Capstone 1: Orthopedic Biomechanical Features

by Michelle Ide

Machine Learning

DESCRIPTION

This supervised classification project uses quantitative biomechanical data taken from patient x-rays to predict results as either Normal or Abnormal. In this project abnormal indicates spondylolisthesis of the lumbar spine specifically.

DATA

Once cleaned, the data contained 309 records with unbalanced target values of 209 abnormal and 100 normal results. There are 6 quantitative features with no null values. All features passed hypothesis testing using an alpha of 0.05. It is recommended the features be scaled during testing to reduce range differences.

METHOD

Imbalanced data was addressed with a stratified train-test split followed by resampling methods. A test size of 30% was selected.

Statisfied train-test split for unbalanced dataset

```
]: X_train, X_test, y_train, y_test = train_test_split(X, Y, test_size=0.3,
                                                       stratify=Y,random_state = 88)    # 30% test set
```

Two resampling methods were tested, ADASYN and SMOTE, and were performed within the GridSearch during parameter tuning.

Parameter tuning was performed with a KFold split in a GridSearch to prevent overfitting. For a multi-featured, quantitative model, an accuracy measurement of F1-scoring was selected. The method below was used for tuning all models.

```
def gridSearchCV(model, param_grid, X_train, X_Test, y_train, y_test, graph = 1,name='none'):

    # Scale the features
    std_scale = StandardScaler()
    X_train_scaled = std_scale.fit_transform(X_train)
    X_test_scaled = std_scale.transform(X_test)

    # Fold parameters
    kf = KFold(n_splits=5, shuffle=False)

    # create pipeline
    resample = SMOTE(random_state=88)
    pipeline = Pipeline([('sampling', resample), ('class', model)])

    # perform gridsearch, fit, and predict
    grid = GridSearchCV(pipeline, param_grid, scoring = 'f1', cv = kf)
    grid.fit(X_train_scaled, y_train)
    print("Training score: ",clf.score(X_train, y_train))
    predictions = grid.predict(X_test_scaled)
```

Performance result details can be reviewed in the addendum at the end of this report and include a Confusion Matrix, ROC AUC along with a classification report that includes Recall and Precision.

MODELS

This is a supervised classification problem with a single binomial target. The following models were tested and compared for accuracy. 5 Discriminative and 1 Generative (Naive Bayes)

- Logistic Regression
- KNearest Neighbors
- Random Forest
- SVM
- Gradient Boost
- Naive Bayes (Generative)

RESULTS

A summary of accuracy scores below compare models using f1 as the scoring method:

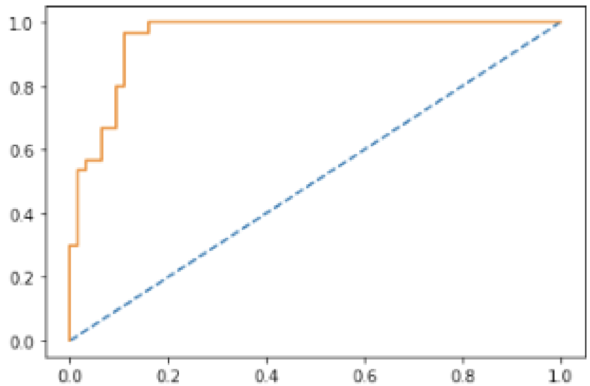
	SMOTE	ADASYN
Model	**Score**	***Model***
logreg	87.88%	85.71%
BGC	80.00%	73.02%

SVM	80.00%	77.61%
RandomForest	81.25%	81.16%
KNN	73.85%	73.85%
Naive Bayes	73.24%	72.22%

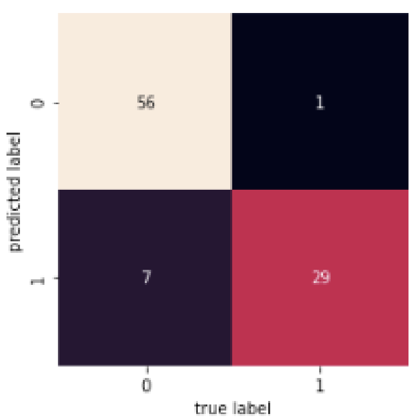
Logistic Regression provided the best performance for both resampling methods. With a recall rate of 89 for normal and 97 for abnormal values, using a SMOTE resampling method, if a total of 100 abnormal results existed, we can expect 97 of these to be properly identified, for 100 normal results 89 would be found. ADASYN's performance was similar with a recall of 100 for abnormal and 84 for normal. Below is a summary of the Logistic Regression scores and confusion matrix.

