150 non-null	int64
35	Neursurgery Performed	150 non-null	int64
36	Surgical Site Infection	150 non-null	int64
37	Slope	150 non-null	float64
38	Average SBP in am	150 non-null	float64
39	Average DBP	150 non-null	float64
40	Average SBP in pm	150 non-null	float64
41	Stdev	150 non-null	float64
42	Gender	150 non-null	object
43	Average	150 non-null	float64
44	StdevGCS	150 non-null	float64
45	Death	150 non-null	int64

Figure 1. Feature names and data types

We prepared some visualizations to demonstrate some features of the dataset:

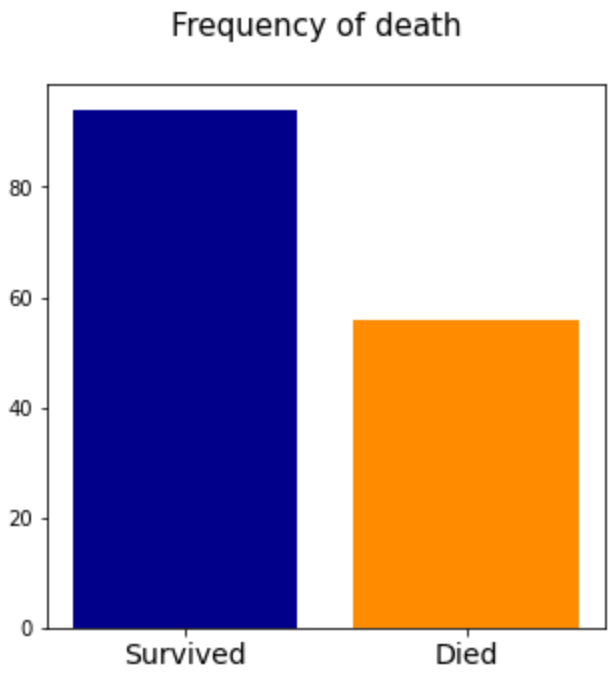


Figure 2. A number of samples for both classes

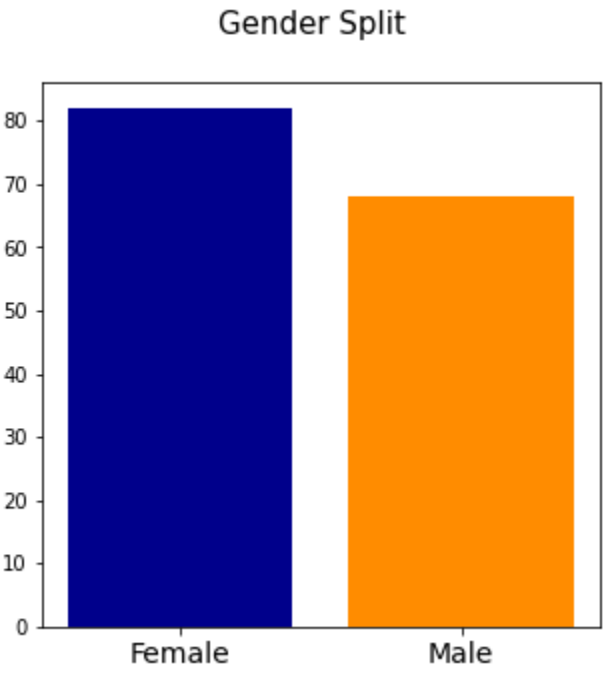


Figure 3. The gender split of patients

