Classification Report – KNN

Kiran Venkatesh Kulkarni-1001848434

Vijetha Shenoy Badiadka- 1001822855

- ## Pre-Processing the data:

  After examining the dataset it was found that there are no null values in the dataset, I performed the pre-processing step using df.info() to check for null values and it.

  Since there are lot of zero values in few columns such as preg, skin, test to select the best three attributes we neglect test as it has more than 50 percent zero values out of 798 entries.

- ## Parameters of KNN:

  n_neighbors: we use number of neighbors to check how many neighbouring clusters we need to choose from to train and test our model. It is nothing but the value of k.

  Weights:
  We specify how many neighbours we need to select to train our model. Default value is uniform- all points are weighted equally.

  Algorithm:
  The algorithm to be used to calculate our neighbors. If we use brute- we calculate distance from every training point. If we use auto- it will decide whether we should use brute or tree based algorithm based on training data.

  Leafsize:
  It determines the number of leaf points associated with a leaf in the tree. If we add 64 points, if each point is linked to a leaf, we get a tree that is seven levels deep, however, if sizteen points are associated with the leaf we only get a tree that is three levels

deep. It is only applicable if we are using BallTree or KDTree algorithm

P:
Power parameter for Minkowski, if p=1- it is Manhattan distance else if p=2 it is Euclidean distance.

Metric:
The distance metric used to compute our neighbors, we can either use Minkowski or Euclidean to calculate the distance. By default it is Minkowski.

N_jobs:
If our processor is good, we can increase the number of jobs that will be run at the same time, n is the number of processors on our system.

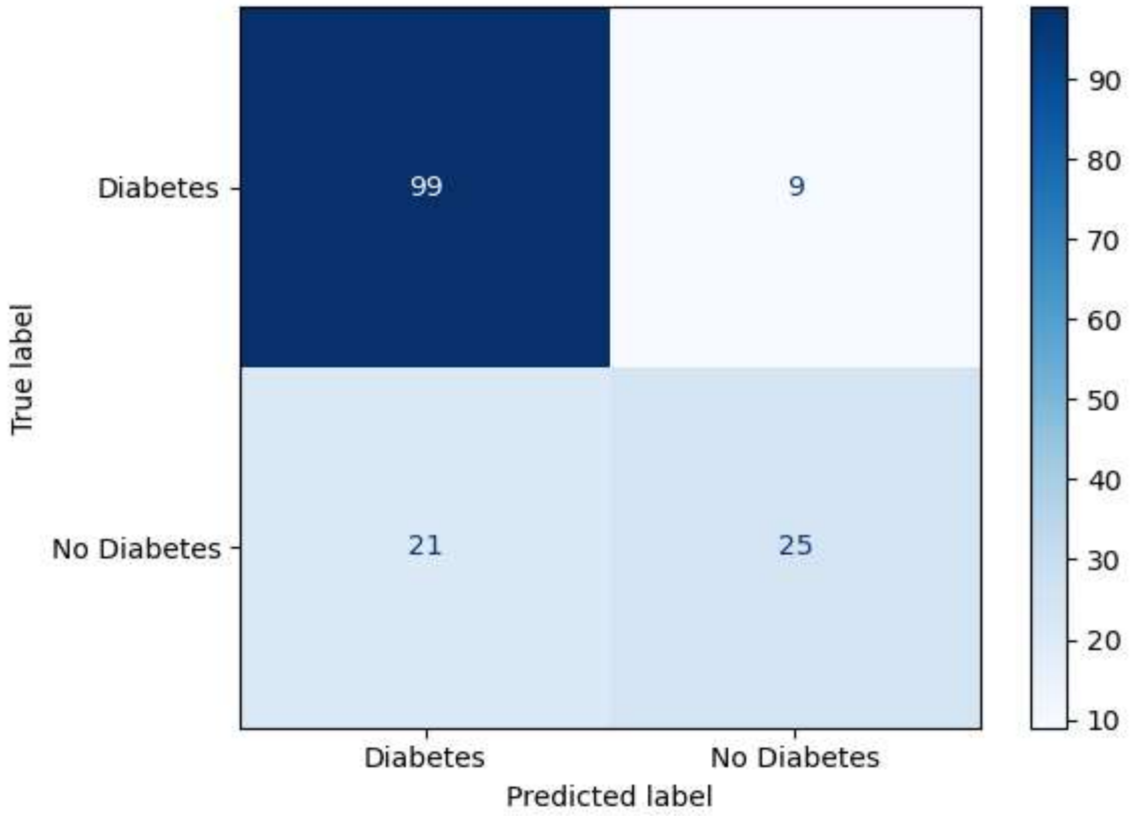
- **Criteria for selecting three attributes**:

