Stroke prediction based on patient data

Introduction

15 million people worldwide suffer stroke each year. This causes death for 5 million and another 5 million are permanently disabled [1]. For the treatment and prevention of stroke it is important that warning signs are seen in time. If the high risk of stroke could be detected in time, it could possibly be completely prevented. In this project, I will try to predict if patient will have risk of a stroke based on patient data.

Problem Formulation

The machine learning problem is to predict if patient will have risk of a stroke based on patient data. I will test which of these three classification methods: random forest, support vector machines or k-nearest neighbour classifier, is the best for this problem and then make a final evaluation with the best method.

In this machine learning problem, data points are patients. Patients are characterized by 9 different clinical features. These features are gender, age, does the patient have hypertension or not, does patient have heart disease or not, is patient married, work type, residence type, average glucose level in blood, body mass index and smoking status. The labels where patients will be classified are stroke or no stroke. This machine learning problem is a classification task because patients are classified into classes.

Method

Description and pre-processing of the data set

Data set and more information about it can be found here [2]:
https://www.kaggle.com/fedesoriano/stroke-prediction-dataset

The original data set contains 5110 data points with 12 attributes: Unique id, gender, age, does patient have hypertension or not, does the patient have heart disease or not, is patient married, work type, residence type, average glucose level in blood, body mass index, smoking status and whether the patient has had a stroke. Some patients have no information about theirs body mass index, so they are deleted from the data set. After this, data set contains 4909 data points which will be used for this classification task.

The last column which contains information, whether the patient has had a stroke is selected to be label vector. The values of the label vector are ‘No’ if the patient does not has had a stroke and ‘Yes’ if the patient has had a stroke. All other columns expect unique id are selected to be features of the patients. Columns gender, is patient married, work type, residence type, smoking status has string values, so they are changed numerical in the following way: gender

(‘Male’ =0, ‘Female’ = 1, ‘Other’ = 2), is patient married(‘Yes’ =1, ‘No’ = 0), work type(‘Private’ = 0, ‘Self-employed’ = 1, ‘children’ = 2, ‘Govt_job’ = 3, ‘Never_worked’ = 4), residence type(‘Urban’ = 0, ‘Rural’ =1) and smoking status (‘never smoked’ = 0, ‘unknown’ = 1, ‘formerly smoked’ = 2, ‘smokes’ = 3).

Now data set contains 4909 patients with feature vector and label. 4700 patients have a label ‘No’ and 209 patients have a label ‘Yes’. To make data set better this needs to be balanced. A method SMOTE from python imblearn library is used to do this. Synthetic Minority Oversampling Technique (SMOTE) is widely used approach to oversample the minority class, so that the oversampled data set does not contain only replicas of the minority class but synthesizes new examples of it [3]. In SMOTE, this is done by k-nearest neighbour algorithm [3]. More information about SMOTE can be found here : https://machinelearningmastery.com/smote-oversampling-for-imbalanced-classification/. After SMOTE, there are 9400 data points with 4700 patients with label ‘No’ and 4700 patients with label ‘Yes’.

With using sklearn.model_selection.train_test_split data set is divided to train and test set, so that test set contains 20% of the data set [4]. Training set will be used to evaluate which of the three classifiers is best for this problem and the test set is used in the final evaluation. Features are also standardized before training the models with sklearn.preprocessing.StandardScaler.

Classification methods and evaluation

I will test which of these methods: random forest, support vector machines or k-nearest neighbour classifier, is best for this problem. All models are evaluated with 5-fold cross-validation and model which produces smallest average validation error (error rate), will be selected for final evaluation. 5-fold cross validation is made with sklearn library’s StratifiedKFold method [4]. Model that gives the lowest error rate is selected to the final estimation. In final estimation, model’s accuracy and confusion matrix is computed with the unseen test data.

A short description of the classification methods used in this project:

Random forest

Random forest is a classification method that uses multiple decision tree classifiers to produce accurate classifications. In random forest, multiple decision trees are trained with randomly selected features and data sets, so that it produces many different decision tree classifiers. Final classification in random forest is made with majority vote so that every decision tree classification is vote and class with most votes is the result of the classification. I use sklearn library’s RandomForestClassifier with 100 estimators in this project to make random forest classification [4]. I chose random forest to this project because it is very effective algorithm for classification, and I think that it will produce accurate classifications for this problem.