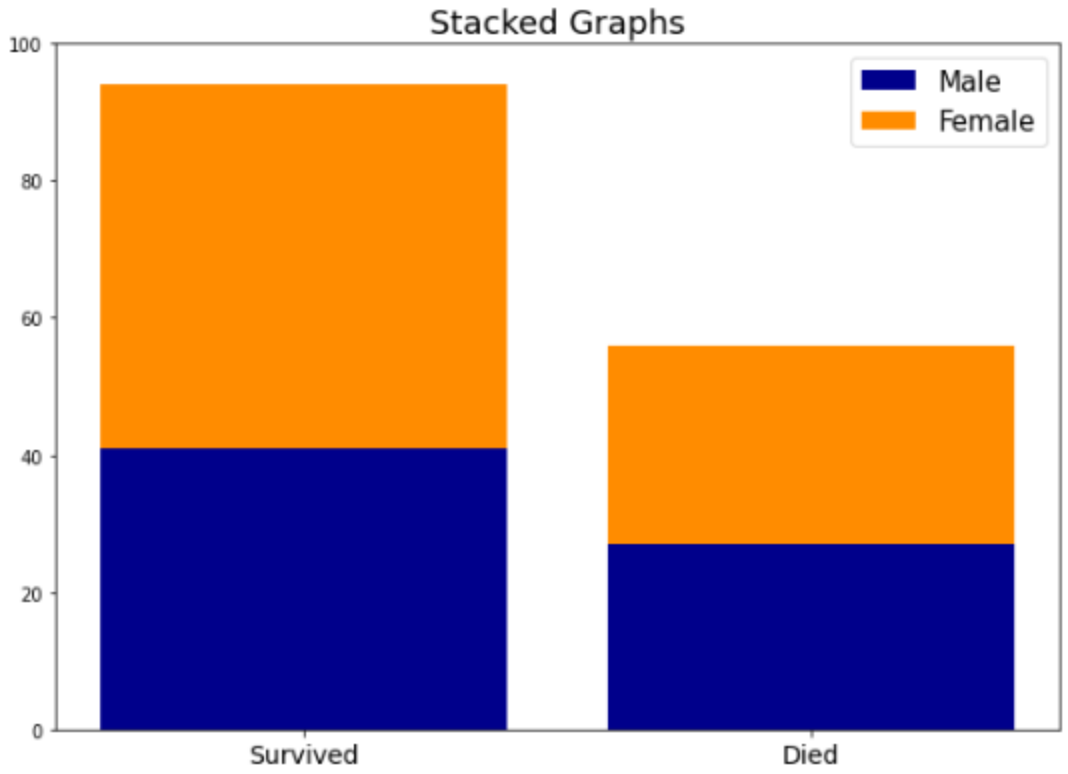


Figure 4. Survival by gender

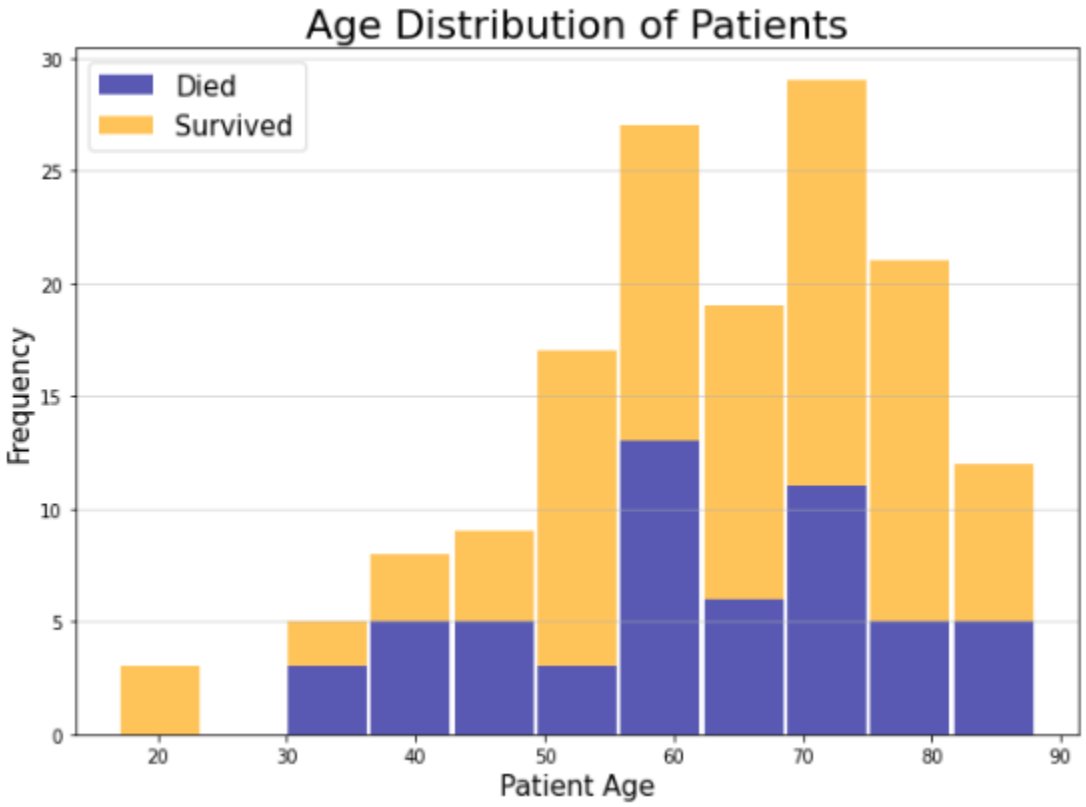


Figure 5. Distribution of patients' age

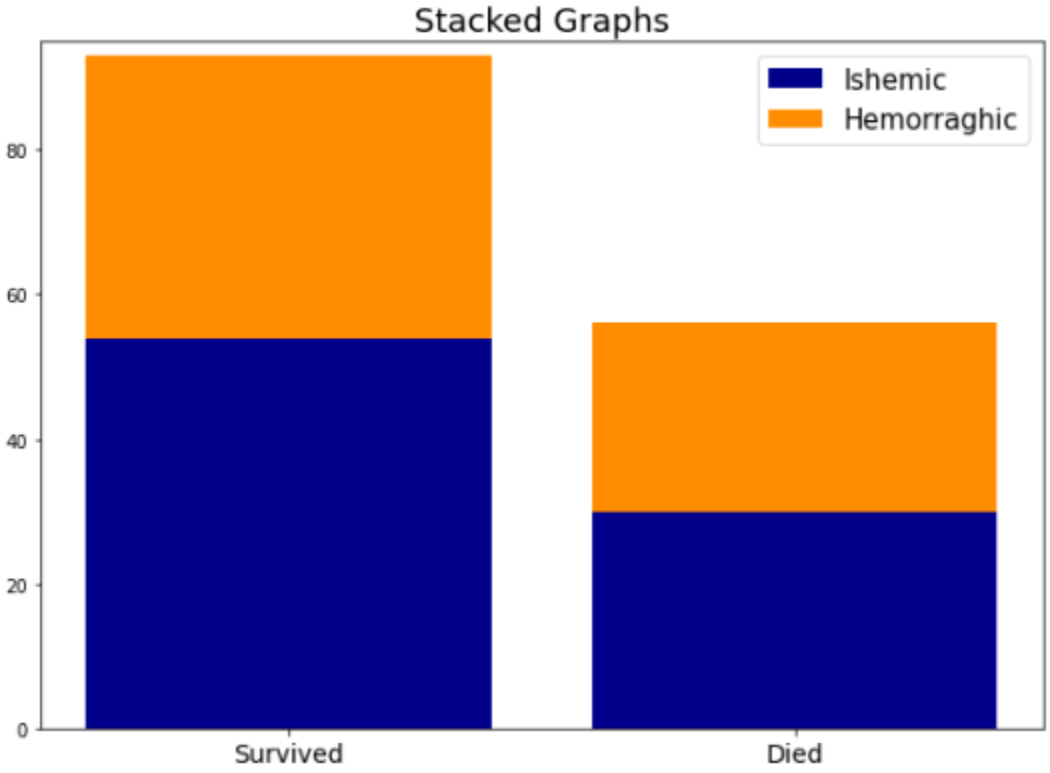


Figure 6. Survival by type of stroke.

Even after the analytical elimination of some features of data, the dataset contained too many features that could negatively affect the accuracy of models. To handle it, we made a data preprocessing in 3 steps.

Firstly, we made a correlation matrix using the seaborn library and pandas built-in function. Using the results of the correlation matrix, we dropped several columns including Diabetes Mellitus, Slope, Metabolic, Urine Tract Infection, and AverageGCS as they had very low correlation with labels. Moreover, the Date of Dysrhythmia column was also dropped as it contained almost the same information as the Dysrhythmia column with a correlation coefficient of 0.91 but a lower correlation coefficient with the label. You can look for the correlation matrix in Appendix A.

Secondly, we used the selectKBest function from sklearn.feature_selection. This function calculates the score for each feature using a chi-squared value so you can see the importance of features by score. After applying this function, we dropped several data points which had scores lower than 0.1 which are located at the bottom in Figure 2.

