Dylan Cozloff
AST04
CS 422 1001

Diabetes Prediction Logistic Regression Model

Installs:
pip install pandas
pip install scikit-learn
pip install matplotlib
pip install seaborn

Dataset: https://www.kaggle.com/datasets/iammustafatz/diabetes-prediction-dataset/data

I used a diabetes prediction dataset (100,000 entries) which is a collection of medical and demographic data from patients, with their diabetes status. All features: age, gender, body mass index (BMI), hypertension, heart disease, smoking history, HbA1c level, and blood glucose level. But, I only manually extracted four continuous features (age, BMI, HbA1c level, and blood glucose level) from this dataset. The binary target was whether the patient was positive or negative for diabetes. I loaded the dataset using the pandas library, in which I use the function read_csv().

Training/Test Data Splits: I used the sklearn.model_selection function train_test_split() to use a random 20% of the dataset for testing, and the remaining for training.

Model Training: I used the sklearn.linear_model functions SGDClassifier() with log_loss parameter indicating logistic regression, alongside with fit() to fit the training data using this logistic regression model. Also, I used sklearn.pipeline make_pipeline(), and sklearn.preprocessing StandardScaler() to make the training and testing streamlined and feature scaled.

Model Predictions: I used the sklearn.linear_model predict() function to perform classification on both the training and test vectors. I also used predict_proba() to get the probability estimates for these vectors as well.

Model Accuracy: I used the sklearn.metrics function accuracy_score() to pass the training & test data with it's classification predictions to get a more precise model accuracy as the training and test accuracy were very close: **Training Accuracy** = 0.9608, **Test Accuracy** = 0.9583. As you can see the model generalized very well, accuracy dropped of by only 0.25%. Comparing these results with the NBC model: **Training Accuracy** = 0.95685, **Test Accuracy** = 0.95585. As you can see the model generalized better, accuracy dropped of by only 0.1% vs 0.25% comparitively suggesting slightly more overfitting from the logistic regression model. But the overall test accuracy was still better than the NBC model by 0.245%.

Dylan Cozloff
AST04
CS 422 1001

Classification Report (Compared with NBC):

Test Data:

	precision	recall	f1-score	support
0	0.96	0.99	0.98	18292
1	0.91 (+0.06)	0.57 (-0.02)	0.69 (+0.01)	1708
accuracy			0.96	20000
macro avg	0.93 (+0.02)	0.79 (-0.01)	0.84	20000
weighted avg	0.95 (+0.01)	0.96	0.95	20000

Sensitivity (Test Data): 57% (down 2%)
Specificity (Test Data): 99%
Log Loss (Test Data): 0.12027655485716757 (down 1%)

Training Data:

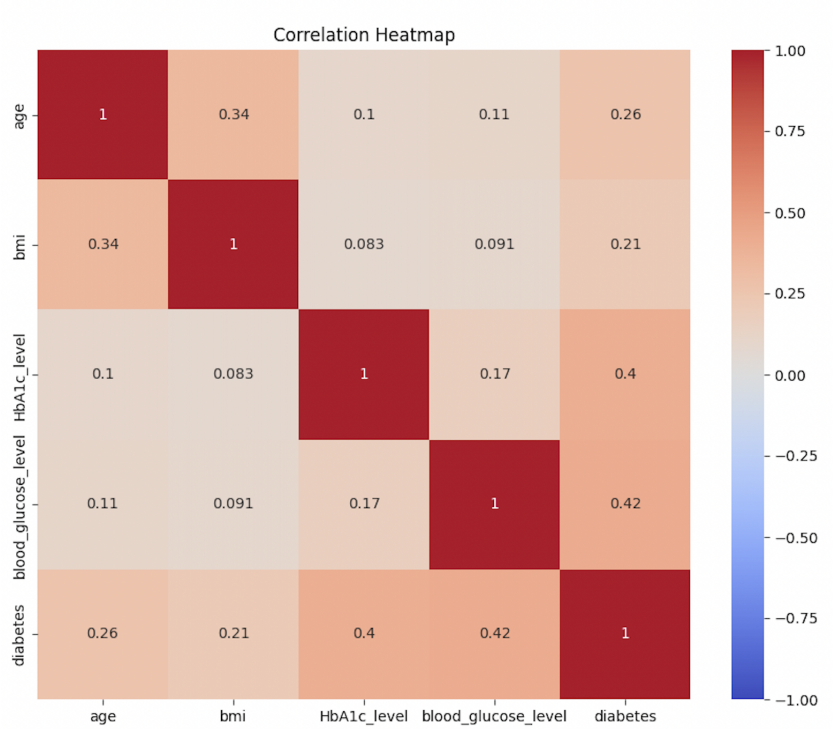
	precision	recall	f1-score	support
0	0.96	1.00 (+0.01)	0.98	73208
1	0.85 (+0.07)	0.59 (-0.01)	0.72 (+0.02)	6792
accuracy			0.96	80000
macro avg	0.94 (+0.03)	0.79	0.85 (+0.01)	80000
weighted avg	0.96 (+0.01)	0.96	0.96 (+0.01)	80000

Sensitivity (Training Data): 59% (down 1%)
Specificity (Training Data): 100% (up 1%)
Log Loss (Training Data): 0.117290095780991859 (up 1%)

NEXT PAGE

Dylan Cozloff
AST04
CS 422 1001

Correlation Heat Map:



I used the corr() function to evaluate the dataset, and matplotlib.pyplot / seaborn to visualize the correlation matrix. As you can see there is definitely a correlation between diabetes and HbA1c_levels, blood glucose levels, age, and bmi.