**AML Group 3**

Ying Hong (yh3538), Rohan Sheelvant (rns2167), Ardrian Wong (aaw2179), Liwen Zhu (lz2512)

**Introduction**

Diabetes is a chronic health condition that affects how your body turns food into energy. With diabetes, patients don’t make enough insulin or can’t use it well, causing much blood sugar to stay in the bloodstream. This can lead to serious health problems like heart disease, vision loss, and kidney disease.

In 2018, the Centers for Disease Control and Prevention reported that 34.2 million Americans have diabetes and 88 million have prediabetes. What’s more, the CDC estimates that 80% of prediabetics are unaware of their risks. Therefore we plan to build models that would identify whether a person is prediabetic or diabetic.

**About the Dataset**

The Behavioral Risk Factor Surveillance System (BRFSS) is a health-related telephone survey collected annually by the CDC. Our dataset comprises 253,680 survey responses to the CDC's BRFSS2015. The target variable *‘Diabetes’* has two classes. 0 is for no diabetes, and 1 is for prediabetes. There is a class imbalance in this dataset. This dataset has 21 independent feature variables, including HighBP, HighChol, Smoker, BMI, Stroke, PhysActivity, HvyAlcoholConsump, Sex, Age, Income, etc.

**Data Preprocessing**

First we started with data cleaning, our dataset had no missing values but did have 24206 duplicate rows which we removed. Next, the dataset was scaled using sklearn’s StandardScaler. Only non tree-based models were trained using the scaled dataset since tree-based models such as decision trees and random forests are scale invariant. We also observed that our dataset has far more non-diabetic samples than diabetic samples (as seen in Appendix Figure 1.). Therefore we used Synthetic Minority Oversampling Technique (SMOTE) to try and combat the class imbalance.

**Evaluation Metrics**

Since the dataset is imbalanced, we will use precision, recall, and F1 scores to evaluate the performance of our models. **Recall** score is the evaluation metric we will pay the most attention to, since the cost of a False Negative is much higher than that of a False Positive. The F1 score is also an important metric because it tells us how well our model is able to learn both classes. We trained models on both the default dataset and a dataset transformed with SMOTE for comparison.

**ML Techniques**

**1) Logistic Regression**

Logistic regression model was used to predict if the person is diabetic (class=1) or not (class=0). Initially, the unbalanced data was used to train the logistic regression model with ‘l1’, ‘l2’, and ‘elasticnet’ regularization with “C” value equal to 1 for ‘l2’ and ‘elasticnet’. While the accuracy of the overall model was around 85%, the recall on the diabetic class was terrible (approx 15%). Hyperparameter tuning was done for these models using the random search. The best hyperparameters were 'C': 0.226 and 'penalty': 'l2' which gave very similar accuracy (=86%) and recall (=15%). Hence, hyperparameter tuning does not help the model learn the data better. Therefore, SMOTE is performed on the data and then the logistic regression models are trained again. After hyperparameter tuning on this data, we were able to achieve an accuracy of 72% and recall on diabetic class of 75%. The optimal hyperparameters were 'C': 3.377, 'penalty': 'l1'. The top 5 important features in order are: GenHlth, BMI, HighBP, Age and HighCol. Refer to figures in the appendix section for feature importance of all features for Logistic Regression (with and without SMOTE)

Optimal Hyperparameters:

Logistic Regression without SMOTE: 'C': 0.226, 'penalty': 'l2'

Logistic Regression with SMOTE: C': 3.377, 'penalty': 'l1'

**2) Decision Tree**

Without SMOTE, we initially trained the Decision Tree with default parameters. Then we chose “ccp\_alpha” and “max\_depth” as hyperparameters and applied hyperparameter tuning using 3-fold cross validation based on recall for Decision Tree and found optimal parameters were {max depth: 100, ccp\_alpha: 1e-6}. The top 5 important features (ordered) are: BMI, Income, Age, PhysHlth, and Education. With SMOTE, we obtained a recall score of 0.85 in non-diabetic class in the Decision Tree with default parameters. After hyperparameter tuning based on recall, we improved the recall score to 0.86, and the optimal hyperparameters were{max depth: 50, ccp\_alpha: 1e-8}. The top 5 important features (ordered) are: BMI, Income, Age, PhysHlth, and Education. SMOTE only improved recall for non-diabete class and had no influence on other metrics (accuracy, precision, F-1 score) for both classes.