Since there were many zero values in test column, when we calculated the best attributes using Chi-Square method, the results we got were skewed towards the test column hence we decided to drop the test column to select the attributes.
When I performed Chi- Square test on other attributes, the results we got are as follows.
Based on the top three scores, we selected Plas, age and mass as our three best attributes.

```
features        scores
0      Plas  1411.887041
4       age   181.303689
2      mass   127.669343
```

- **Visualization of the classifier in a 2-D projection, and write your observations.**

FOR NEIGHBORS=7

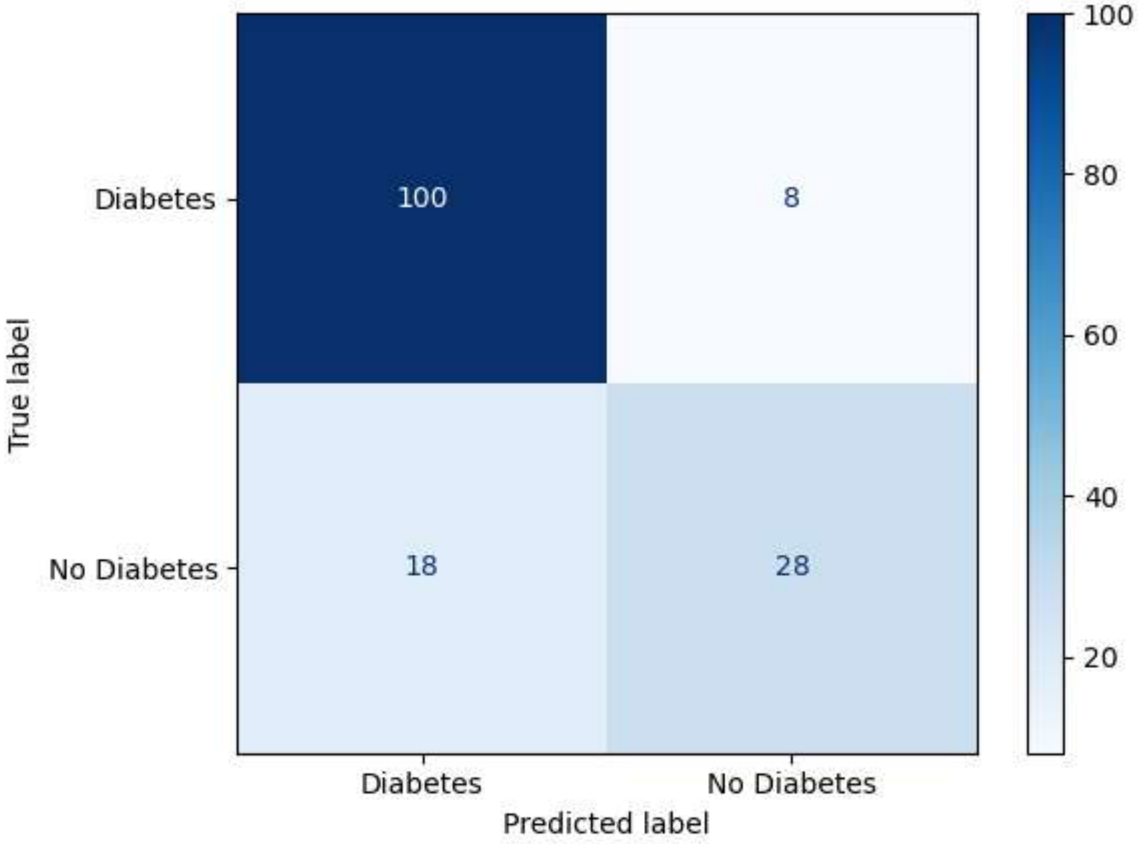


```
Classification Report:
              precision    recall  f1-score   support

           0       0.82      0.92      0.87       108
           1       0.74      0.54      0.62        46

    accuracy                           0.81       154
   macro avg       0.78      0.73      0.75       154
weighted avg       0.80      0.81      0.80       154
```