SMOTE

```
area under curve (auc):  0.9523809523809523
```

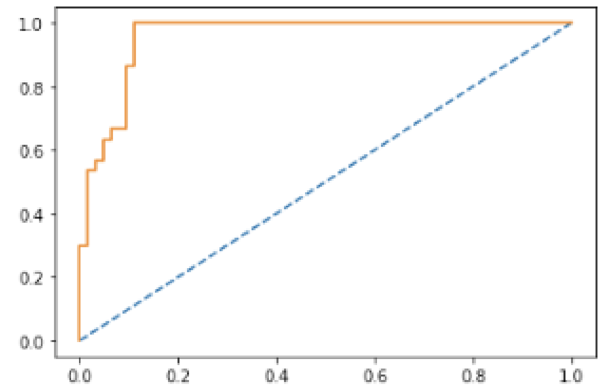



```
              precision    recall  f1-score   support

           0       0.98      0.89      0.93        63
           1       0.81      0.97      0.88        30

    accuracy                           0.91        93
   macro avg       0.89      0.93      0.91        93
weighted avg       0.93      0.91      0.92        93

Best Parameters:  {'class__C': 10, 'class__penalty': 'l2', 'class__solver': 'newton-cg'}
```

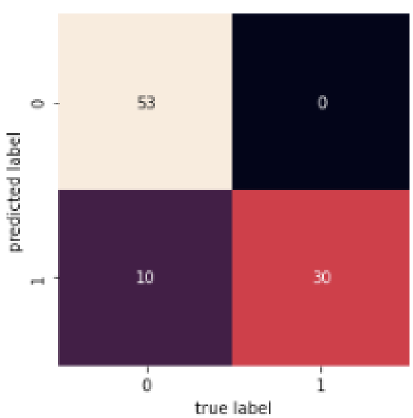


ADASYN

```
area under curve (auc):  0.9560846560846561
```




```
              precision    recall  f1-score   support

           0       1.00      0.84      0.91        63
           1       0.75      1.00      0.86        30

    accuracy                           0.89        93
   macro avg       0.88      0.92      0.89        93
weighted avg       0.92      0.89      0.90        93

Best Parameters:  {'class__C': 10, 'class__penalty': 'l2', 'class__solver': 'newton-cg'}
```



Detailed results are contained in the addendum at the end of this report.

ADDENDUM

Summary of Accuracy Scores for both resampling methods:

SMOTE

	ROC AUC	Recall - Abnormal	Recall - Normal	Precision - Abnormal	Precision - Normal	Weighted Average - Precision
Logistic Regression	95%	89%	97%	98%	81%	93%
Gadient Boosting	96%	86%	87%	93%	74%	87%
SVM	NA	86%	87%	93%	74%	87%
Random Forest	95%	87%	87%	93%	76%	88%
KNeighbors	89%	83%	80%	90%	69%	83%
Gaussian NB	89%	76%	87%	92%	63%	83%

ADASYN

	ROC AUC	Recall - Abnormal	Recall - Normal	Precision - Abnormal	Precision - Normal	Weighted Average - Precision
Logistic Regression	96%	84%	100%	100%	75%	92%
Gadient Boosting	94%	84%	77%	88%	70%	82%
SVM	N/A	93%	70%	83%	87%	86%
Random Forest	95%	83%	93%	96%	72%	88%
KNeighbors	81%	83%	80%	90%	69%	83%
Gaussian NB	86%	75%	87%	92%	62%	82%

CONFUSION MATRIX SUMMARY

	SMOTE	ABN	NOR		**ADASYN**	ABN	NOR
LR	ABN	56	1	LR	ABN	53	0
	NOR	7	29		NOR	10	30
GB	ABN	54	4	GB	ABN	53	7
	NOR	9	26		NOR	10	23
SVM	ABN	54	4	SVM	ABN	52	4
	NOR	9	26		NOR	11	26
RF	ABN	55	4	RF	ABN	52	2
	NOR	8	26		NOR	11	28
KNN	ABN	52	6	KNN	ABN	52	6
	NOR	11	24		NOR	11	24
NB	ABN	48	4	NB	ABN	47	4
	NOR	15	26		NOR	16	26

CORRECTLY LABELED