Project Report

Team Members: Garabed, Ko

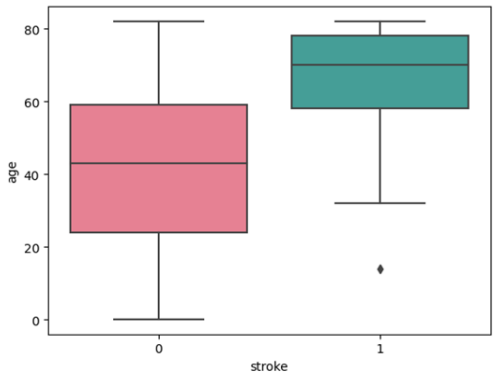
Introduction:

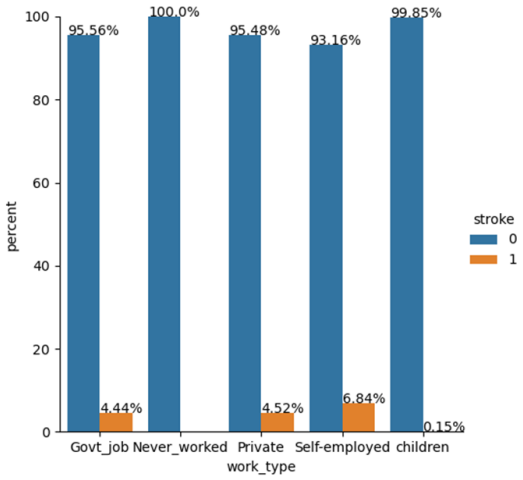
Stroke is one of the leading causes of death and disability worldwide. The objective of this project is to predict the possibility of stroke in patients based on various factors such as age, gender, hypertension, heart disease, etc. The dataset used for this project was obtained from Kaggle, which contains 5110 samples with 11 features.

Data preparation:

Data preparation involved removing the ID column and filling in missing values with the mean value. Encoding was also performed to convert text data to numerical data, and normalization was carried out to scale the features into the range [0,1] to transform features to be on a similar scale. Data visualization was also performed to get insights into the data and understand the relationships between different features.

```
In [64]: ax = sns.boxplot(x="stroke", y="age", data=df, orient='v',  palette='husl')
```



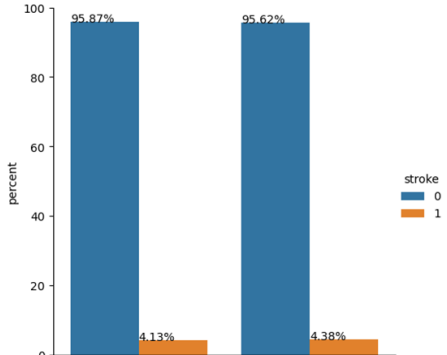



```
In [16]: x,y = 'Residence_type', 'stroke'

df1 = df.groupby(x)[y].value_counts(normalize=True)
df1 = df1.mul(100)
df1 = df1.rename('percent').reset_index()

g = sns.catplot(x=x,y='percent',hue=y,kind='bar',data=df1)
g.ax.set_ylim(0,100)

for p in g.ax.patches:
    txt = str(p.get_height().round(2)) + '%'
    txt_x = p.get_x()
    txt_y = p.get_height()
    g.ax.text(txt_x,txt_y,txt)
```



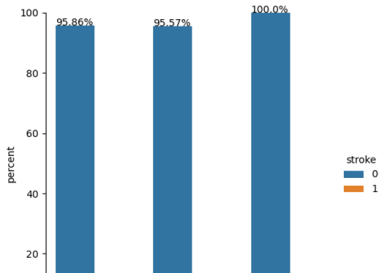

```
In [59]: x,y = 'gender', 'stroke'

df1 = df.groupby(x)[y].value_counts(normalize=True)
df1 = df1.mul(100)
df1 = df1.rename('percent').reset_index()

g = sns.catplot(x=x,y='percent',hue=y,kind='bar',data=df1)
g.ax.set_ylim(0,100)

for p in g.ax.patches:
    txt = str(p.get_height().round(2)) + '%'
    txt_x = p.get_x()
    txt_y = p.get_height()
    g.ax.text(txt_x,txt_y,txt)
```

```
posx and posy should be finite values
posx and posy should be finite values
```

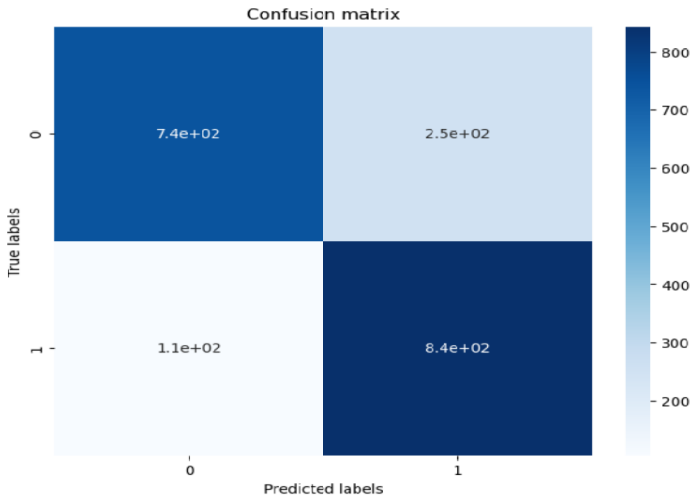


Methods:

SVM Classifier: To select the important features, a decision tree was used with the criterion set to gini impurity. The decision tree helps identify the most relevant features for the prediction model. The SVM algorithm was then used to train the data with the Kernel set to 'linear' since the data is not very complicated. The hyperparameter C was set to 1 to create a wider margin. This helps improve the model's accuracy and generalization. To address the issue of overfitting, three oversampling methods, namely RandomOverSampler, SMOTE, and ADASYN, were used. These techniques help balance the class distribution and reduce the impact of over-represented samples. The use of these oversampling methods can help improve the model's performance and accuracy.

Random Forest Classifier: Random forest was used for both feature selection, and for creating the binary classifier model to predict if a person will have a stroke. For feature selection, random forests uses information gain, or gini index, to calculate to importance of each feature for each fold. Moreover, it calculates the average of each feature score across all folds, and returns features with the highest scores. For building the model, gridSearchSV was used to find the best hyperparameters for the random forest classifier. However there was overfitting, the percision score for no stroke was very high and for yes stroke was very low. The reason for this is because the instances that are classified as yes had a very small percentage. Therefore the model was always predicting no. to Solve this problem an oversampling method was used. SMOTE, chooses the target with the lower percentage of instances and oversamples those instances by finding relations between the features, until our dataset is balanced. Here are the results for this method.

```
{'max_depth': 20, 'max_leaf_nodes': 20, 'n_estimators': 500}
Accuracy score: 81.53%
```



```
Cross-validation scores: [0.80475504 0.82925072 0.82636888 0.85086455 0.84942363 0.84726225
 0.84293948]
```

1]:

```
print(classification_report(y_test, y_pred))
```

```
              precision    recall  f1-score   support

           0       0.88      0.75      0.81       995
           1       0.77      0.89      0.82       949

    accuracy                           0.82      1944
   macro avg       0.82      0.82      0.81      1944
weighted avg       0.82      0.82      0.81      1944
```

Evaluation:

Due to the data imbalance, the model had issues with overfitting to the majority class, 'no_stroke,' which led to poor performance on the minority class, 'yes_stroke.' Without oversampling, the precision, recall, and F1 score for 'yes_stroke' were zero, and the true negative was also zero, even though the accuracy was around 95%. To address this issue, oversampling techniques such as RandomOverSampler, SMOTE, and ADASYN were used. After applying these techniques, the model's performance improved significantly, and all metrics became normal. The use of oversampling methods helped balance the class distribution and reduced the impact of over-represented samples, resulting in a more accurate and reliable model. Among the three oversampling methods used, using SMOTE led to a considerable improvement in the performance. Moreover, Smoking_status feature had alot of Unknown values, approximately 30% of the dataset. Logistic Regression was used to predict the Unknown values, but got very low accuracy. However, tuning the hypermarameters and more feature engineering could lead to a better accuracy. Moreover, using the frequently appeared value gave relatively good results.

```
clf = LogisticRegression(multi_class='multinomial',**best_hyperparams)
clf.fit(X_train,y_train)
y_predict = clf.predict(X_test)
y_predict_train = clf.predict(X_train)
accuracy = accuracy_score(y_train, y_predict_train)
print(accuracy)
accuracy1 = accuracy_score(y_test, y_predict)
print(accuracy1)
```

```
0.4413838695460555
0.47841409691629955
```

Conclusion:

In conclusion, this project has shown that the use of SMOTE oversampling can significantly improve the accuracy of the stroke prediction model. However, there is still room for improvement. To further improve the model's performance, we could try using different values of C for the SVM algorithm to change the margin, and we could also experiment with different kernel options such as ‘poly’, ‘rbf’, ‘sigmoid’, and ‘precomputed’. For the decision tree feature selection, we could try “log_loss” and “entropy” for the criterion instead of using gini impurity. These modifications could potentially improve the model's performance and increase its accuracy.Additionally, the same goes for random forest classifier.tuning the hyperparameters could lead to better results Overall, this project has shown the potential of using machine learning techniques to predict strokes and improve patient outcomes.

Related Works (Ko)

There are many related works in the stroke prediction domain, however, two previous works have a significant impact on the research. One of the paper ("Predictive Analytics For Stroke Disease," 2019) focuses on using predictive analytics to improve the diagnosis and treatment of stroke disease. The researchers propose the use of an artificial neural network (ANN) algorithm to build a predictive model for stroke disease, which achieved an accuracy of 95.15%. The ANN algorithm can learn complex patterns and relationships in data, making it a useful tool for healthcare providers.

