Springboard Data Science Course

Data Science Capstone Project 1

Orthopedic Biomechanical Features

Michelle Ide - 12/29/2020

~~~ SUMMARY REPORT ~~~

This project proposed to improve speed in identifying spondylolisthesis of the lumbar back using quantitative x-ray measurements. What follows is a description of the dataset, specific considerations for machine learning models, and the results of model testing.

DATA

A single dataset was gathered from Kaggle which included 6 quantitative features and a single string target. There were 0 empty values and one record was removed as an outlier using Tukey's algorithm and visual comparison. The target values were encoded as 0 for Abnormal and 1 for Normal. The Normal targets consisted of ⅓ of the population numbering 100. The Abnormal population consisted of 209. All features had normal-ish distribution with the 'degree' feature skewed to the right. Setting alpha to 0.5 prior to hypothesis testing, statistical analysis proved significance between Normal and Abnormal targets did indeed exist proving viability of the project. Feature correlations over 90% would be removed, however, none met the limit although it should be noted 2 features were very close and could be considered for feature selection if additional tuning were performed. These were 'incidence' and 'tilt'. The 'degree' feature was most highly correlated with the target results which also contained the greatest degree of variance within the "abnormal" set, more the 2 times the variance of the "normal" portion. Cleaning and EDA concluded with the cleaned features 'X' and encoded target 'Y' values stored in csv files for easy upload into new notebooks by importing "X.csv", "Y.csv", and "df.csv" for the completed encoded set.

TUNING

A stratified train-test split began the modeling to address unbalanced data. To prevent overfitting KFold cross-validation was included within the GridSearch during parameter tuning. Upsampling using SMOTE and ADASYN for comparison were included in the model fit to balance the data during tuning. As you might expect, SMOTE provided slightly better performance on most models, most likely due to the higher variance of the minority class of the 'degree' feature.

With the importance of accurate results in healthcare, I searched for additional ways to provide a heavy bias with 100% accuracy in either class with minimal loss of accuracy. Application of a Linear Discriminant Transform on the feature data resulted in many models reporting such confidence. For confident, time-saving, sorting of results by class I would recommend application of this transform.

TESTING

Model accuracy was determined measuring ROC AUC, F1, and Recall scores with a confusion matrix plot. Most models performed similarly with the exception of Guassian Naive Bayes, possibly due to some feature correlations that existed within the data. If a NB model were preferred, additional feature selection could improve the accuracy, more testing is needed. Below is a summary of the data and test results. Note each model includes recall scores to determine best models based on bias selection.

Data Descriptions

- 6 quantitative Features
- 1 binomial Target of 209 Normal (0) and 100 Abnormal (1) labels
- alpha of 0.5 for hypothesis test
- Supervised learning
- Unbalanced dataset addressed with resampling using SMOTE
- K-Fold Cross-validation to prevent over-fitting
- Accuracy measured with F1 score, ROC AUC plots, and confusion matrix.

Models Tested

5 Discriminative

- ○ Logistic Regression

  - KNearest Neighbors
  - Random Forest
  - SVM
  - Gradient Boosting

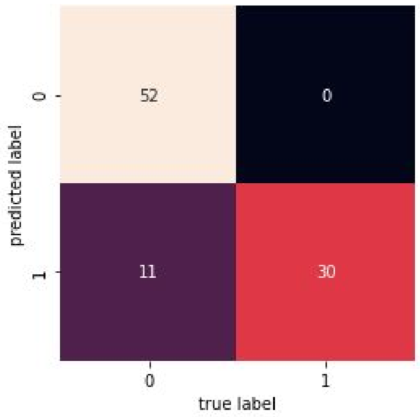
1 Generative

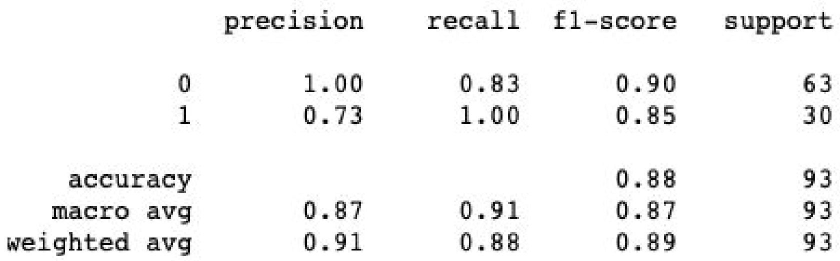
  - Naive Bayes

Results for top performer - Logistic Regression

Logistic Regression Results (SMOTE, LDA, C=0.001, penalty='l2', solver = 'newton-cg')

A heavy bias resulting in 100% recall of "Normal" classification



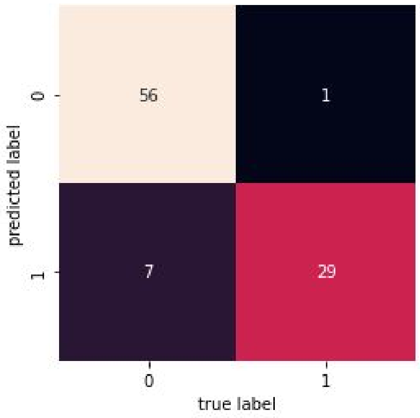


