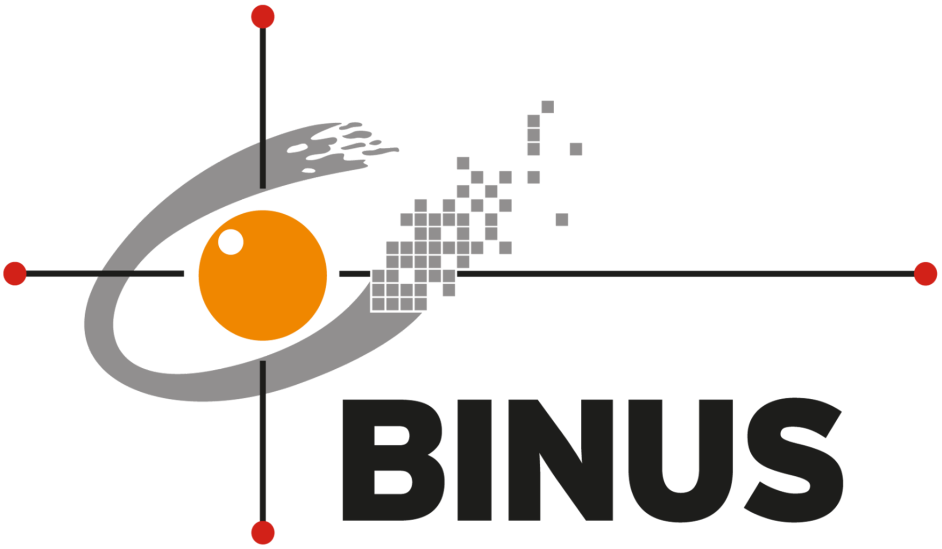


STROKE PREDICTION REPORT

COMP6577 Machine Learning LC01

2301855154 Axel Lie
2301937173 Dean Ananda Ramadhan
2301920941 Kelly Cornelya Kesuma
2301853136 Reinaldy Sukamto

Bina Nusantara University
Jakarta, Indonesia

I. Introduction

Menurut WHO, Stroke adalah penyakit yang dapat menyebabkan kematian terbesar kedua di dunia. Stroke dapat dideteksi dengan beberapa fitur-fitur tertentu seperti jenis kelamin, umur, ataupun penyakit seperti hipertensi, penyakit jantung, dan penyakit lainnya yang dapat menyebabkan terjadinya stroke serta fitur-fitur lainnya. Oleh karena itu, kami mencoba menggunakan Machine Learning dengan library tensorflow untuk melakukan klasifikasi apakah seseorang terindikasi stroke atau tidak dengan beberapa fitur yang menjadi parameter dalam project kami.

I.I Purpose

Membuat model yang dapat melakukan prediksi apakah seseorang mengidap penyakit stroke atau tidak.

I.II Benefit

➔ User dapat mencegah kematian yang diakibatkan oleh stroke dengan mengetahui lebih dini apakah dirinya mengidap stroke atau tidak.

➔ User dapat melakukan pengecekan secara mandiri (tanpa bantuan ahli / dokter) apakah dirinya dinyatakan mengidap stroke atau tidak oleh system.

➔ User dapat menerima hasil pengecekan secara cepat.

II. Literature Review

Dari eksperimen yang dilakukan pada pertemuan sebelumnya, permasalahan yang dihadapi adalah imbalanced dataset dimana data non-stroke lebih banyak dari stroke. Dari jurnal [1], salah satu metode yang digunakan untuk mengatasi masalah imbalanced dataset adalah metode oversampling dengan menggunakan SMOTE. Hal ini menjadi fokus utama kelompok kami, dikarenakan dataset yang kami gunakan memiliki masalah berupa imbalanced dataset.

Berdasarkan jurnal [2], SMOTE merupakan resampling jenis over-sampling dengan strategi cluster-based untuk mengatasi imbalanced data. SMOTE akan membuat data sintesis dari minority class berdasarkan feature space dari data tersebut. Dengan menggunakan K Nearest Neighbor, SMOTE dapat menentukan cluster yang didapatkan dan membuat data di cluster-cluster tersebut.