	Specs	Score
13	Vasopressors	25.277727
36	Stdev	17.218393
27	Renal Failure	12.841945
26	Pulmonary failure	11.505279
18	MI	10.224924
4	Chronic Renal Failure	6.620061
38	StdevGCS	6.280877
7	Cerebral edema	5.538671
33	Average SBP in am	3.934379
28	Liver failure	3.890578
9	Arterial hypertension CVD	3.669127
29	Ventilator Associated Pneumonia	3.410928
17	Shock	3.357143
34	Average DBP	2.276030
23	Systemic Inflam Syndrome	2.163889
2	IHD	2.106711
16	Age	1.780427
15	Heart Rhythm disturbance	1.515139
14	Ishemic heart disease	1.374575
22	Sepsis	1.103343
5	Chronic Liver Failure	1.103343
0	DIC Occurrence	1.082364
10	Coronary heart disease	0.949631
30	Gastr.Intestin. Bleeding	0.685790
1	HyperTension	0.425532
21	Heart Failure	0.376533
24	Recurrent Stroke	0.274316
25	Hemmorhagic transformation of stroke	0.260068
8	Stroke Diagnosis	0.220577
37	Gender	0.163609
12	Anticoagulants	0.091294
19	Dysrhythmia	0.068627
20	Hyperglycemia	0.058282
6	Hemiparesis	0.056991
11	RHD	0.041033
35	Average SBP in pm	0.032760
31	Neursurgery Performed	0.031372
3	A-Fibrillation	0.013263

Figure 7. Selecting best features by score

Afterwards, we applied the ensemble learning method to apply feature selection. ExtraTreeClassifier from sklearn was chosen.

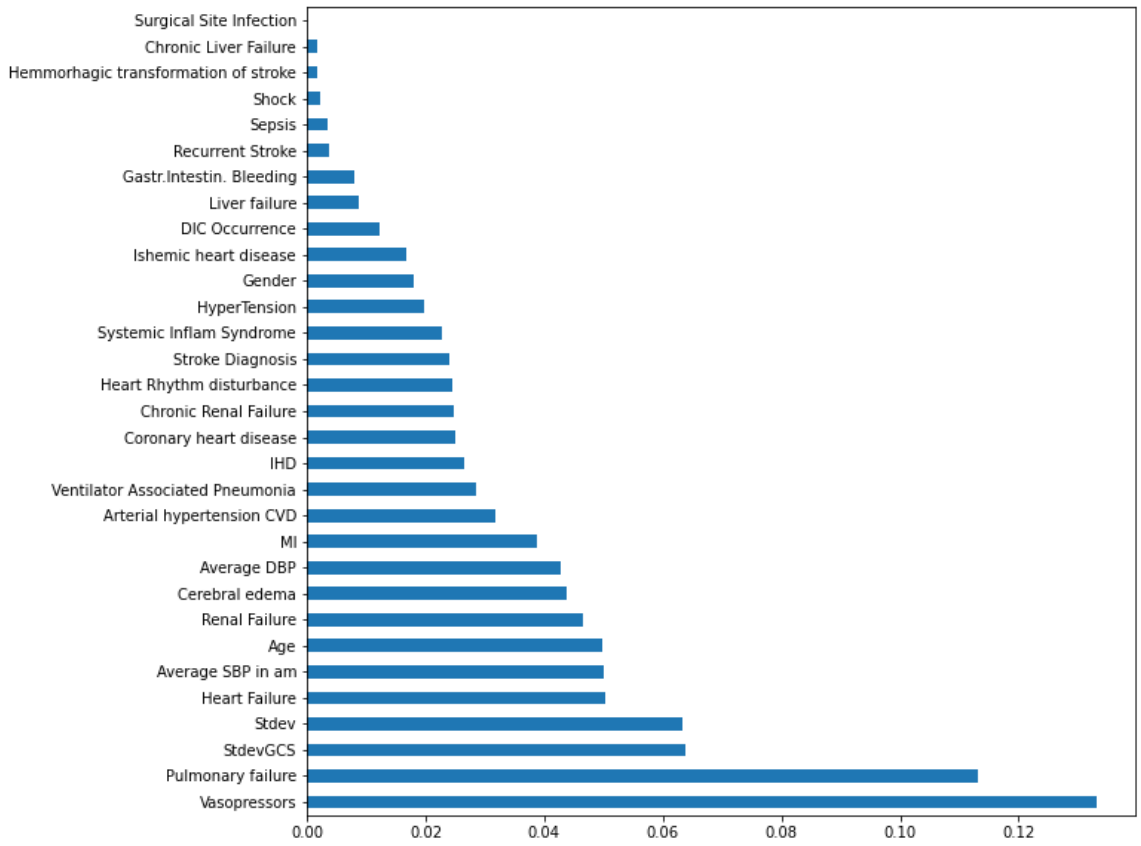


Figure 8. Feature importance values.