```
              precision    recall  f1-score   support

           0       1.00      0.83      0.90        63
           1       0.73      1.00      0.85        30

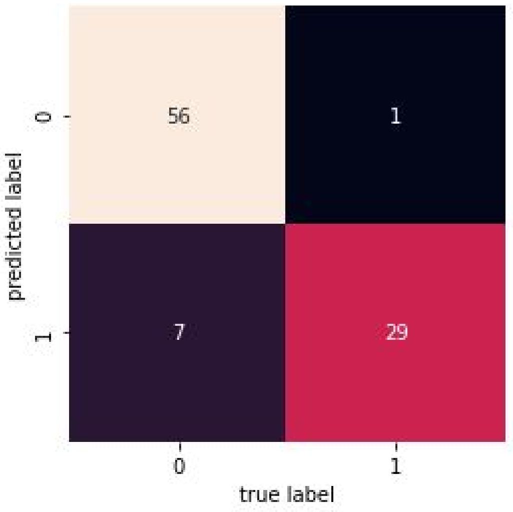
    accuracy                           0.88        93
   macro avg       0.87      0.91      0.87        93
weighted avg       0.91      0.88      0.89        93
```

Balanced bias - without the Linear transformation

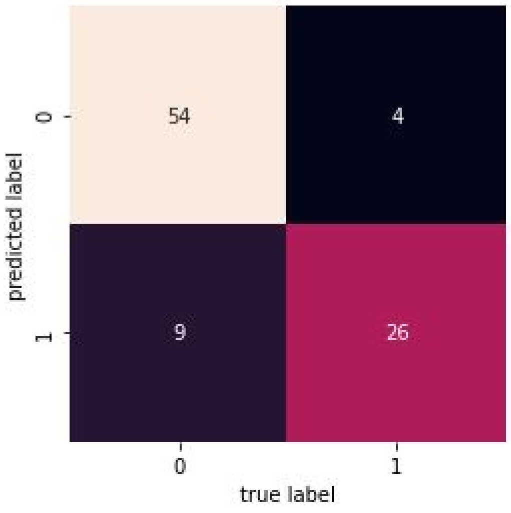


Results for all models without LDA transformation (more balanced bias)