Optimal Hyperparameters:

Decision Tree with SMOTE: {max depth: 50, ccp\_alpha: 1e-8}

Decision Tree without SMOTE: {max depth: 100, ccp\_alpha: 1e-6}

**3) Random Forests**

Without SMOTE, we initially obtained an accuracy of 0.84 in the Random Forests with default parameters. We chose “max\_depth” and “n\_estimators'' as hyperparameters and applied hyperparameter tuning using 3-fold cross-validation based on recall. We obtained the optimal hyperparameters which were {max depth: 20, number of estimators: 100} and improved the accuracy to 0.85. The top 5 important features (ordered) are: BMI, Age, GenHlth, Income, PhysHlth. With SMOTE, we initially obtained a recall score in the diabetic class as 0.19. Then we improved it to 0.52 after hyperparameter tuning and found the optimal hyperparameters based on recall were {max depth: 10, number of estimators: 300}. The top 5 important features (ordered) are: HighBP, HighChol, GenHlth, DiffWalk, BMI. SMOTE improved recall and F1-score on diabetic class and precision on non-diabetic class. However, SMOTE did not improve accuracy in Random Forests on our dataset.

Optimal Hyperparameters:

Random Forests with SMOTE: {max depth: 10, number of estimators: 300}

Random Forests without SMOTE: {max depth: 20, number of estimators: 100}

**4) Support Vector Machine**

Initially, we tried to train a dual support vector machine with kernels (linear, poly, rbf), but due to our high number of sample points we were unable to train a model feasibly. Trying to train one dual support vector machine with 5-fold cross-validation during model selection would take at least 60 hours. It took over 30 hours just to complete ⅖ folds. Hence we opted to train a primal support vector machine using sklearn’s LinearSVC implementation the regularization term C and on the dataset with and without SMOTE. The random search with 100 iterations was used for the hyperparameter search and the final model was chosen based on the highest recall. The support vector machine trained with SMOTE had a lower accuracy on the test set than the support vector machine trained without SMOTE. The model trained without SMOTE had a higher recall, and f1-score, but a lower precision on the non-diabetic class than the model trained with SMOTE. In terms of the diabetic class, the model trained with SMOTE had a much higher recall and higher f1-score, but a lower precision than the model trained without SMOTE. Transforming the dataset with SMOTE was very beneficial in raising the recall and f1-score on the diabetic class, except for at the expense of lower accuracy and precision. We also observed that the four most important features overall for both SVM models are GenHlth, Age, BMI, and HighBP (not ordered). Refer to figure 9 & 10 in the appendix section for feature importance graphs.

Optimal Hyperparameters:

SVM with SMOTE: {'C': 0.046954761925470656 }

SVM without SMOTE: {'C': 5.4881350392732475}

**5) Neural Network**

We built the neural network by choosing the Rectified Linear Unit and softmax as activation functions. We chose Adam as our optimizer and categorical\_crossentropy as the loss. When we fitted the model, we set the batch size equal to 128 and 10 epochs. To prevent overfitting, we applied batch normalization and dropout at 20%. The model without SMOTE has five layers. After applying SMOTE, we noticed that more layers helped the model learn better from the minority class, so we added two more. We did not add more layers or neurons to prevent overfitting. After applying SMOTE to the data, the recall and F1 score of the diabetic class increases, but at the cost of a drop in accuracy and diabetes’ precision score. For the non-diabetic class, its precision score increased while the recall and F1 scores decreased. Since we value the recall score of the minority class, we think it is wise to apply SMOTE. Please refer to figures 11-15 for loss, accuracy, precision, recall, and F1 over epochs for the model without applying SMOTE and figures 16-20 for the model applied SMOTE.

Optimal Hyperparameters:

Without SMOTE:

* First five layers: units = 64, 32, 16, 8, 4, activation = ReLU, Dropout = 0.2, Batch Normalization
* The fifth layer: units = 2, activation = Softmax, Dropout = 0.2, Batch Normalization
* Optimizer: Adam & Loss: categorical\_crossentropy
* Batch size = 128, 10 epochs

With SMOTE:

* First seven layers: units = 64, 48, 32, 24, 16, 8, 4, activation = ReLU, Dropout = 0.2, Batch Normalization
* The fifth layer: units = 2, activation = Softmax, Dropout = 0.2, Batch Normalization
* Optimizer: Adam & Loss: categorical\_crossentropy
* Batch size = 128, 10 epochs