SMOTE bekerja dengan memilih contoh titik yang dekat di feature space, lalu menggambar garis di antara titik di feature space dan menggambar sampel baru pada titik di sepanjang garis tersebut.

Untuk pemilihan model yang akan kami pakai, mengacu pada jurnal [3] & [4], dikatakan bahwa terdapat beberapa method yang umum digunakan dalam melakukan Binary Classification. Dari beberapa pilihan metode yang umum digunakan dalam melakukan binary classification, kami memilih beberapa metode yang telah kami pahami, diantaranya adalah Support Vector Machine (SVM), Feedforward Neural Network (FNN), Decision Tree, dan juga Logistic Regression.

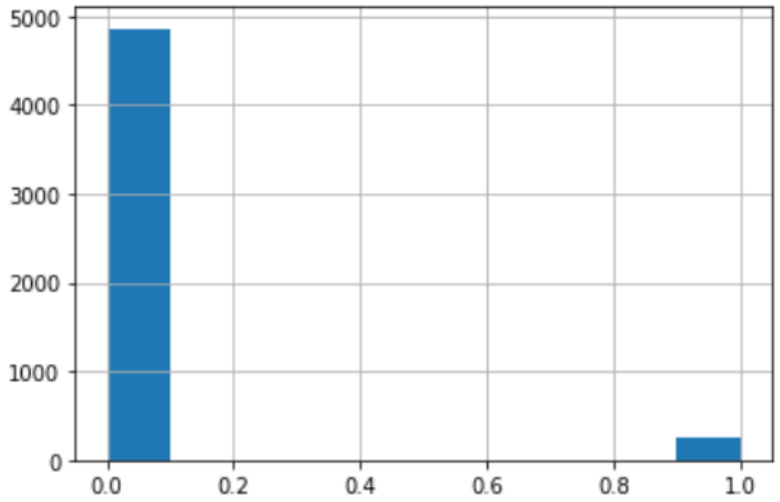
III. Experiments & Methods

Dataset yang kami gunakan adalah *stroke prediction dataset* [5]. Dataset ini bersumber dari sumber confidential dan hanya dapat digunakan untuk tujuan riset semata.

Pertama-tama, kelompok kami melakukan eksplorasi terhadap dataset yang kami gunakan. Dataset kami memiliki 10 fitur berupa gender, age, hypertension, heart_disease, ever_married, work_type, Residence_type, avg_glucose_level, bmi, smoking_status dan label berupa stroke. Lalu, kami melihat korelasi tiap-tiap fitur terhadap label dan menemukan bahwa fitur bmi memiliki nilai korelasi yang sangat kecil sehingga kami memutuskan untuk tidak memakai fitur tersebut.

```
stroke              1.000000
age                 0.245257
heart_disease       0.134914
avg_glucose_level   0.131945
hypertension        0.127904
bmi                 0.042374
```

Setelah itu, kami melakukan plotting dan melihat jumlah value dari masing-masing fitur dan label pada dataset kami. Kami menemukan bahwa label yang ada pada dataset kami, merupakan imbalance class dengan perbandingan antara false-value dengan true-value sebesar 19,5 : 1.

