Classification Using Several Classification Techniques

Harold Okai

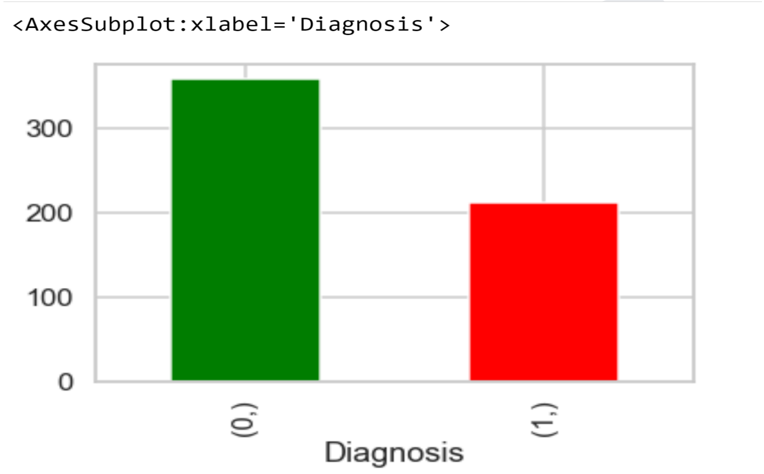
Data Origin

Features for this dataset were computed from a digitized image of a fine needle aspirate (FNA) of a breast mass. They describe characteristics of the cell nuclei present in the image. A few of the images can be found at http://www.cs.wisc.edu/~street/images/

Separating plane described above was obtained using Multisurface Method-Tree (MSM-T) [K. P. Bennett, "Decision Tree Construction Via Linear Programming." Proceedings of the 4th Midwest Artificial Intelligence and Cognitive Science Society, pp. 97-101, 1992], a classification method which uses linear programming to construct a decision tree. Relevant features were selected using an exhaustive search in the space of 1-4 features and 1-3 separating planes.

Logistic Regression

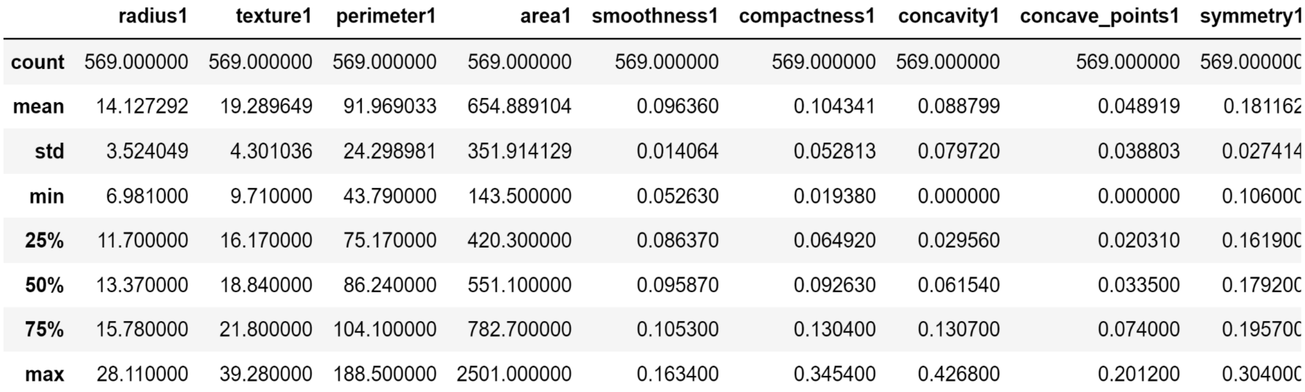
The main objective of this analysis was to use three different Classification techniques to predict breast cancer malignancy. The target dataset was labeled B for benign and M as malignant. Which I later encoded to 1 for malignant and 0 for benign.



```
<AxesSubplot:xlabel='Diagnosis'>
```

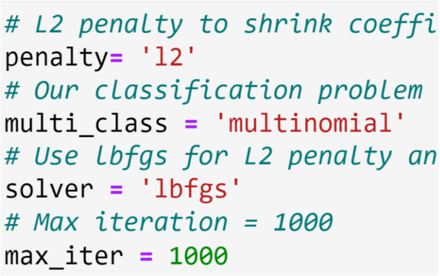
We can see that the classes are a little bit unbalanced but we will deal with that when we get to the classification section. Right now some summary statistics on our dataset.