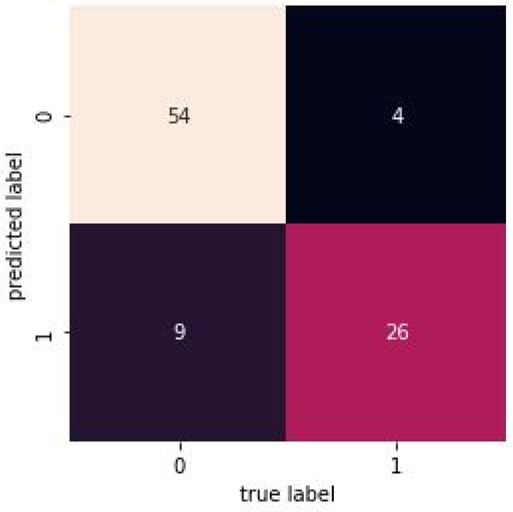
Logistic Regression



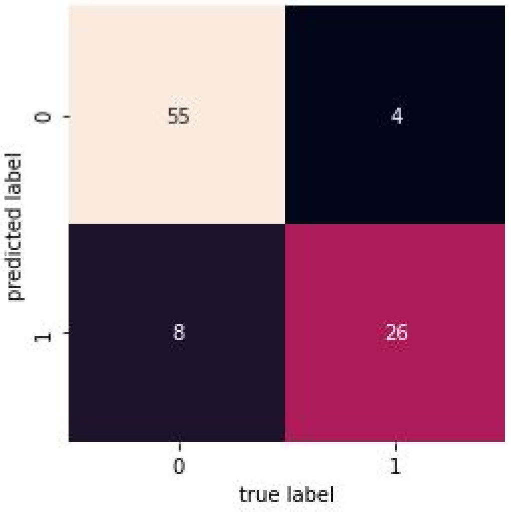
Gradient Boosting Classifier



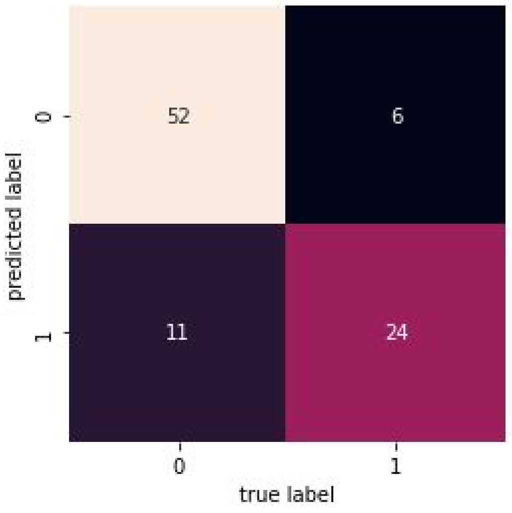
Support Vector Machine



Random Forest Classifier

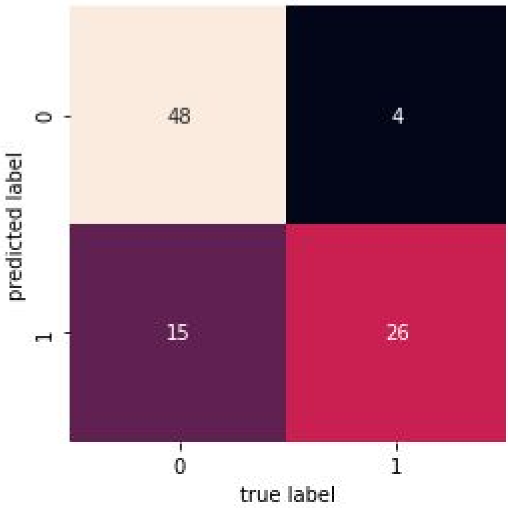


KNneighbors



Gaussian Naive Bayes

```
Best Parameters:  {'class__var_s
```

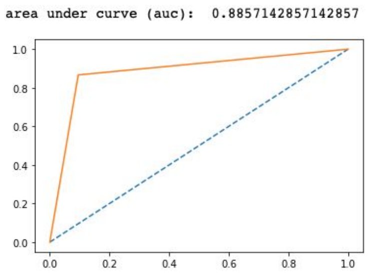


```
LogisticRegression(random_state=42)
              precision    recall  f1-score   support

