Case Study 1

Hammad Khan 2584665K
Sarvesh Tendulkar 2582865T
Sebin Thomas 2604952T

Introduction

This case study presents the problem of predicting Central Neuropathic Pain (CNP) in people with Spinal Cord Injury (SCI) with the help of Electroencephalogram (EEG) data.
CNP is a neurological disorder caused by damage to the sensory pathways of the Central Nervous System[1] (CNS), example of which is Spinal Cord Injury (SCI). It is estimated that approximately 50% of the people who have SCI develop CNP at a later stage[2].
Generally, due to lack of medical treatment after diagnosis, only preventative medication is used which has strong side effects but predicting whether a patient would be diagnosed with CNP at a later stage would help in selective treatment. It is found that the EEG data has characteristic markers that could help in predicting whether a person with SCI would later develop CNP solely based on EEG data[2].
The EEG data presented in this case study is collected using a 48 electrode EEG device of 18 subjects, 10 of which developed CNP within 6 months of data collection while the remaining 8 subjects did not.
The device captured alpha, beta, and theta band powers while eyes closed, and eyes opened of the subject. These values along with the ratio of eyes-opened to eyes-closed formed the 9 features of each electrode resulting in a 432 high-dimensional dataset.

The task of this case study is to implement a feature engineering strategy which optimizes the prediction accuracy of whether a patient is likely to develop pain based on their EEG data.

Methods

The general idea is to eliminate the features that do not contribute significantly to the classifier's decision-making process. Out of 432 features (D) find a subset M, such that M << D, which on average significantly improves the performance of the classifier, specifically in terms of Leave one subject out cross-validation accuracy, sensitivity, specificity, and AUC.

Although the data is pre-processed, centering and scaling the data removes classifier bias amongst the features and markedly improves the performance even before feature selection methods are applied. Feature Selection methods namely, Recursive Feature Elimination, ANOVA, Mutual Information and Embedded Method using Lasso are used to find optimal features to train Logistic Regression and Linear SVM models.

Wrapper Methods:

Recursive Feature Elimination with Cross Validation:

RFECV recursively eliminates a number of features based on their average ranking across all validation folds. In RFECV, the cross-validation finds the best number of features to select which is then used by the underlying RFE algorithm to recursively remove features until the best number of features is achieved[3].

The important parameters to tune for RFECV are the estimator to be used to fit the data for feature selection, this must be a supervised estimator that by default provides information on feature importance, and the scoring method to be used while selecting the best features.

Filter Methods:

Filtering methods work by selecting the K best features based on relevance calculated by statistical tests. We tried multiple univariate statistical tests for feature selection which are defined below. After getting relevance scores from statistical tests, we use the SelectKbest method to select top K relevant features. K is decided using cross-validation.

ANOVA F-test:

ANOVA (analysis of variance) uses F-test to measure the equality of means when we have three or more groups. We can understand the F-scores as a measure of how informative each feature is for our dataset. F-test is carried out to assess each feature[7] meaning this is a univariate feature selection method. The F-scores are the test statistic for the F-test, and they basically represent the ratio between the explained and the unexplained variance[8].

Mutual Information:

Mutual information (MI) is a non-negative value between two random variables, It tells us how much a given feature can explain our target variable, or more technically, how much information about the target variable will be obtained by having observed a feature. Mutual info is also a univariate feature selection method. It is equal to zero if and only if two random variables are independent, and higher values mean higher dependency[5].

Chi-Squared test:

Chi-squared is a univariate statistic used to measure the correlation between two random variables. It is defined as the sum of the squares of the normalized difference between observed and expected values. In this context, the chi-squared test returns a set of p-values which shows whether features are independent of the labels[9].
Grid Search is used to determine the optimal number of features to keep using the chi-squared test.