An alternative approach could be to use an ensemble model of different machine learning algorithms, like decision trees and support vector machines, to improve accuracy and robustness. Incorporating other relevant data sources, such as genetic information or environmental factors, could also improve the accuracy of the model.

The other article ("Using Machine Learning Method to Predict Stroke Risk" 2021) describes the development of a machine learning prototype that predicts the occurrence of stroke based on observed characteristics and symptoms. By using feature selection methods, the study identified the best features of stroke disease, and various machine learning algorithms were compared. The Decision Tree classifier had a good accuracy of 98% when applied with feature selection techniques. The study provides a promising approach for early identification, medication, and therapy for stroke.

The paper discusses the use of machine learning models for stroke prediction, comparing logistic regression, K-Nearest Neighbor, and random forest models. The study used real-life data of 5110 patients, with 268 experiencing a stroke, and identified physical characteristics and lifestyle indicators as predictors. The random forest model outperformed the other two models, and the

study concludes that simple machine learning models, particularly the random forest, can perform well in stroke prediction and are easier to implement compared to integrated models. The paper recommends future studies to explore these possibilities further. The study also addresses the issues of missing and imbalanced data, using mean imputation and z-score normalization. A better approach could be to use more advanced techniques like deep learning algorithms to improve the prediction accuracy and to handle the missing data more efficiently. Additionally, it would be beneficial to have a larger dataset with more diverse features to capture more information about stroke risk factors.

Related works(Garabed)

In the article, Applying Machine Learning Techniques for Stroke Prediction in Patients, the author experiments with four types of feature engineering. Principal Component Analysis (PCA), Chi square test (chi), PCA with chi, and removing outliers with PCA and chi. Moreover, it uses 3 different types of classification, Decision Trees, Support Vector Machines and Naive Bayes. I am familiar with the classification algorithms, used in this article from the lectures. Decision tree would be my choice for this problem. For feature selection, the author used PCA, and chi. PCA is used for dimension reduction. To find PCA, the algorithm finds the line that best fits the data set, it does so by drawing random lines and calculate the distance between the origin and the point where the data is projected on the line. It does this for every single point and calculates the sum of the squared of all distances-(which is also called the eigen value, and it chooses the line with the largest distance value. Next, it will calculate the eigen vector, and eigen value. Finally, it will use the eigen value to get the variance for each PCA, and uses the PCA's with the largest values to represent the data. Chi square test, uses the observed values between features. And the expected table value; which can be obtained by multiplying the totals of each feature divided by the sum of totals. Then it calculates X1 which is the sum of the squared of observed value minus expected value, divided by the expected value. It also calculates X tabular, by finding degree of freedom and using the tabular data to get the values. Then it check if X1 is bigger than X tabular, if it is, then that means those features have significant relation.

In the article Analysis and Prediction of Stroke using Machine Learning Algorithms, The author also does comparison. It uses J48, decision tree algorithm, that uses top-down approach, Bayes Network Classifier, Naïve Bayes Classifier, AdaBoost; which is an iterative ensemble, meaning it combines multiple classifiers to make a stronger one. In this paper J48 and AdaBoost gave the best results. However, during feature engineering the author removed all the instances that had unknow value for the smoking status feature which counts for 30.7 percent of the data set. This is a lot of tuples, and I would not take this approach. It is better to leave it as is, or use a classifier to predict it's value. Moreover, I would remove the instance that has the value other in the gender feature.

Reference

A. F. Nur Masruriyah, T. Djatna, M. K. Dewi Hardhienata, H. H. Handayani and D. Wahiddin, "Predictive Analytics For Stroke Disease," 2019 Fourth International Conference on Informatics and Computing (ICIC), Semarang, Indonesia, 2019, pp. 1-4, doi: 10.1109/ICIC47613.2019.8985716.

Y. Miao, "Using Machine Learning Method to Predict Stroke Risk," 2021 2nd International Conference on Information Science and Education (ICISE-IE), Chongqing, China, 2021, pp. 665-668, doi: 10.1109/ICISE-IE53922.2021.00156.

V. JalajaJayalakshmi, V. Geetha and M. M. Ijaz, "Analysis and Prediction of Stroke using Machine Learning Algorithms," 2021 International Conference on Advancements in Electrical, Electronics, Communication, Computing and Automation (ICAECA), Coimbatore, India, 2021, pp. 1-5, doi: 10.1109/ICAECA52838.2021.9675545.

R. K. Kavitha, W. Jaisingh and S. R. Sujithra, "Applying Machine Learning Techniques for Stroke Prediction in Patients," 2021 International Conference on Advancements in Electrical, Electronics, Communication, Computing and Automation (ICAECA), Coimbatore, India, 2021, pp. 1-4, doi: 10.1109/ICAECA52838.2021.9675652.