**Model Performance Summary**

| Model | Class | Accuracy | Precision | Recall | F1 |
| --- | --- | --- | --- | --- | --- |
| Logistic Regression  With smote | Non-diabetic | 0.71 | 0.94 | 0.71 | 0.81 |
| Diabetic | 0.32 | 0.76 | 0.45 |
| Logistic Regression  Without smote | Non-diabetic | 0.85 | 0.86 | 0.98 | 0.92 |
| Diabetic | 0.54 | 0.15 | 0.24 |
| Decision Tree  With Smote | Non-diabetic | 0.77 | 0.87 | 0.86 | 0.86 |
| Diabetic | 0.29 | 0.32 | 0.30 |
| Decision Tree  Without Smote | Non-diabetic | 0.77 | 0.87 | 0.85 | 0.86 |
| Diabetic | 0.29 | 0.32 | 0.30 |
| Random Forests  With Smote | Non-diabetic | 0.81 | 0.91 | 0.86 | 0.88 |
| Diabetic | 0.40 | 0.52 | 0.45 |
| Random Forests  Without Smote | Non-diabetic | 0.85 | 0.86 | 0.98 | 0.92 |
| Diabetic | 0.54 | 0.15 | 0.23 |
| SVM With Smote | Non-diabetic | 0.71 | 0.94 | 0.70 | 0.81 |
| Diabetic | 0.32 | 0.77 | 0.45 |
| SVM Without Smote | Non-diabetic | 0.85 | 0.85 | 0.99 | 0.92 |
| Diabetic | 0.58 | 0.07 | 0.12 |
| Neural Network  With Smote | Non-diabetic | 0.85 | 0.85 | 0.99 | 0.92 |
| Diabetic | 0.69 | 0.06 | 0.12 |
| Neural Network  Without Smote | Non-diabetic | 0.74 | 0.93 | 0.75 | 0.83 |
| Diabetic | 0.33 | 0.70 | 0.45 |

**Conclusion**

The SMOTE technique helps most of the models improve their recall and f1-score on the diabetic class, but at the cost of decreased precision and accuracy. The logistic regression model with SMOTE and the Support Vector Machine model with SMOTE perform well. They have a high accuracy score and learn well from the diabetic class and the non-diabetic class. If we were doctors, we would choose these two models to predict whether a patient is prediabetic or diabetic. However, we do note that this recall is not very high, so in practice we can’t completely rely on these models.

Github link of project submission: <https://github.com/W4995-AML/final-project-deliverable---code-rohansheelvant>

**Appendix**

****

****

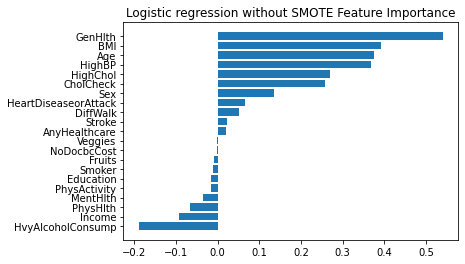
****

Figure 3.

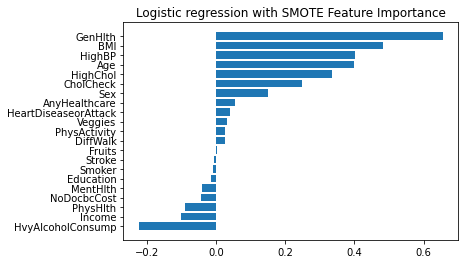
****

Figure 4.