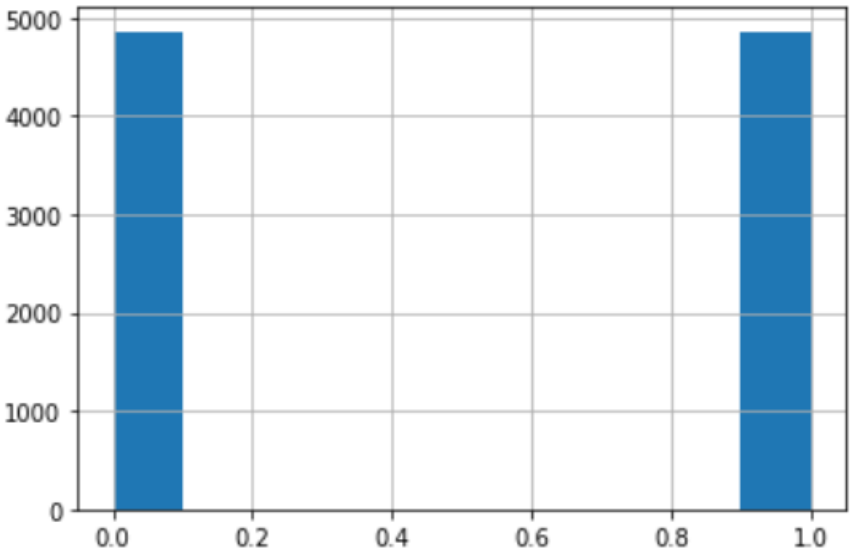


Oleh karena permasalahan imbalanced pada label kami, kami mencoba untuk melakukan random resampling terhadap label untuk meningkatkan jumlah minority value.

Namun, cara ini tidak sepenuhnya berhasil, sebab model yang dihasilkan menjadi overfitting.

Setelah melakukan percobaan dan membaca literature lebih lanjut, kami menemukan bahwa SMOTE dapat mengatasi permasalahan tersebut dengan membuat beberapa fitur dan label sintetik dengan menggunakan metode KNN dan menemukan bahwa cara ini dapat mekukan oversampling terhadap label kami dan menghasilkan model yang tidak overfitting dengan akurasi yang cukup memuaskan.

Kami menggunakan *imblearn.over_sampling.SMOTE* dari library imbalanced-learn untuk mengimplementasikan SMOTE. SMOTE akan mengambil features yang telah di-encode menggunakan *sklearn.preprocessing.OrdinalEncoder* dari library scikit-learn untuk membuat data-data sintetis tersebut. Kami menggunakan K-Value sebesar 5, dan sampling_strategy adalah auto.



Tahap akhir pada proses preprocessing kami adalah splitting dataset. Kami melakukan scaling dan splitting dataset. Kami menggunakan bantuan *preprocessing.StandardScaler* dari library scikit learn untuk melakukan scaling terhadap fitur yang kami gunakan. Kami menggunakan bantuan *model_selection.train_test_split* dari library yang sama untuk melakukan splitting dataset menjadi train dan test dataset dengan rasio sebesar 8:2 dimana 80% train dataset dan 20% test dataset.

Setelah tahap preprocessing selesai, kami melanjutkan ke tahap selanjutnya yaitu tahap training. Kami menggunakan empat metode untuk mencari metode apa yang paling cocok untuk digunakan pada dataset kami. Metode yang kami gunakan untuk melakukan binary classification tersebut adalah Support Vector Machine (SVM), Feedforward Neural Network (FNN), Decision Tree, dan juga Logistic Regression. Berikut penjelasan lebih lanjut mengenai model-model yang kami gunakan.

III.I Support Vector Machine (SVM)

SVM digunakan untuk mencari hyperplane terbaik dengan memaksimalkan jarak antar kelas. Hyperplane adalah sebuah fungsi yang dapat digunakan untuk pemisah antar kelas. Dalam 2-D fungsi yang digunakan untuk klasifikasi antar kelas disebut sebagai line whereas, fungsi yang digunakan untuk klasifikasi antas kelas dalam 3-D disebut plane similarly, sedangkan fungsi yang digunakan untuk klasifikasi di dalam ruang kelas dimensi yang lebih tinggi disebut hyperplane.