After all these techniques, the dimensionality of the dataset is reduced to 26.

Evaluation Metric

Following evaluation, metrics were used: confusion matrix, f1 score, and balanced accuracy. Balanced accuracy is the average of the true positive rate obtained in each class and better on unbalanced data. Because the problem is not balanced (~38% death cases) accuracy is not a good performance metric alone. F1 score is the harmonic mean of precision and recall that includes false positive and false negative ratios which are essential in evaluation. We chose it as the primary metric because it estimates models' ability to predict both classes correctly.

Metrics name	Equation
Balanced Accuracy	$0.5 * (\frac{TP}{TP+FN} + \frac{TP}{TN+FP})$

F1 score	$2 * \frac{precision*recall}{precision + recall}$
Precision	$\frac{TP}{TP+FP}$
Recall	$\frac{TP}{TP+FN}$

Table 5. Equation of each metric

Results & Analysis

Dataset was split into 80% and 20% parts randomly. 80% of the data were used for training and 20% for validation purposes. Moreover, we ran models using two datasets: the original dataset and the dataset after feature selection.

Method	**Balanced accuracy score without feature selection**	**Balanced accuracy score with feature selection**
Naive Bayes	0.722	0.639
Decision trees	0.8	0.708
Support Vector Machine	0.5	0.5
Logistic Regression	0.667	0.722
Weighted ensemble from AutoML	0.84479	0.850304

Table 6. Balanced accuracy with and without feature selection of classification methods that were used.

Method	**F1 score without feature selection**	**F1 score for with feature selection**
Naive Bayes	0.689	0.593
Decision trees	0.722	0.667

Support Vector Machine	0.3	0.3
Logistic Regression	0.615	0.667
Weighted ensemble from AutoML	0.788462	0.839286

Table 7. F1 with and without feature selection scores of classification methods that were used.

All confusion matrices are included in **Appendix 2**.

To analyze, it can be seen from the results that the model's accuracy showed different results on different datasets. Logistic regression showed better results on the dataset after feature selection while others showed slightly fewer results. The highest result was obtained on the Decision trees model with a value of 0.8 without the application of feature selection. F1 scores for all models were relatively small indicating the highest value of 0.722 for Logistic Regression.

In contrast to traditional sklearn models, autoML showed a good performance with a balanced accuracy of 0.85 and an f1-score of 0.839. Moreover, autoML results after feature selection improved slightly.

The following results indicate that feature selection is an effective method to increase the model’s performance. Also, results that indicate more parameter tuning should be done in traditional sklearn models as autoML obtained higher accuracy as it has tuned parameters automatically. We tuned the parameters of the model but it was not enough as the results suggest.

Overall, picked models to perform binary classification showed relatively good performance. However, they have a small accuracy in contrast to reviewed models in the literature. There are several factors that possibly affect models' performance. Firstly, the number of samples in a dataset was very small. 150 samples are not enough to properly train the model on classification problems. Secondly, feature selection from original, unfiltered, and uncleaned datasets required some knowledge in medicine. Because of it, we possibly did select unnecessary features from the dataset that negatively affect the model's performance. Despite all these factors, a metrics score of around 80% indicates that the dataset and model could be considered as good.