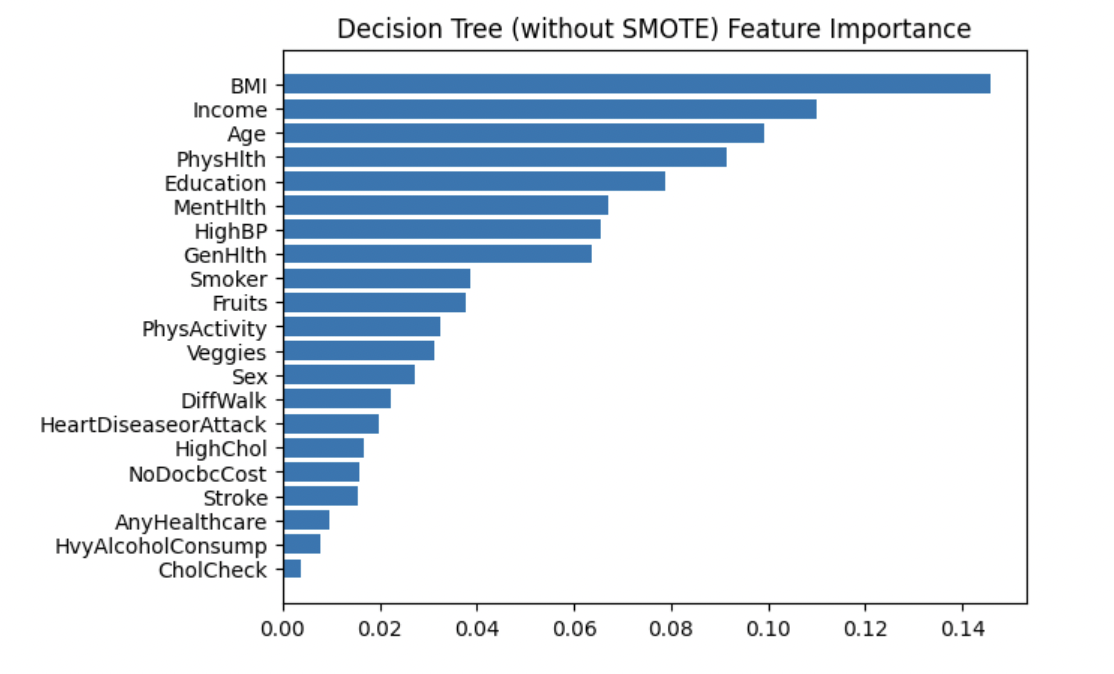
****

Figure 5.

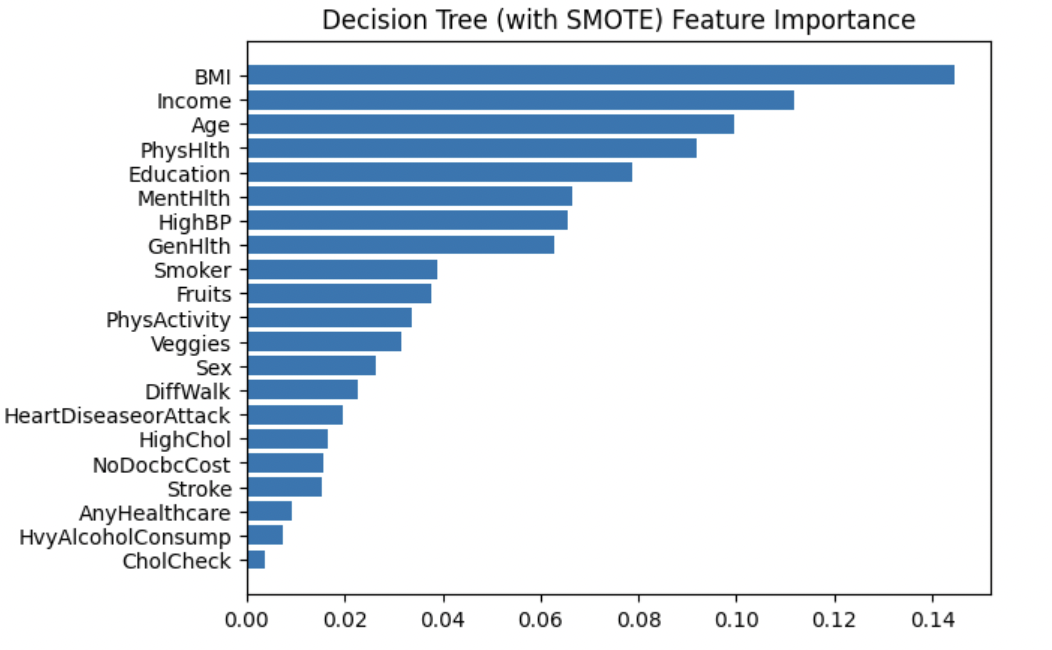


Figure 6.

