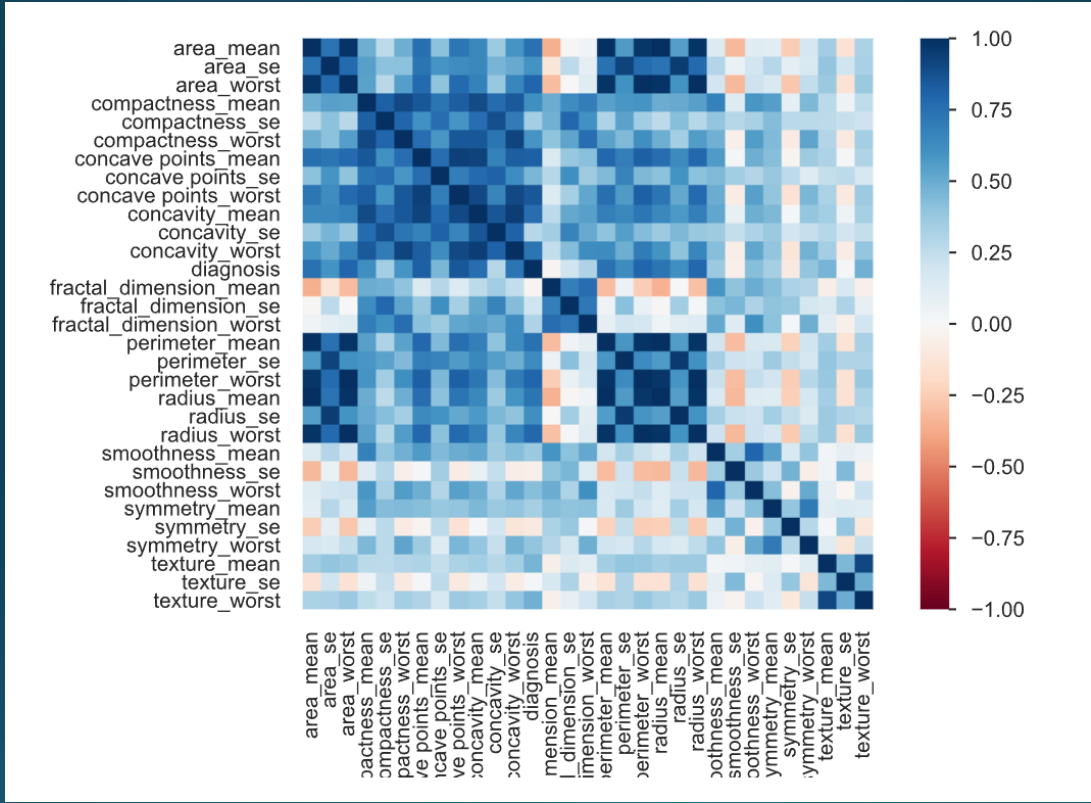
Report on Breast Cancer Prediction using Logistic Regression

PREPARED BY
SHAKIL AHAMMED

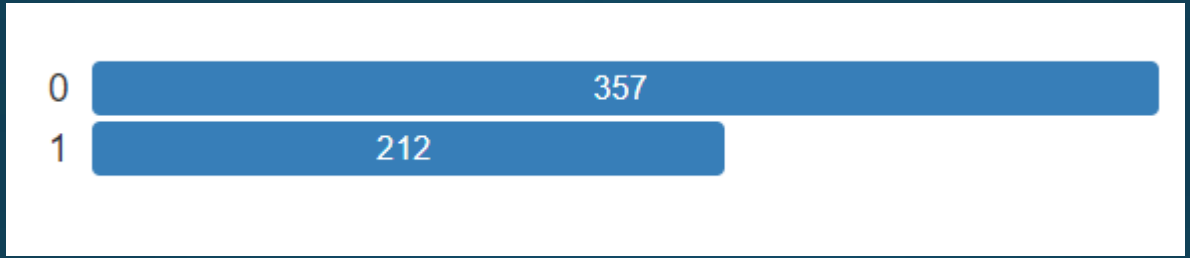
Introduction

We will be working with the Breast Cancer dataset, which contains some very detailed measurements of cells. Along with each observation of measurements, we have the diagnosis of the cell (malignant or not).
Our goal is to train a model that will be able to predict whether or not a given cell is malignant given only its measurements.

Insights



Correlation between columns.



In diagnosis column, M(Malignant)= 1, B(Benign)= 0. Most cases are Benign(357).

Normalize the data

```
from sklearn.preprocessing import StandardScaler

# Create a scaler object
scaler = StandardScaler()

# Fit the scaler to the data and transform the data
X_scaled = scaler.fit_transform(X)

# X_scaled is now a numpy array with normalized data
```

The data is not yet normalized. This can be a problem, because the units of our variables are not necessarily in the same units. Also, there might be some outliers that could cause our model to perform badly.
What we do in these cases is normalize the data before feeding it into our model. This will improve the performance of our machine learning algorithm.

```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X_scaled, y, test_size=0.30, random_state=42)
```

```
from sklearn.linear_model import LogisticRegression

# Create logistic regression model
lr = LogisticRegression()

# Train the model on the training data
lr.fit(X_train, y_train)

# Predict the target variable on the test data
Predict = lr.predict(X_test)
```

We then split the dataset into a training set and a testing set. Then create a Logistic Regression Model.

Evaluate the model

```
lr.score(X_test, y_test)
```

```
0.9824561403508771
```

```
from sklearn.metrics import classification_report
print(classification_report(y_test, Predict))
```

```
              precision    recall  f1-score   support

           0       0.99      0.98      0.99       108
           1       0.97      0.98      0.98        63

    accuracy                           0.98       171
   macro avg       0.98      0.98      0.98       171
weighted avg       0.98      0.98      0.98       171
```

we have trained a logistic regression model to predict the target variable using a dataset of input features. As you can see here, after training the model on the training set and evaluating its performance on the test set, we achieved a final accuracy of 0.98. This is a strong performance and indicates that the model is able to make accurate predictions on new, unseen data.

Conclusion

We have finished our analysis. We have used the data from the open dataset Breast Cancer in order to build a model that will predict if a given cell is malicious or not given certain measurements of its nucleus. This model, now that it is trained, can, evidently, be extremely useful to perform punctual analysis on given cells for a hospital.

However, since the model is easily callable in a python function to make predictions, this kind of model can easily be added to a server technology such as Flask and serve a front-end application that doctors can use.

For example, we could build an interface where a doctor inputs some measurements that she has performed and the model would output if the cell is malicious or not. Or maybe, a more realistic use of this model could be to connect the backend application to a machine that takes a sample of tissue, measures all the cells and the performs a diagnosis.

By using python, the API of such an application would extremely minimalist, and all we did to save the lives of people was train a logistic regression model on the hospital's data.