2023-01-04 Rina Tse, A21

# Project: Develop an end-to-end Machine Learning Pipeline

## Use Case:

This project aims to identify individual and possible environmental contributors to the occurrence of strokes and to use those features to predict the occurrence of strokes.

## Data Analysis: Data Processing:

The dataset used in the project was provided by DSTI. Data cleaning entailed dropping the unnecessary patient ID column. This project aims to identify characteristics of patients that contribute to whether they have had a stroke to which the patient ID has no bearing.

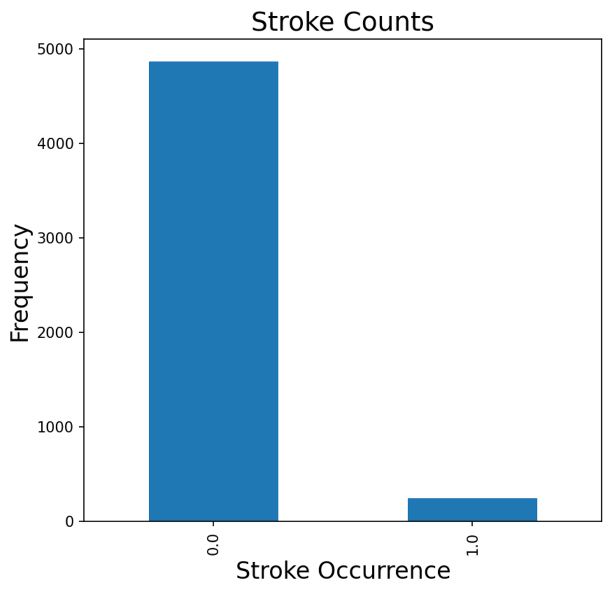
I used the function whitespace_remover() to remove any possible extraneous empty spaces around values. I removed NaN values by replacing them with an integer that's larger than any other value in the dataset, then replaced those values with np.nan. I then used KNNImputer to predict the missing values. This filled missing values with a predicted value while retaining values of 0, which do not equate to an 'empty value' in this case.

I used One pd.get_dummies() to one hot encode categorical labels, which converts string values to a numerical representation while avoiding assumptions about hierarchy. For example, the label 'dog' might be encoded to 1 while the label 'cat' is encoded to 2. One Hot Encoder avoids the assumptions that 2 in this case is greater than 1.

## Data Analysis: Exploratory Analysis

During exploratory analysis, I found that the dataset contained numerical and categorical values, which I encoded.

Classes in the target column consisted of the rates of a binary count of 1, stroke and 0, no stroke. These classes were severely unbalanced:



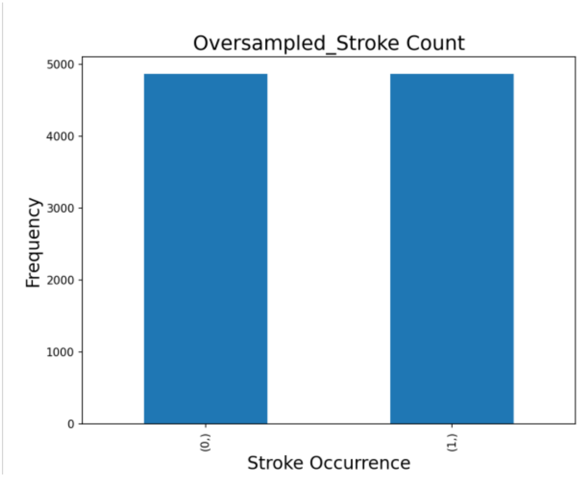
```
Value Counts:

0: 4861

1: 249
```

Continuing analysis with an imbalanced dataset would result in an inaccurate model that predicts the most prevalent class. In this case, simply predicting 0 would have resulted in an accuracy of 95%.

To counter this imbalance, I used Random Oversample to randomly generate more instances of the minority classes until the ratio of classes became more even. I used oversampling rather than under sampling because undersampling would have resulted in a significant loss of data.



## Data Analysis: Feature Engineering and Pruning:

I used a support vector machine with recursive feature elimination with cross validation to select features that most significantly pertain to my target variable. SVM finds a hyperplane within N-dimensional space to classify data points, using support vectors. Support vectors are datapoints nearest to the hyperplane, which influence how the hyperplane is positioned. SVM then finds the coefficients of each attribute and ranks the coefficients according to their size. Recursive feature elimination runs through this process, removes the lowest ranking coefficient then repeats until all attributes have been eliminated.

To validate these features, I used repeated k-fold cross-validation. Repeated k-fold cross validation repeats the process of cross validation where the dataset is split into k number of groups, one is used as a test set and the rest are used as training. The process is repeated until each group is used as a test set then the mean result is reported. The features selected were:

- Age
- avg_glucose_level
- bmi

## Data Analysis: Justification of Feature Engineering Methods

I used a support vector machine for this dataset because it can be applied to linear and nonlinear problems and it is effective in high dimensional space, which was a requirement of this dataset[1].

I used recursive feature selection with cross validation and cross validation to validate the feature selection because it allows the model to make the most of the data available without reserving a portion of the dataset for testing. Cross validation also allows the model to train on multiple train-test splits, which provides a better prediction as to how the model will respond to unseen data. The advantage of this method

[1] Witten, I., Frank, E., Hall, M. and Pal, C., n.d. Data Mining, 4th Edition. p.Chapter 10.

is that the mean result of multiple repeats will likely be more accurate than a single run. I used RFECV that adds a cross validation step to automatically select the number of features rather than setting that number arbitrarily[2].

