Bard collage at simons's rock

Diabetes Prediction

Machine learning project

Name Muska arghistani

We used Logistic Regression to set up a model for binary classification in this project. Logistic Regression is a simple but strong model for figuring out how likely it is that a certain event will happen, like whether a person has diabetes or not. The model projects a chance score between 0 and 1 based on a number of health-related factors, such as age, glucose levels, and body mass index (BMI). If the number is more than 0.5, the model says the person probably has diabetes. If it's less than 0.5, it says they don't have diabetes.

Logistic Regression is easy to understand in terms of how it works. We use a linear model to explain it. For each feature , like glucose, we multiply it by a variable and then add up the results. After this sum is added up, it is run through a sigmoid function, which turns it into a chance between 0 and 1. Logistic Regression works well for this case because there are only two possible outcomes.

I want to make a project that can look at health info about a new patient and guess if they might have diabetes or not. As an example: New information about patients: Blood sugar = 135, BMI = 30, age = 50 Prediction for the project: No diabetes (0)

The model's key features include:

- **Interpretability**: The model provides interpretable coefficients that show how each feature affects the prediction.
- **Efficient Training**: Logistic Regression is computationally efficient, meaning it trains quickly even on relatively large datasets.
- **Probability Outputs**: Rather than just predicting 0 or 1, the model outputs a probability, which can be useful in medical decision-making, where uncertainty might need to be considered.

Strengths:

- **Simplicity**: Logistic Regression is easy to implement and understand, making it a great choice for many classification problems.
- **Interpretability**: The model's coefficients provide insight into the relationship between the input features and the predicted outcome.
- **Low computational cost**: The model trains quickly and requires fewer resources than more complex models like neural networks.

Limitations:

- **Linearity**: Logistic Regression assumes a linear relationship between the features and the log-odds of the outcome. If the relationship is more complex, the model might struggle to capture it.
- **Binary Classification**: Logistic Regression is primarily suited for binary outcomes. Extensions like multinomial logistic regression are needed for multi-class problems.
- **Sensitive to outliers**: Logistic Regression can be affected by extreme values in the data, leading to unstable coefficients.

Data Adaptation and Model Tuning

Getting the data from a CSV file is the first thing that needs to be done to prepare the information. To read the data and look at the structure of it, we used the following code:

```
import pandas as pd
# Load the dataset
diabetes_dataset = pd.read_csv('/content/diabetes.csv')
```

We looked at the features after getting the data. Women who are pregnant, blood sugar, blood pressure, skin thickness, insulin, body mass index (BMI), diabetes pedigree function, age, and the result column (result) that shows if the patient has diabetes (1) or not (0).
Next, we used StandardScaler to make sure that all of the features in the data were on the same scale. This makes sure that no feature stands out because of how big it is.

```
scaler = StandardScaler()
standardized_data = scaler.transform(x)
x = standardized_data
y = diabetes_dataset['Outcome']
```

Feature Selection

When choosing features, we chose to keep all of them because each one is important for health in its own way. For instance, glucose levels and BMI are known signs of diabetes, so leaving any of these out could make it harder for the model to make accurate estimates. Because it is normalized, Logistic Regression also naturally minimizes the effect of features which aren't important.

Parameter Tuning

In Logistic Regression, we set the max_iter option to determine how many times the model needs to run before it ends. At first, the model did not converge with the usual number of rounds (100), so we raised it to 1000 to make sure it did. For this project, no other factors were set. To find the best hyperparameters for more complicated models, methods such as grid search or random search would be used.

```
classifier = LogisticRegression(max_iter=1000)
classifier.fit(X_train, Y_train)
```

Challenges and Solutions

I faced with a lot challenges such as find the dataset and look for suitable codes. My biggest challenge, Making sure the data was being handled correctly was one of the problems we had to deal with. If you didn't scale, some features, like glucose, had much higher values than others, like age. This could cause the model to put too much weight on these features. This issue was fixed by using StandardScaler, which made sure that all features were scaled to the same range.

The Logistic Regression model had problems with converge, which had to be fixed by making the number of steps bigger.

Evaluation Metrics

The high accuracy score was the main way we measured how well the model performed. Accuracy is the number of accurately expected instances relative to the total number of cases in the collection.

```
X_train_prediction = classifier.predict(X_train)
training_data_accuracy = accuracy_score(X_train_prediction, Y_train)
```
However, because this is a medical job, things like F1 score, accuracy, and memory are also important. These measurements help figure out how well the model can identify the small group (diabetes persons).

Results Discussion

The model had a training accuracy of X% and a test accuracy of Y%, which means it did well on the training data and does well with new data as well. But in a medical setting, where false positives and false negatives could have big effects, accuracy might not be enough on its own.

Insights

What I learned from this project shows how important it is to scale features and tune parameters carefully. Logistic Regression made a model that was easy to understand, and the results helped figure out which health factors had a significant impact on the projected incidence of diabetes.

Reflection on the Learning Process

As I started on this project, I learned how to preprocess data, choose features, and evaluate models. One of the most difficult parts was figuring out how different preparation methods, like scaling, could change how well the model performed. In addition to making and tuning the model, it was fun to learn how to use scikit-learn.

I learned a lot more about how machine learning models apply in the real world, especially in healthcare, thanks to this project. I learned how to use models like Logistic Regression to make guesses based on trends in data and how important it is to test these models carefully to make sure they succeed and

Citetion

-
- Kurnaz, F. S., Hoffmann, I., and Filzmoser, P. "Robust and Sparse Estimation Methods for High-Dimensional Linear and Logistic Regression." *Chemometrics and Intelligent Laboratory Systems*, vol. 172, 2018, pp. 211–222. https://doi.org/10.1016/j.chemolab.2017.12.005.
-
- Ponnet, J., Segaert, P., Van Aelst, S., et al. "Robust Inference and Modeling of Mean and Dispersion for Generalized Linear Models." *Journal of the American Statistical Association*, 2023. https://doi.org/10.1080/01621459.2022.2140054.
-
- Domínguez-Rodríguez, S., Serna-Pascual, M., Oletto, A., et al. "Machine Learning Outperformed Logistic Regression Classification Even with Limited Sample Size: A Model to Predict Pediatric HIV Mortality and Clinical Progression to AIDS." *PLOS ONE*, vol. 17, no. 10, 2022, e0276116. https://doi.org/10.1371/journal.pone.0276116.