Support Vector Machines

Support Vector Machines (SVM) are supervised learning models which can be used for classification, regression and outlier’s detection. I use sklearn library’s SVC with its default values in this project to make SVM classification.[4] I chose SVM model to this project because it is effective in high dimensional space and I believe that it would produce good predictions for this problem.

k-Nearest Neighbours

K-nearest neighbour classifier makes classification by computing majority vote of the nearest neighbours of every point. The class that has the most representatives within the nearest neighbours of the point is the result of a classification. I use sklearn library's KNeighborsClassifier with 5 neighbours to perform KNN classification.[4] I chose KNN classification to this project because I wanted to see how this simple algorithm performs in this task.

Results

Table of the results of each model after 5-fold cross-validation:

	Random forest	SVM	KNN
Validation error	0.027	0.091	0.074
Training error	0.000	0.078	0.049

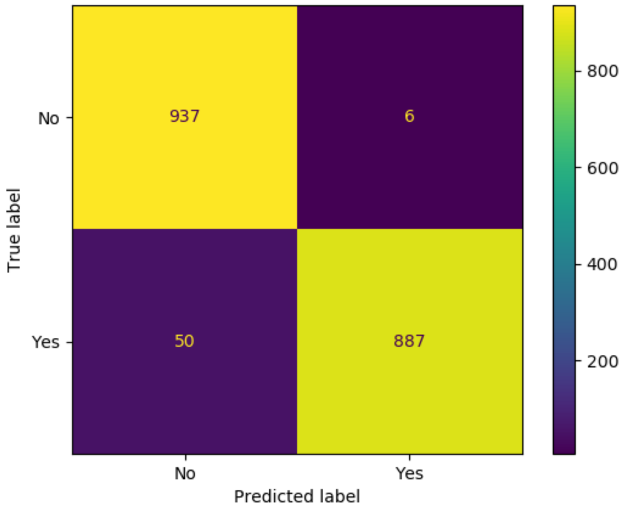
According to the results of the 5-fold cross-validation random forest has lowest error rate (0.027), so it is chosen to the final estimation.

Final estimation:

Accuracy for the training data = 1.00

Accuracy for the test data = 0.970

Confusion matrix for the test data:



Accuracy of the random forest for the test data is 0.970 which is very good and says that random forest gives very good prediction whether a patient will have the risk of a stroke or not. According to the confusion matrix there is only 56 misclassifications of the 1880 examples which is a very good result.

One way how the accuracy of the classifiers could have got better is that a grid search had been performed for every model to find optimal parameters, for example optimal number of estimators for random forest, optimal gamma and C value for SVC, and optimal number of neighbours for KNN classifier.

Conclusion

After testing how three different classification methods perform for predicting whether patient will have risk of a stroke or not, has come to the conclusion that random forest is best of these models for this machine learning problem. Random forest was picked to be the best model because it produced smallest average error rate for the test sets in 5-fold cross-validation. After this, final evaluation where model was used to predict labels of the unseen data was made for the random forest. Random forest produced accuracy of `0.970` which is a very good result of the classification.

Although the result of the classification is good, I think that there is still room for improvement. I think that if the optimal parameters for the models would be searched with grid search, better results could be obtained. Differences between error rates of the classifiers was relatively small, so optimal parameters could have shown that KNN or SVM would perform better for this problem.

All in all, we can say that whether the patient will have risk of a stroke or not can be predicted from patient data which contains following features: gender, age, does patient have hypertension or not, does patient have heart disease or not, is patient married, work type, residence type, average glucose level in blood, body mass index and smoking status. Best classifier of the tested classifiers for this problem is the random forest with accuracy of `0.970`. For future study, to get better accuracy, could be tried to find optimized parameters for the models.

References

[1] “Stroke statistics”. The internet stroke center. [Cited 23 March 2021] Available at: http://www.strokecenter.org/patients/about-stroke/stroke-statistics/

[2] fedesoriano. “Stroke Prediction Dataset”. Kaggle. 2021. [Cited 23 March 2021]. Available at: https://www.kaggle.com/fedesoriano/stroke-prediction-dataset

[3] Brownlee, J. “SMOTE for Imbalanced Classification with Python”. Machine Learning Mastery. 2021. [Cited 23 March 2021]. Available at: https://machinelearningmastery.com/smote-oversampling-for-imbalanced-classification/

[4] F. Pedregosa, et.al. “Scikit-learn: Machine Learning in Python“. Journal of Machine Learning Research, vol. 12, p. 2825--2830, 2011.