           0       0.93      0.90      0.92        63
           1       0.81      0.87      0.84        30

    accuracy                           0.89        93
   macro avg       0.87      0.89      0.88        93
weighted avg       0.90      0.89      0.89        93

area under curve (auc):  0.8857142857142857
```

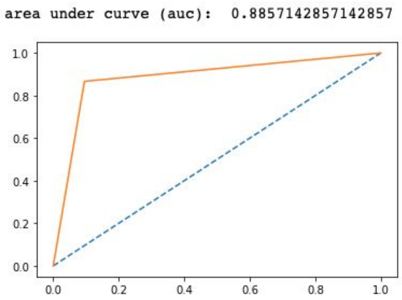

area under curve (auc): 0.8857142857142857

```
GradientBoostingClassifier(random_state=42)
              precision    recall  f1-score   support

           0       0.93      0.90      0.92        63
           1       0.81      0.87      0.84        30

    accuracy                           0.89        93
   macro avg       0.87      0.89      0.88        93
weighted avg       0.90      0.89      0.89        93

area under curve (auc):  0.8857142857142857
```

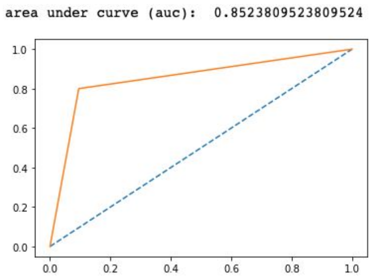

area under curve (auc): 0.8857142857142857

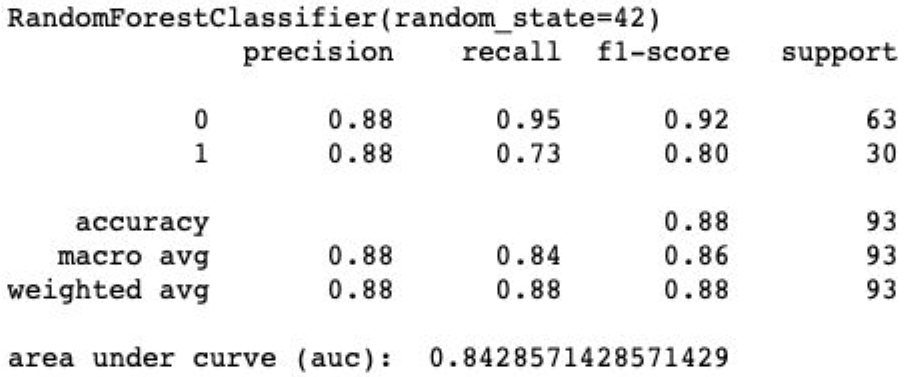
```
SVC(random_state=42)
              precision    recall  f1-score   support

           0       0.90      0.90      0.90        63
           1       0.80      0.80      0.80        30

    accuracy                           0.87        93
   macro avg       0.85      0.85      0.85        93
weighted avg       0.87      0.87      0.87        93

area under curve (auc):  0.8523809523809524
```


area under curve (auc): 0.8523809523809524

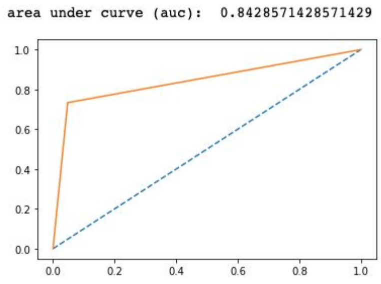


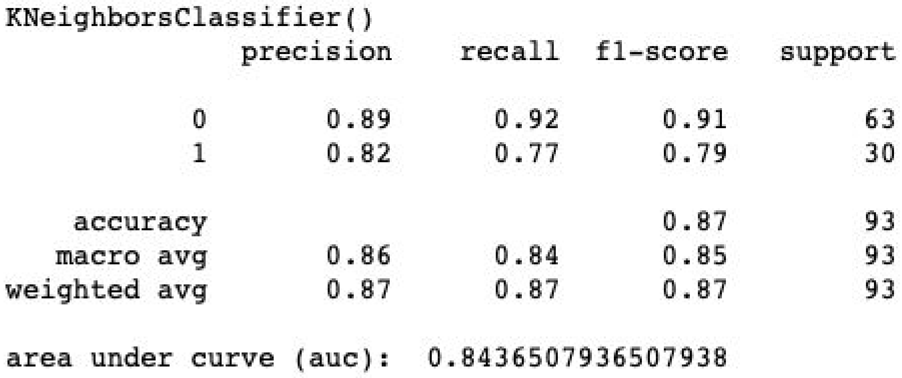
```
RandomForestClassifier(random_state=42)
              precision    recall  f1-score   support