Code - https://github.com/Jonerbay/Stroke-project

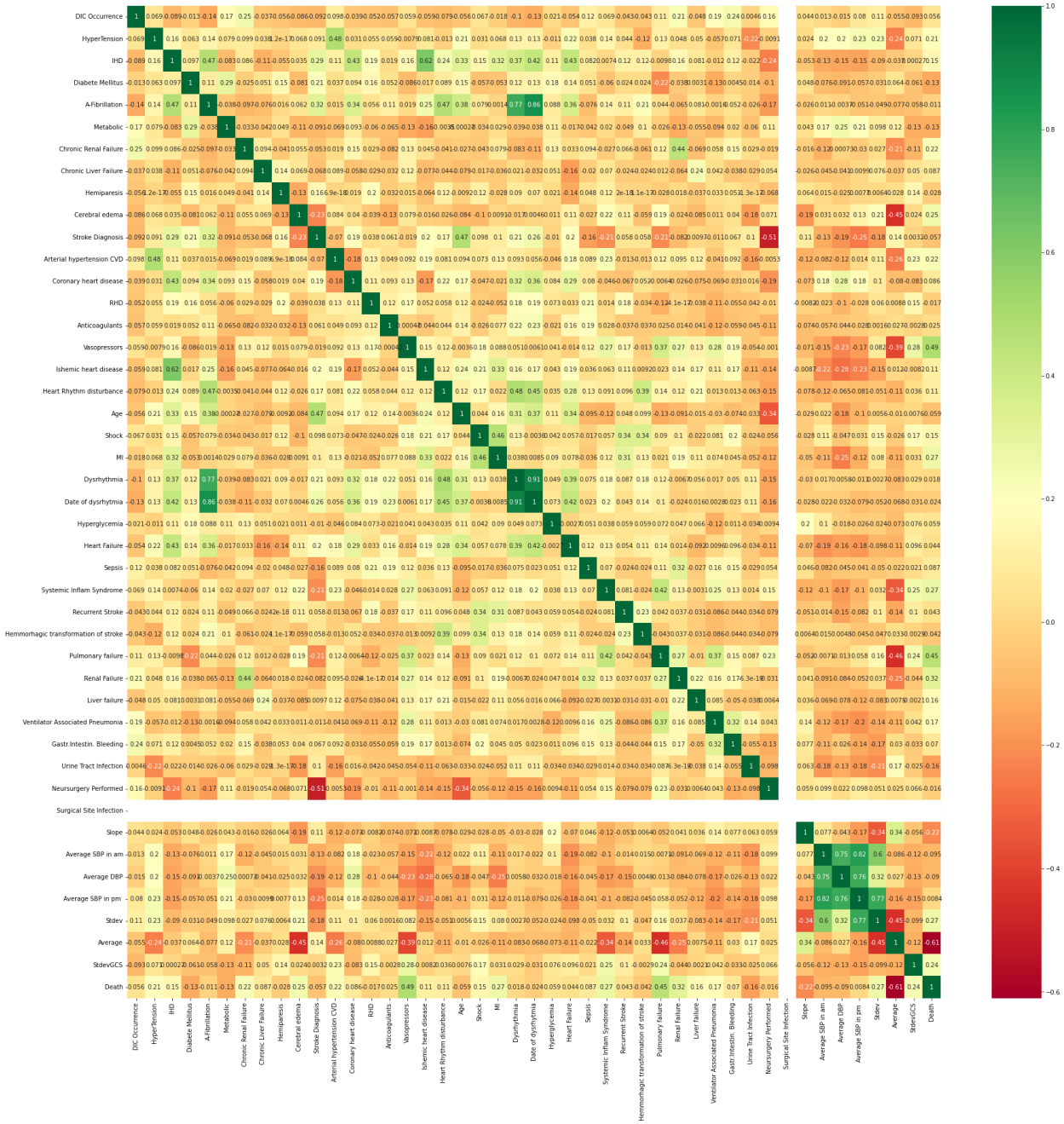
References

1. R. S. Jeena and S. Kumar, "Stroke prediction using SVM," *2016 International Conference on Control, Instrumentation, Communication and Computational Technologies (ICCICCT)*, 2016, pp. 600-602, doi: 10.1109/ICCICCT.2016.7988020.
2. C. Hung, W. Chen, P. Lai, C. Lin and C. Lee, "Comparing deep neural network and other machine learning algorithms for stroke prediction in a large-scale population-based electronic medical claims database," *2017 39th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, 2017, pp. 3110-3113, doi: 10.1109/EMBC.2017.8037515.
3. Arslan, A. K., Colak, C., & Sarihan, M. E. (2016). Different medical data mining approaches based prediction of ischemic stroke. *Computer Methods and Programs in Biomedicine*, *130*, 87–92.
4. Indarto, E. Utami and S. Raharjo, "Mortality Prediction Using Data Mining Classification Techniques in Patients With Hemorrhagic Stroke," *2020 8th International Conference on Cyber and IT Service Management (CITSM)*, 2020, pp. 1-5, doi: 10.1109/CITSM50537.2020.9268802.
5. Lecture materials
6. Auto.gluon.ai. 2022. *AutoGluon: AutoML for Text, Image, and Tabular Data — AutoGluon Documentation 0.4.0 documentation*. [online] Available at: <https://auto.gluon.ai/stable/index.html> [Accessed 6 May 2022].