Embedded Method (Lasso):

Embedded methods use algorithms that have built-in feature selection methods. For e.g., algorithms that use L1 (Lasso regularization) or tree-based algorithms that has built-in feature importance (Random Forest).
In Lasso penalization for regularization, many coefficients will be 0 (or close to 0) which are not important in terms of prediction, and we select the features with non-zero coefficients meaning only important features which are contributing to discriminate between our target classes. In Lasso alpha is an important parameter to tune, the higher the alpha parameter, the fewer features are selected[7].

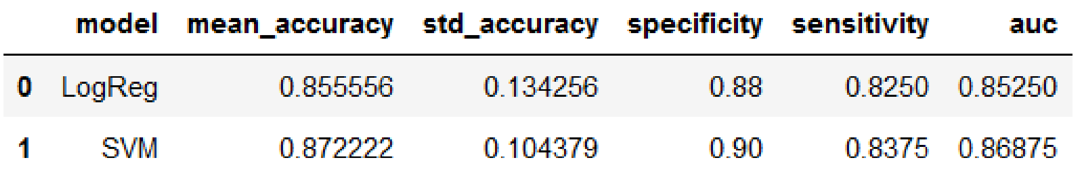
Results

The given data was collected from a total of 18 subjects with 10 repetitions amounting to 180 rows of sorted by subject-number basis. In this case study, one of the objectives was to perform Leave one subject out cross validation method, to achieve this K Fold cross validation technique was used to split the data into 18 folds such that each fold leaves exactly one subject out from the training set, thus achieving the leave one subject out prediction accuracy.
In each iteration of the fold, before fitting the data to the model, train-set and test-set were independently standardized using StandardScaler technique and then the model performance was evaluated.

The purpose of this case study was to devise a feature selection strategy that gives optimal performance for the chosen model and to best achieve that we performed a comparative study on the feature selection techniques that were introduced through this course, mainly three types of techniques: filter, wrapper and embedded. The rationale behind performing these experiments was to implement various types of feature selection techniques and compare the results in an Iterative process that ultimately selected the method that provided the most optimal model performance.

After transforming and splitting the data different feature selection strategies were applied only on the training set of the data. We primarily measured performance of the model on four parameters namely, accuracy, sensitivity, specificity, and area under the curve (AUC); where for accuracy, mean and standard deviation are calculated across each fold whereas, for the other 3 parameters only one value is obtained per model due to Leave one subject out cross validation method. The performance was evaluated for Logistic Regression and Linear Support Vector Machine (SVM) models, and the results of the experiment are detailed below. In comparing the model performance, the primary parameter we focused on is AUC, as it is generally a good metric to judge a model's performance based on AUC.

Baseline Score on Raw Data-
To set the baseline score and judge whether the data transformation and feature selection techniques are providing a significant improvement in terms of model performance, we set a baseline score for both Logistic Regression and SVM models as noted below. The baseline AUC obtained was 0.85 and 0.86 for Logistic Regression and SVM, respectively.



	model	mean_accuracy	std_accuracy	specificity	sensitivity	auc
0	LogReg	0.855556	0.134256	0.88	0.8250	0.85250
1	SVM	0.872222	0.104379	0.90	0.8375	0.86875

Table 1: Baseline Score on Raw Data

Baseline Score on Scaled Data-

After obtaining a baseline score on raw data, we measured the model performance on scaled data to justify whether further pre-processing is necessary on the data, as the given data was already pre-processed. Both the classifiers performed significantly better on scaled data as compared to the baseline scores.