Summary statistics show some of the features in the dataset, but also



	radius1	texture1	perimeter1	area1	smoothness1	compactness1	concavity1	concave_points1	symmetry1
count	569.000000	569.000000	569.000000	569.000000	569.000000	569.000000	569.000000	569.000000	569.000000
mean	14.127292	19.289649	91.969033	654.889104	0.096360	0.104341	0.088799	0.048919	0.181162
std	3.524049	4.301036	24.298981	351.914129	0.014064	0.052813	0.079720	0.038803	0.027414
min	6.981000	9.710000	43.790000	143.500000	0.052630	0.019380	0.000000	0.000000	0.106000
25%	11.700000	16.170000	75.170000	420.300000	0.086370	0.064920	0.029560	0.020310	0.161900
50%	13.370000	18.840000	86.240000	551.100000	0.095870	0.092630	0.061540	0.033500	0.179200
75%	15.780000	21.800000	104.100000	782.700000	0.105300	0.130400	0.130700	0.074000	0.195700
max	28.110000	39.280000	188.500000	2501.000000	0.163400	0.345400	0.426800	0.201200	0.304000

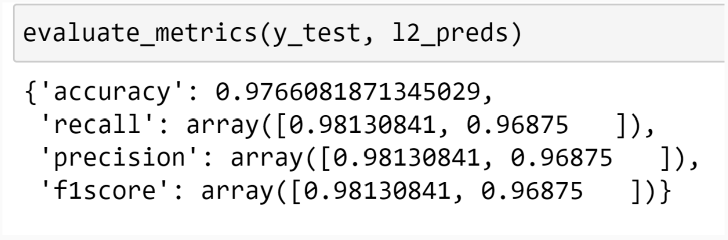
more importantly it shows that the features are highly disproportionate and that we might need to scale them. But first let us look at how our dataset did on a simple logistic regression model. Below are some of the hyperparameters used in the classification.



```
# L2 penalty to shrink coeffi
penalty= 'l2'
# Our classification problem
multi_class = 'multinomial'
# Use lbfgs for L2 penalty an
solver = 'lbfgs'
# Max iteration = 1000
max_iter = 1000
```

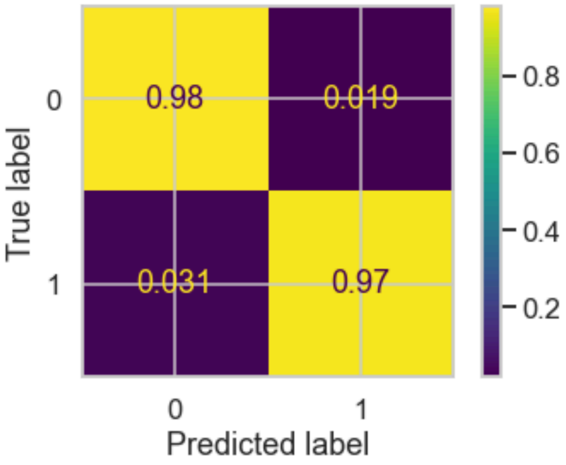
I used the L2 penalty as regularization and multi class classification, with a 1000 iterations.

We can see that the L2 logistic regression model did well on our hold out set. With not only a good f1 score but our precision and recall was also high.



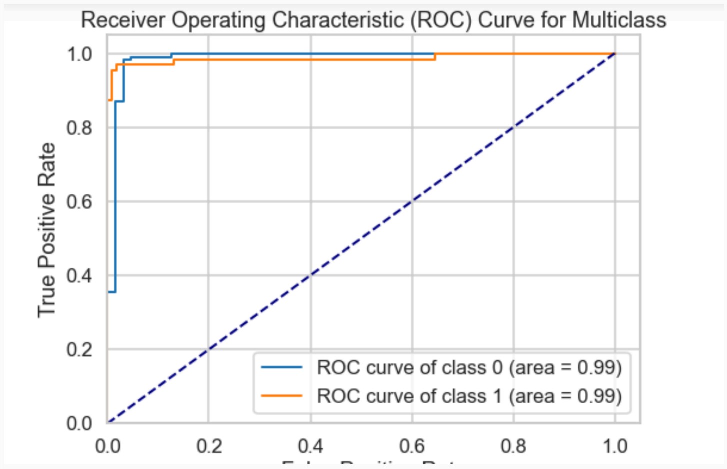
```
evaluate_metrics(y_test, l2_preds)
```

```
{'accuracy': 0.9766081871345029,
 'recall': array([0.98130841, 0.96875   ]),
 'precision': array([0.98130841, 0.96875   ]),
 'f1score': array([0.98130841, 0.96875   ])}
```



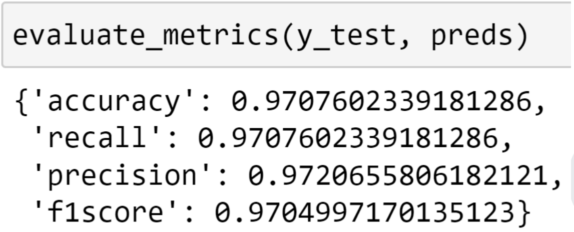
Our confusion matrix can be seen to the left. The model was able to predict with an f1 score of 0.98, which was a little bit better than our accuracy score. Our TP and TN were high enough to suggest a good model.