Appendices

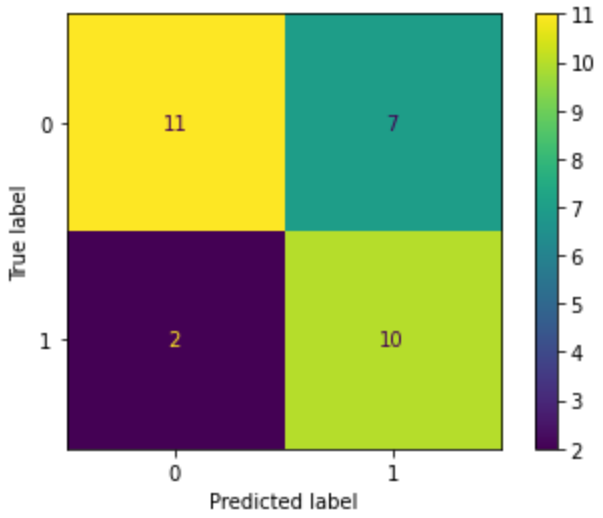
Appendix A

Correlation Matrix

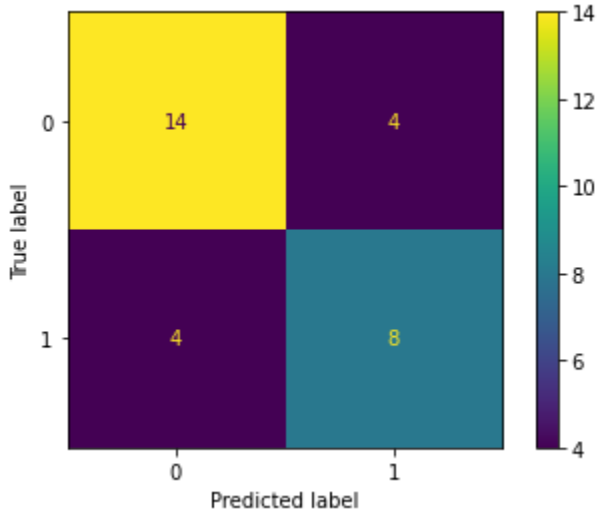


Appendix B.

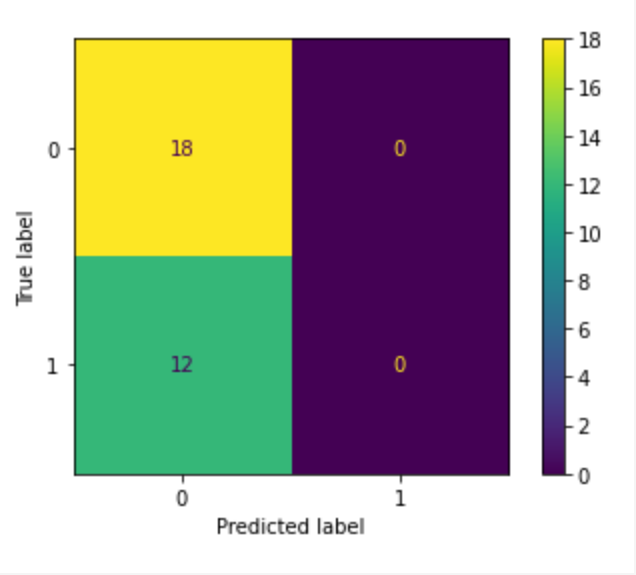
Confusion matrix for Naive Bayes without feature selection