Persamaan Support Vector Machine:

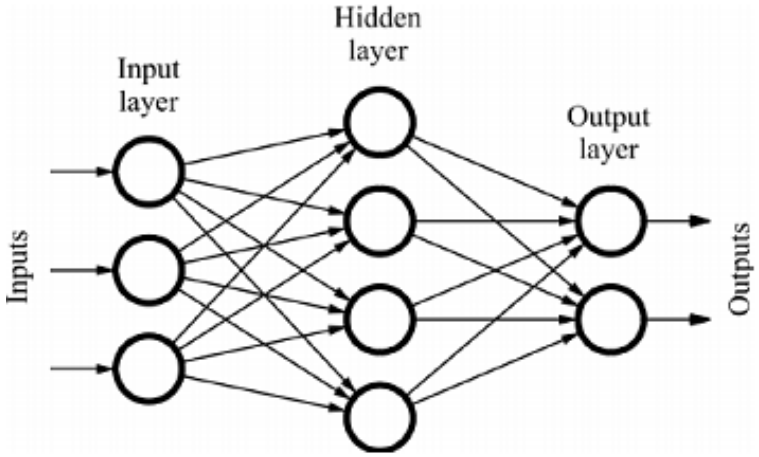
$$f(x) = w.\ x + b \text{ (4) atau } f(x) = \sum aiyiK(x, xi\ ) + b\ m\ i{=}1$$

Keterangan :

w : parameter hyperplane yang dicari (garis yang tegak lurus antara garis hyperplane dan titik support vector) *x* : titik data masukan Support Vector Machine *ai* : nilai bobot setiap titik data *K*(*x*, *xi*) : fungsi kernel
b : parameter hyperplane yang dicari (nilai bias)

Kami menggunakan bantuan *svm.SVC* dari library scikit-learn dengan hyperparameter sebagai berikut: Kernel yang digunakan adalah *rbf*, dengan C regularization value sebesar 1.0. Untuk gamma, digunakan *auto*.

III.II Feedforward Neural Network (FNN)

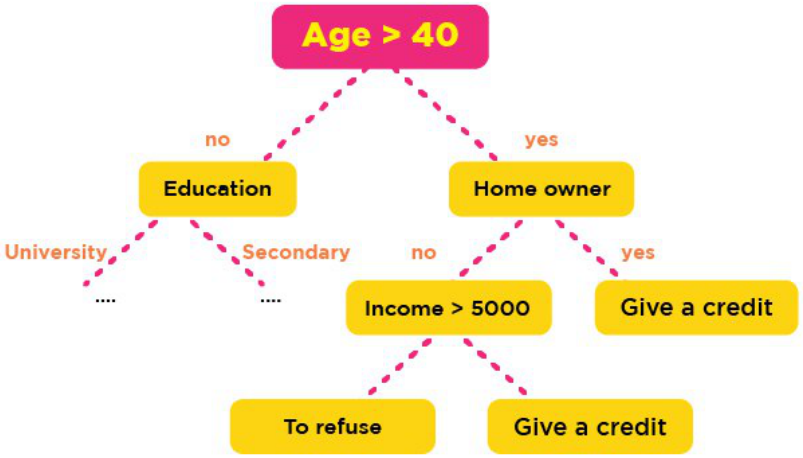


Feed-forward Neural Network adalah neural network yang mempunyai tiga bagian utama yaitu layer input, layer tersembunyi (hidden layer) dan layer output. Ada beberapa struktur model dari neural network dan salah satunya adalah neural network berstruktur feedforward (feedforward neural network). Secara sederhana, neural network yang mempunyai struktur feedforward mempunyai karakteristik tidak ada pengulangan pembelajaran (loop) di mana signal bergerak dari layer input dan melewati layer tersembunyi dan kemudian menuju layer output.

Dalam project ini, kami menggunakan bantuan *tensorflow.keras*. Kami membangun arsitektur neural network menggunakan 3 hidden layer dengan 128 neuron pada masing-masing layer. Kami menggunakan ReLU activation function pada hidden layer dan sigmoid activation function untuk output layer mengingat task pada project ini adalah binary classification. Oleh karena itu, *binary crossentropy* digunakan sebagai loss function serta Adam Optimizer dengan learning rate **isi…** sebagai optimizer function. Karena dataset imbalanced dan sebagian besar records dari stroke merupakan records sintesis, maka recall akan digunakan sebagai metrik evaluasi. Untuk mendapatkan model yang terbaik, kami menggunakan bantuan fungsi callback *ModelCheckpoint* dari *Keras* untuk menyimpan model terbaik pada saat proses training. Model yang disimpan oleh *ModelCheckpoint* inilah yang kemudian akan kami gunakan untuk evaluasi.