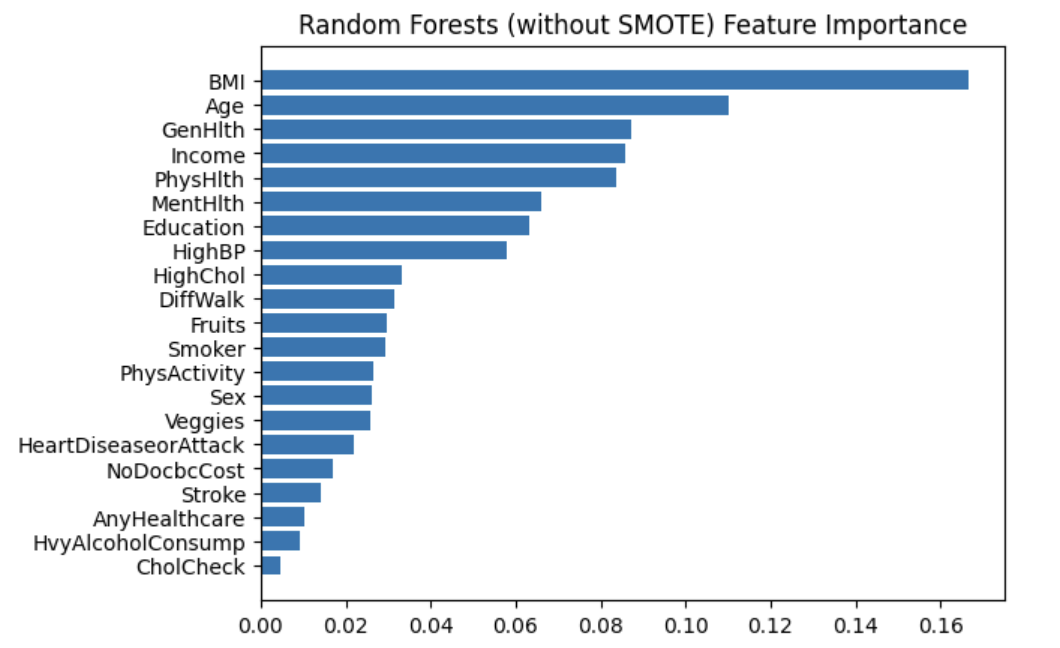


Figure 7.

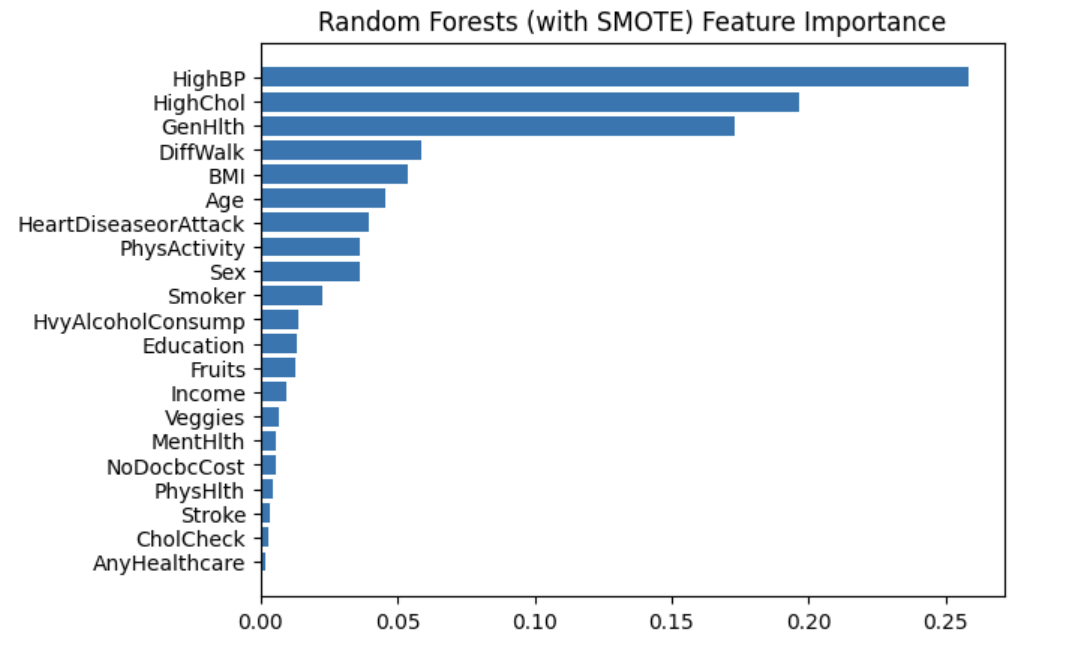


Figure 8.

****

****

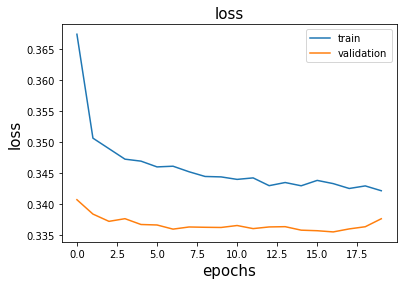


Figure 11.