           0       0.88      0.95      0.92        63
           1       0.88      0.73      0.80        30

    accuracy                           0.88        93
   macro avg       0.88      0.84      0.86        93
weighted avg       0.88      0.88      0.88        93

area under curve (auc):  0.8428571428571429
```



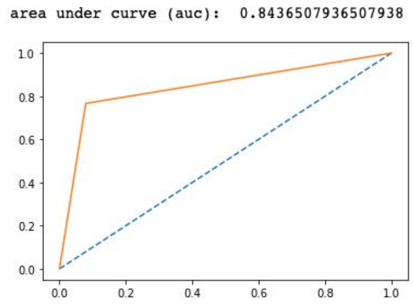


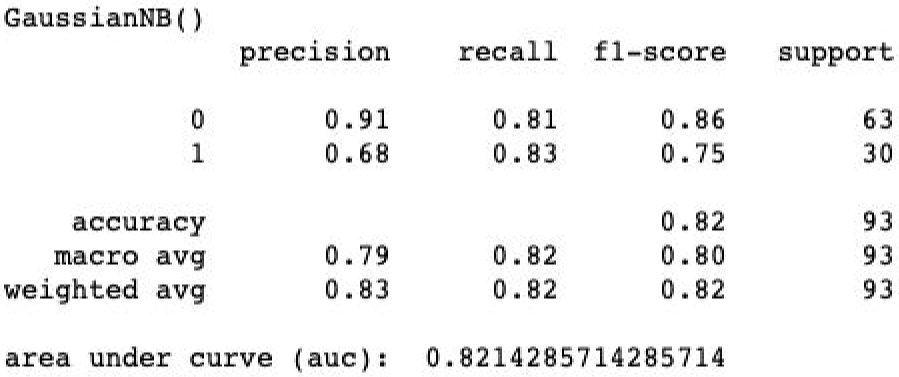
```
KNeighborsClassifier()
              precision    recall  f1-score   support

           0       0.89      0.92      0.91        63
           1       0.82      0.77      0.79        30

    accuracy                           0.87        93
   macro avg       0.86      0.84      0.85        93
weighted avg       0.87      0.87      0.87        93

area under curve (auc):  0.8436507936507938
```



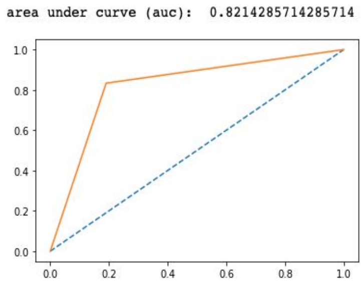


```
GaussianNB()
              precision    recall  f1-score   support

           0       0.91      0.81      0.86        63
           1       0.68      0.83      0.75        30

    accuracy                           0.82        93
   macro avg       0.79      0.82      0.80        93
weighted avg       0.83      0.82      0.82        93

area under curve (auc):  0.8214285714285714
```



With parameter tuning, both Logistic Regression and Gradient Boosting algorithms performed best, and identically: ROC AUC of 88.56%, an f1 score for normal: 84% and abnormal: 92% with only 11% of test data mislabeled.

SVM, RandomForest also performed well with f1 scores at or above 80% for all and ROC AUC averaging 84%.

KNeighbors and GaussianNB broke down below 80% for normal results.