FOR NEIGHBORS=9

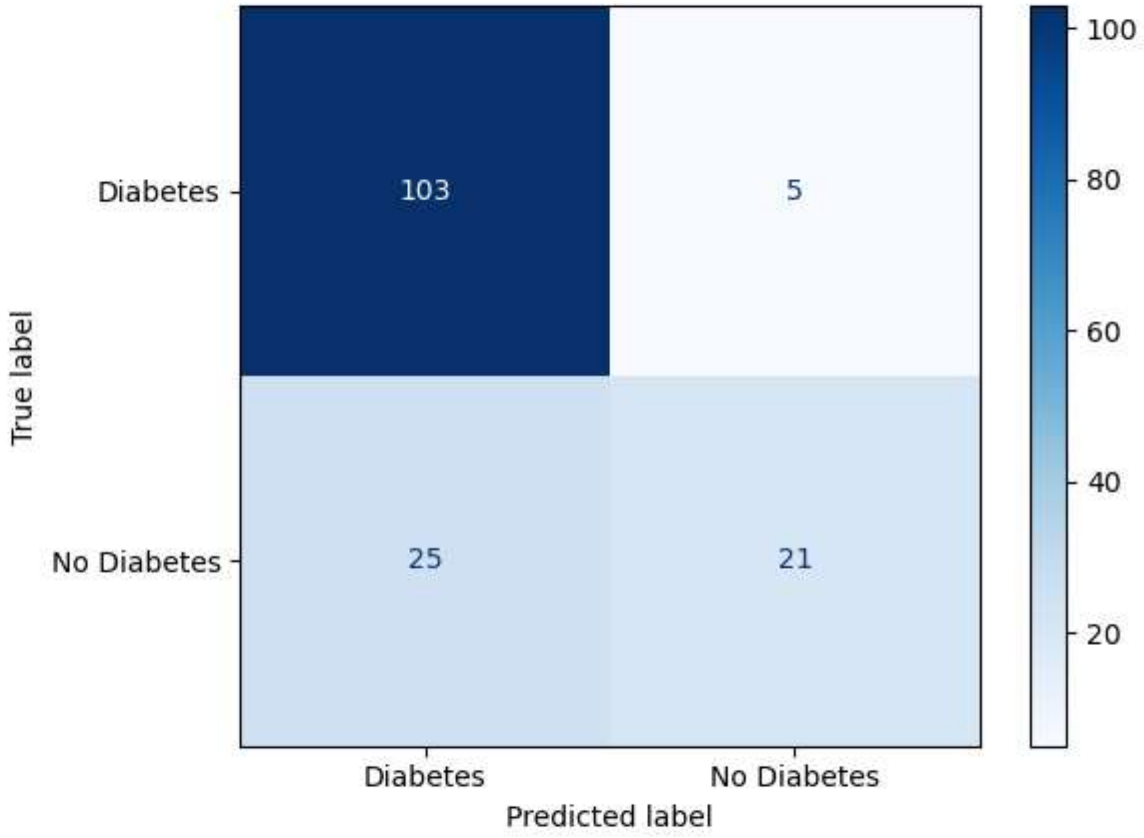


```
Classification Report:
              precision    recall  f1-score   support

           0       0.85      0.93      0.88       108
           1       0.78      0.61      0.68        46

    accuracy                           0.83       154
   macro avg       0.81      0.77      0.78       154
weighted avg       0.83      0.83      0.82       154
```

FOR NEIGHBORS=10



```
Classification Report:
              precision    recall  f1-score   support

           0       0.80      0.95      0.87       108
           1       0.81      0.46      0.58        46

    accuracy                           0.81       154
   macro avg       0.81      0.71      0.73       154
weighted avg       0.81      0.81      0.79       154
```

• Interpret and compare your results.

Based on the number of neighbours the accuracy for the confusion matrix changes
Accuracy when neighbour = 7 is 80
Accuracy when neighbour = 9 is 83.11
Accuracy when neighbour = 10 is 80.51

We get the best accuracy when we consider the neighbour value = 9

References

- https://scikit-learn.org/stable/modules/generated/sklearn.neighbors.KNeighborsClassifier.html
- Kaggle
- https://towardsdatascience.com/knn-using-scikit-learn-c6bed765be75