III.III Decision Tree

Decision tree adalah model prediksi menggunakan struktur pohon atau struktur berhirarki. Konsep dari decision tree adalah mengubah data menjadi decision tree dan decision rules. Manfaat utama dari penggunaan decision tree adalah kemampuannya untuk melakukan break down proses pengambilan keputusan yang kompleks menjadi lebih simple, sehingga pengambil keputusan akan lebih menginterpretasikan solusi dari permasalahan.



Kami menggunakan bantuan *sklearn.linear_model.LogisticRegression* dari library scikit-learn dengan hyperparameter default.

III.IV Logistic Regression

Logistic Regression adalah model prediksi yang memisahkan data menjadi dua kategori untuk mencari hubungan antara input (variabel tak-terikat) dengan hasil output (variabel terikat). Konsep yang digunakan dalam *logistic regression* adalah dengan menggunakan bantuan *threshold value* yang telah ditentukan untuk menentukan batasan antara dua kategori.

Kami menggunakan bantuan *sklearn.tree.DecisionTreeClassifier* dari library scikit-learn dengan hyperparameter sebagai berikut: Criterion yang digunakan adalah *gini*, splitter nya *best*

IV. Result

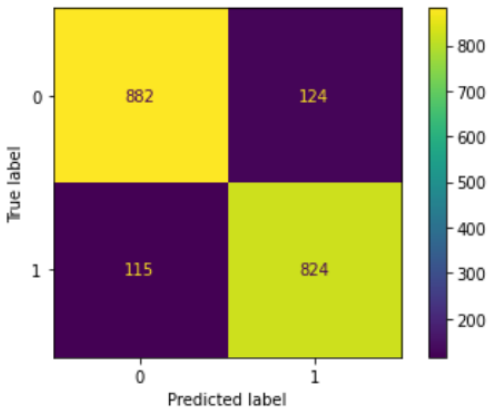
Hasil saat menggunakan ***random-sampling*** dan hasil setelah menggunakan ***SMOTE*** :

SMOTE (Synthetic Minority Oversampling Technique)		
	B E F O R E (Random Sampling)	*A F T E R (SMOTE)*
Accuracy	**88%**	**88%**
Precision	**14%**	**87%**
Recall	**20%**	**88%**
F1	**16%**	**87%**

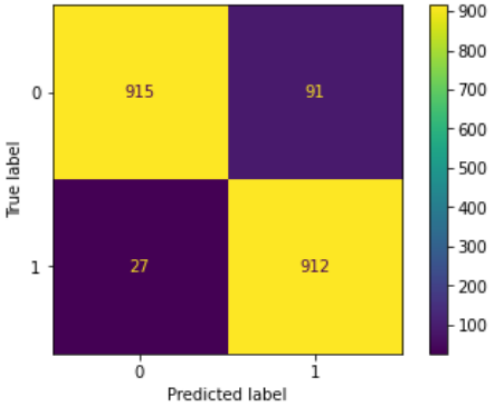
FINAL RESULT				
	SVM	***FNN***	***Decision Tree***	***Logistic Regression***
Accuracy	**88%**	**88%**	**94%**	**77%**
Precision	**87%**	**87%**	**95%**	**74%**
Recall	**88%**	**88%**	**94%**	**80%**
F1	**87%**	**87%**	**94%**	**77%**

Selain dari metrik-metrik tersebut, kami juga membuat *confusion matrix* dari setiap *classifier* yang kami gunakan dalam eksperimen seperti berikut :