The accuracy in this model for feature selection was 96.2%.

## Model Training: Comparison of Different Models and Model Selection

I used the following models for binary classification, with features selected during the previous step

| Model | Accuracy |
|---|---|
| NeighborhoodComponentsAnalysis () KNeighborsClassifier() | 89% |
| LogisticRegression() | 89% |
| Svm() | 89% |
| GaussianNB() | 89% |
| RandomForestClassifier() | 89% |
| DecisionTreeClassifier() | 89% |

These models performed equally within two significant digits. It is possible that the feature selection step contributed more to the accuracy of the model than the type of classifier used for prediction.

## Model Training: Evaluation Metric and Results Interpretation.

The accuracy score using REFECV selected features, and 6 models appropriate for binary classification was 89%. I used an accuracy score because the dataset was balanced by oversampling[3].

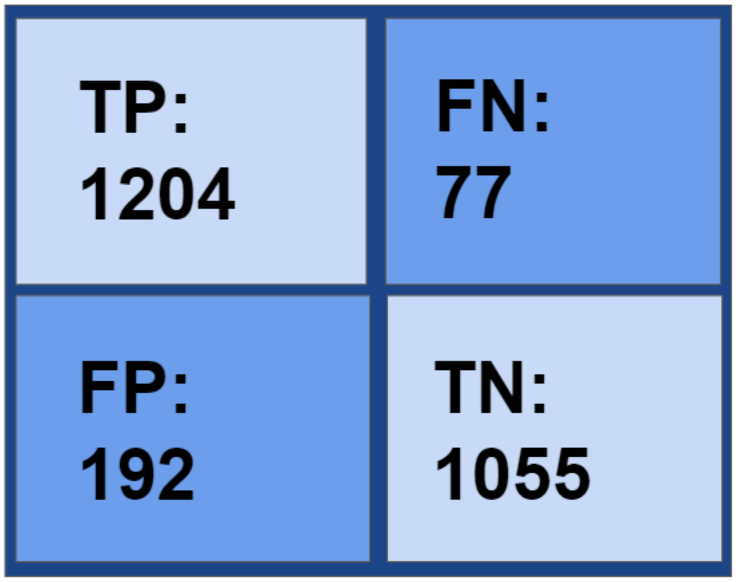
According to the confusion matrix produced by NeighborhoodComponentsAnalysis and KNeighborsClassifier the model predicted:

- 1204/1282 (93%) correct predictions for a stroke occurring
- 1055/1247 (84%) correct predictions for a stroke not occurring

[2] *Sklearn.feature_selection.RFECV*. scikit. (n.d.). Retrieved January 6, 2023, from https://scikit-learn.org/stable/modules/generated/sklearn.feature_selection.RFECV.html

[3] Korstanje, J. (2021, August 31). *The F1 score*. Medium. Retrieved January 4, 2023, from https://towardsdatascience.com/the f1-score-bec2bbc38aa6

Confusion Matrix for Stroke Prediction



| | |
|---|---|
| TP: 1204 | FN: 77 |
| FP: 192 | TN: 1055 |

- Average expected loss: 0.121
- Average bias: 0.085
- Average variance: 0.036

The average expected loss represents the average of absolute differences between the predicted and actual value. This model uses Mean Squared Error to calculate loss[4]:

$$\text{MSE} = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2$$

Average bias is the difference between the average prediction of the model and the true value the model is trying to predict [5]. This measurement concerns overfitting/ underfitting where a high bias indicates that the model is oversimplified and does not respect the training data.

Average variance concerns the model's ability to accurately classify unknown data. Models with high variance respect the training data but are not able to generalize in order to respond accurately to unknown data.

[4] Seif, G. (2022, February 11). *Understanding the 3 most common loss functions for machine learning regression*. Medium. Retrieved January 4, 2023, from https://towardsdatascience.com/understanding-the-3-most-common-loss-functions-for-machine-learning-regression-23e0ef3e14d3

[5] Singh, S. (2018, October 9). *Understanding the bias-variance tradeoff*. Medium. Retrieved January 4, 2023, from https://towardsdatascience.com/understanding-the-bias-variance-tradeoff-165e6942b229

This model had low loss, bias and variance with a high accuracy score. Accuracy was lower for when predicting the occurrence of a stroke not occurring despite that being originally the class with the most instances. The model brought down the accuracy from simply predicting 0 (no stroke), which was originally 95%. I think this is preferable in that oversampling prevented overfitting caused by imbalance in the dataset.

## Conclusion

This project identified features that most pertinently contribute to patient's risk of stroke. These features were age, blood glucose levels and BMI, which is consistent with medical literature[6]. The model was able to use these features to predict the occurrence of a stroke with 89% accuracy (93% accurate for TP occurrence and 84% accurate for TN occurrence).

Source code can be found on github: https://github.com/Rtse716/Predict_strokes/edit/main/README.md

[6] Association between Body Mass Index and Stroke Risk Among Patients with Type 2 Diabetes. J. Clin. Endocrinol. Metab. 2020;105:96–105. doi: 10.1210/clinem/dgz032. - DOI - PMC - PubMed