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

Detailed results for all models with both resampling methods are contained in the addendum at the end of this report.

Both sampling techniques performed similarly with Logistic Regression demonstrating the best performance. However, for weighting results to prevent false negatives, ADASYN resampling would provide better results.

Using SMOTE resampling, Logistic Regression had a 95% ROC AUC and Recall of 97% for Normal and 89% for Abnormal results. Therefore, 97 out of 100 normal results were accurately found and 89 out of every 100 abnormal results were found and properly classified.

With ADASYN, the Logistic Regression recall of 100% for Normal and 84% Abnormal provides important accuracy by determining 100 out of 100 normal results.

Of the remaining models, Random Forest was a close second to Logistic Regress however it performed similarly to SVM, and Gradient Boosting.

KNNeighbors and Naive Bayes are not competitive with the other models and not recommended for this purpose.

ADDENDUM

Summary of Accuracy Scores for both resampling methods:

SMOTE

	Trianing Score	ROC AUC	Recall - Abnormal	Recall - Normal	Precision - Abnormal	Precision - Normal	Weighted Average - Precision
Logistic Regression	63%	95%	89%	97%	98%	81%	93%
Gadient Boosting	41%%	96%	86%	87%	93%	74%	87%
SVM	0%	NA	86%	87%	93%	74%	87%
Random Forest	34%	95%	87%	87%	93%	76%	88%
KNeighbors	0%	89%	83%	80%	90%	69%	83%
Gaussian NB	0%	89%	76%	87%	92%	63%	83%

ADASYN

	Trianing Score	ROC AUC	Recall - Abnormal	Recall - Normal	Precision - Abnormal	Precision - Normal	Weighted Average - Precision
Logistic Regression	61%	96%	84%	100%	100%	75%	92%
Gadient Boosting	36%	94%	84%	77%	88%	70%	82%
SVM	0%	N/A	93%	70%	83%	87%	86%
Random Forest	5%	95%	83%	93%	96%	72%	88%
KNeighbors	0%	81%	83%	80%	90%	69%	83%
Gaussian NB	0%	86%	75%	87%	92%	62%	82%

CONFUSION MATRIX SUMMARY

SMOTE

		ABN	NOR
LR	ABN	56	1
	NOR	7	79
GB	ABN	54	4
	NOR	9	26
SVM	ABN	54	4
	NOR	9	26
RF	ABN	55	4
	NOR	8	26
KNN	ABN	52	6
	NOR	11	24
NB	ABN	48	4
	NOR	15	26

ADASYN

		ABN	NOR
LR	ABN	53	0
	NOR	10	30
GB	ABN	53	7
	NOR	10	23
SVM	ABN	52	4
	NOR	11	26
RF	ABN	52	2
	NOR	11	28
KNN	ABN	52	6
	NOR	11	24
NB	ABN	47	4
	NOR	16	26