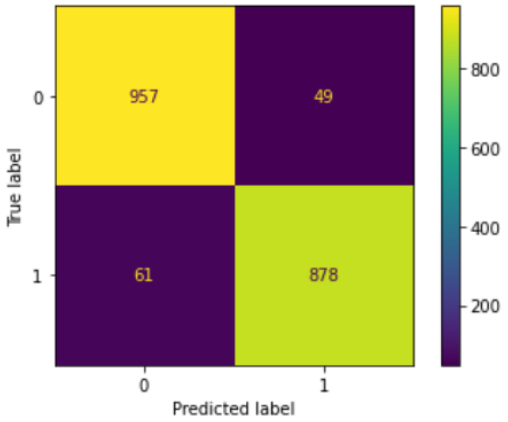
1. SVM



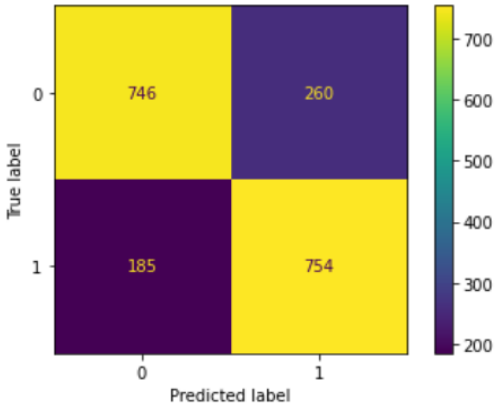
2. FNN



3. Decision Tree



4. Logistic Regression



V. Conclusion & Future Works

Kelompok kami melakukan beberapa percobaan pada tahap ***preprocessing & exploration*** dan tahap ***training***. Pada tahap ***preprocessing & exploration***, kami membandingkan dua metode oversampling, yaitu metode ***random-sampling*** dan juga metode **SMOTE**. Disini, kami melihat bahwa metode *random-sampling* tidak dapat kami gunakan pada dataset kami, sebab metode *random-sampling* hanya menambahkan *minority class* dengan melakukan duplikasi secara acak. Hal ini menyebabkan model yang *overfitting*. Oleh karena itu, kami menggunakan **SMOTE** yang bekerja dengan membuat data sintetik dan menghasilkan model yang lebih general dengan akurasi yang cukup memuaskan.

Pada tahap training, kami menggunakan empat metode yang telah kami ajukan dan kami implementasikan sebelumnya. Dimana metode-metode tersebut adalah SVM, FNN, Decision Tree, dan juga Logistic Regression. Model yang menghasilkan hasil paling tinggi adalah Decision Tree dengan akurasi sebesar 94,34%.

Kami sepenuhnya sadar bahwa project yang kami lakukan sekarang jauh dari sempurna. Oleh karena itu, beberapa hal yang kami sebutkan berikut diharapkan dapat menjadi ide untuk meningkatkan kualitas *Stroke Prediction Project* dimasa yang akan datang:

- ➔ Mencoba menggunakan metode yang paling mutakhir dan state-of-the art. Seperti gradient boosting, ataupun random forest dan metode ensemble learning lainnya.
- ➔ Dataset yang diberikan dapat dieksplorasi lebih dalam lagi dan mungkin akan ditemukan *insight* yang lebih besar daripada yang telah ditemukan.
- ➔ Tidak hanya dependent variable yang imbalanced, namun beberapa feature seperti *hypertension*, dan *heart_disease* memiliki class yang lebih banyak daripada class yang lain.

References

[1] Haixiang, G., Yijing, L., Shang, J., Mingyun, G., Yuanyue, H., & Bing, G. (2017). Learning from class-imbalanced data: Review of methods and applications. Expert Systems with Applications, 73, 220-239.

[2] Kotsiantis, S. B., Zaharakis, I., & Pintelas, P. (2007). Supervised machine learning: A review of classification techniques. Emerging artificial intelligence applications in computer engineering, 160(1), 3-24.

[3] Kumari, R., & Srivastava, S. K. (2017). Machine learning: A review on binary classification. International Journal of Computer Applications, 160(7).

[4] Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). SMOTE: synthetic minority over-sampling technique. Journal of artificial intelligence research, 16, 321-357.

[5] https://www.kaggle.com/fedesoriano/stroke-prediction-dataset, fedesoriano.