Our ROC curve also shows that we able have a near perfect score with the curves of both 0 and 1 being close to the border lines to top left meaning our FP rate was low.



K Nearest Neighbour

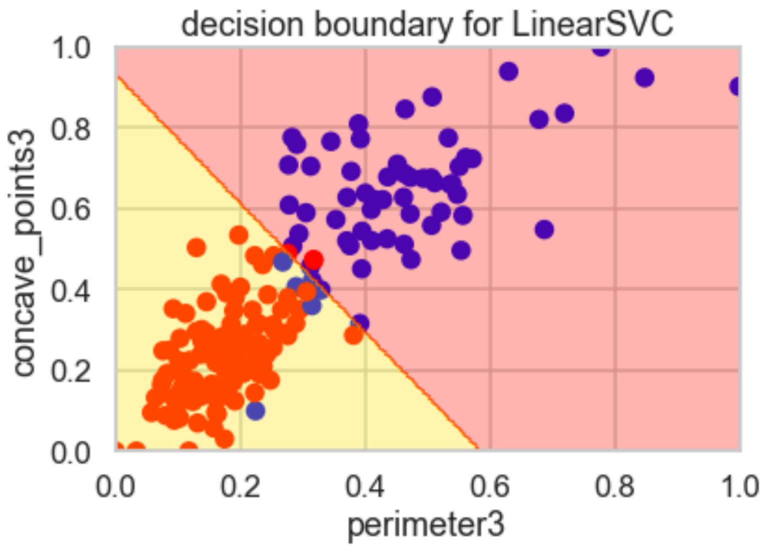
Next we train the KNN with 2 neighbors since we already know the target number of the classes. The KNN model also shows high f1 and accuracy scores on our hold out sets.

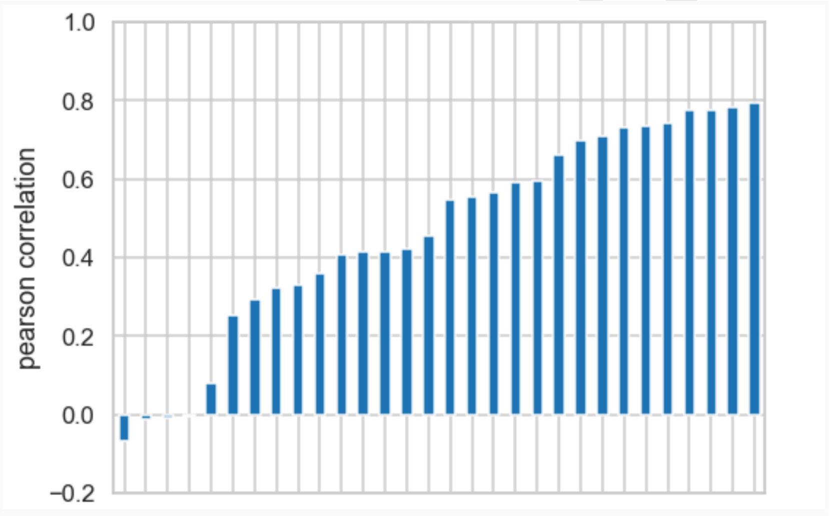


```
evaluate_metrics(y_test, preds)
```

```
{'accuracy': 0.9707602339181286,
 'recall': 0.9707602339181286,
 'precision': 0.9720655806182121,
 'f1score': 0.9704997170135123}
```

SVC and a Linear Decision Boundary
Last but not least is the linear SVC model which also shows a good decision boundary for the two classes. The features used were the most correlated columns in the dataset to our target data. It seemed that the higher the pearson correlation to the target features, the better our SVC model is able to separate our classes.





Best Model for the dataset?
From experience working on this project I it was evidently clear that the best datasets work better when you have continuous variables as your features. Categorical features that are '0's and '1's tend to get stuck in the upper and lower limits of the graphs and make it difficult to visualize. The linear SVC model also doesn't do very well with predictions beyond binary classifications. They can be difficult to visualize and interpret and often quite messy.

Conclusion

The luck of having a good dataset in this project really helped in the classification supporting all three chosen methods. I guess I would like to see another dataset that has lower correlation to the target variable to see how bad it predicts the target variable.

References

Wolberg,William, Mangasarian,Olvi, Street,Nick, and Street,W.. (1995). Breast Cancer Wisconsin (Diagnostic). UCI Machine Learning Repository. https://doi.org/10.24432/C5DW2B.