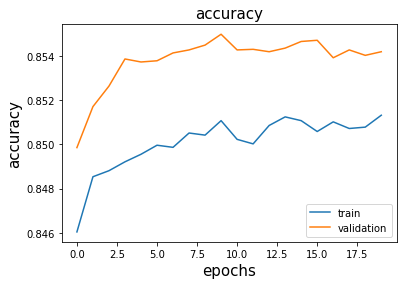


Figure 12.

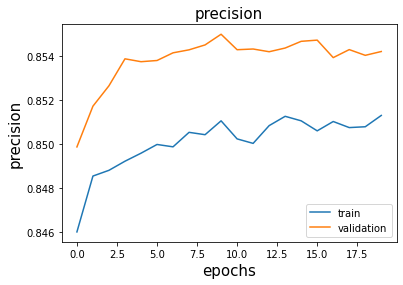


Figure 13.

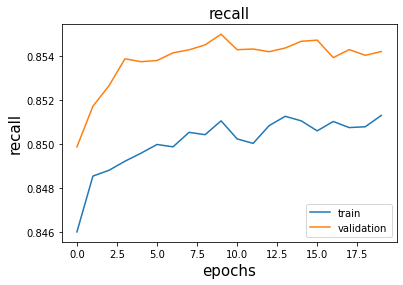


Figure 14.

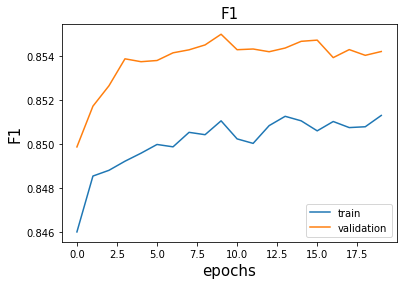


Figure 15.

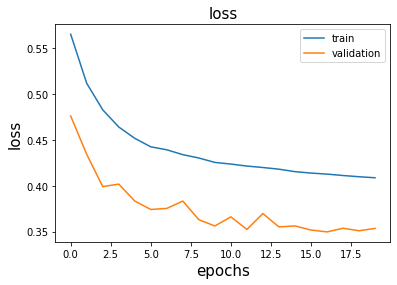


Figure 16.

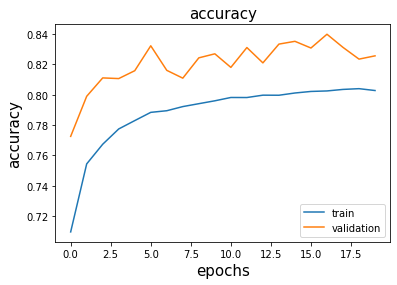


Figure 17.

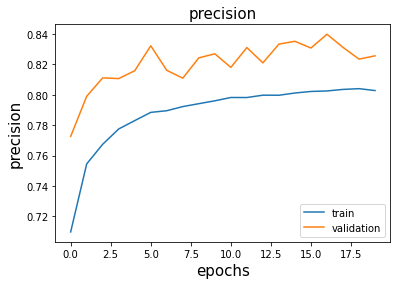


Figure 18.

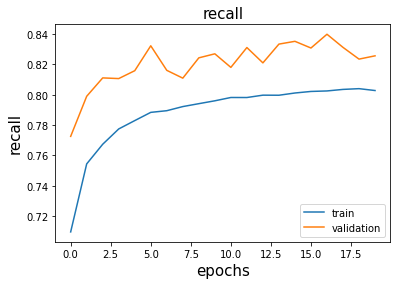


Figure 19.

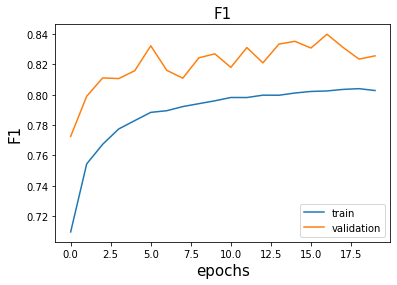


Figure 20.

**Reference**

<https://www.cdc.gov/diabetes/basics/diabetes.html>

<https://www.kaggle.com/datasets/alexteboul/diabetes-health-indicators-dataset?select=diabetes_binary_health_indicators_BRFSS2015.csv>