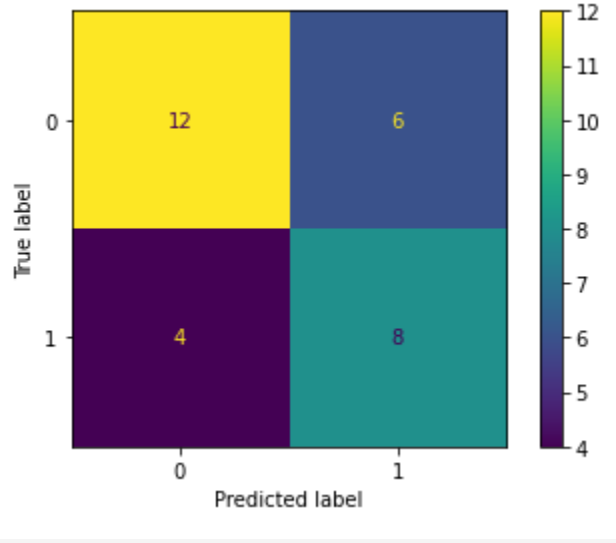
Confusion matrix for Decision trees without feature selection



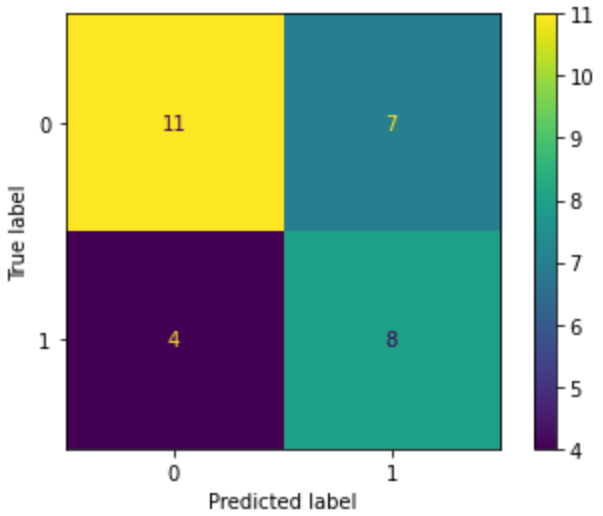
Confusion matrix for Support Vector Machine without feature selection



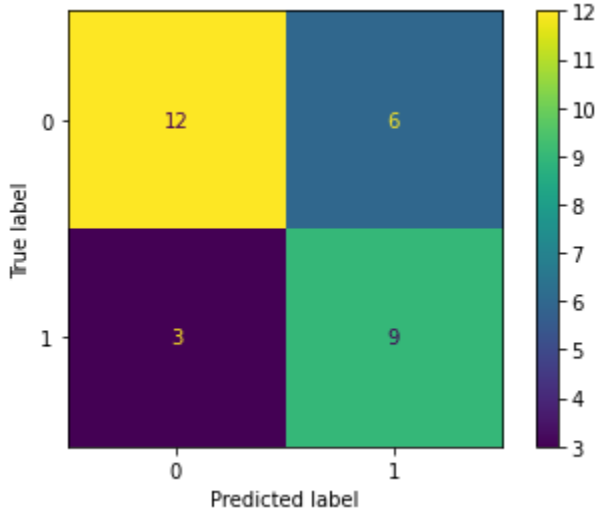
Confusion matrix for Logistic Regression without feature selection



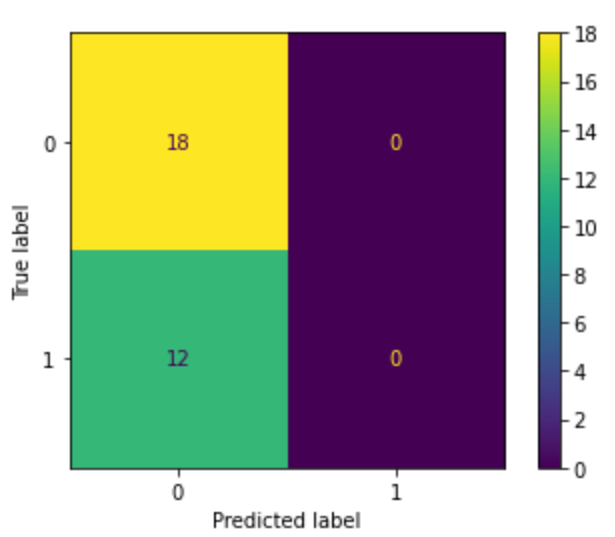
Confusion matrix for Naive Bayes after feature selection



Confusion matrix for Decision trees after feature selection



Confusion matrix for Support Vector Machine after feature selection



Confusion matrix for Logistic Regression after feature selection