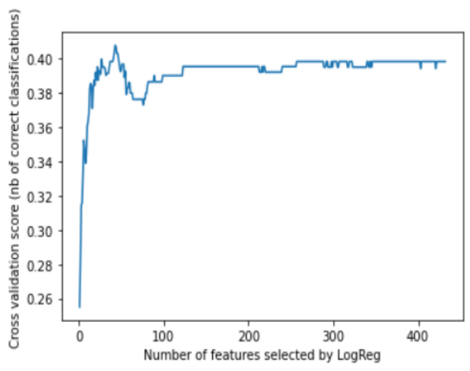
	model	mean_accuracy	std_accuracy	specificity	sensitivity	auc
0	LogReg	0.933333	0.081650	0.94	0.925	0.9325
1	SVM	0.900000	0.094281	0.90	0.900	0.9000

Table 2: Baseline Score on Scaled Data

RFECV on Scaled Data-

After setting up baseline scores, the first experiment was based on a wrapper method using recursive feature elimination with cross validation. The AUC score could not be optimized for each fold hence, we tested two scoring functions, F1 and Accuracy score, and compared the results.

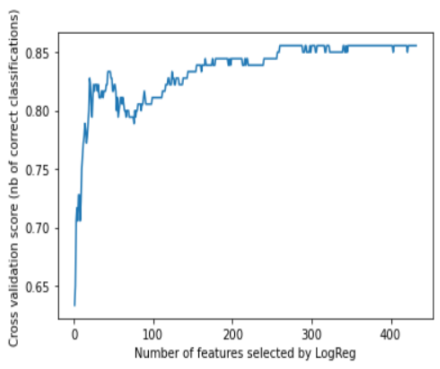
```
Optimal number of features : 43
```



	model	mean_accuracy	std_accuracy	sensitivity	specificity	auc
0	LogReg	0.916667	0.083333	0.925	0.91	0.9175

Fig 1 (a): Logistic Regression Scoring ‘F1’

```
Optimal number of features : 260
```

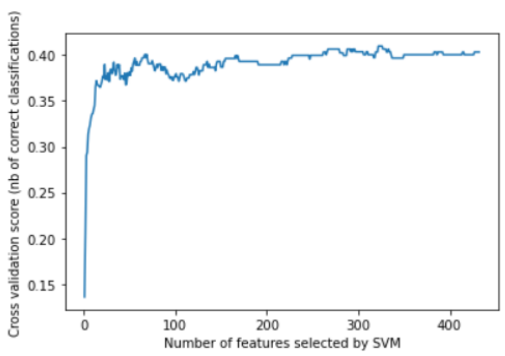


	model	mean_accuracy	std_accuracy	sensitivity	specificity	auc
0	LogReg	0.927778	0.080316	0.9125	0.94	0.92625

Fig 1 (b): Logistic Regression Scoring ‘Accuracy’

When we used F1 as a scoring function in Fig 1 (a), the number of features decreased drastically to just 43 but AUC dropped a bit from the scaled baseline. In Fig 1 (b), Accuracy has a better AUC score than F1 however, it is less than the baseline score with significantly more features selected.

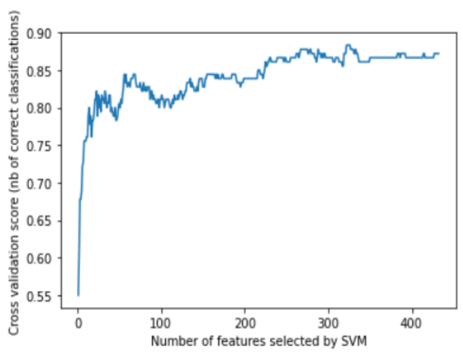
```
Optimal number of features : 322
```



	model	mean_accuracy	std_accuracy	sensitivity	specificity	auc
0	SVM	0.916667	0.095743	0.8875	0.94	0.91375

Fig 2 (a): SVM Scoring ‘F1’

```
Optimal number of features : 322
```



	model	mean_accuracy	std_accuracy	sensitivity	specificity	auc
0	SVM	0.916667	0.095743	0.8875	0.94	0.91375

Fig 2 (b): SVM Scoring ‘Accuracy’

For Linear SVM there was no difference in the performance; both scoring methods gave an AUC score of 0.91 better than baseline score with only 322 features selected.

ANOVA on Scaled Data-

Next, we tried filter methods to perform feature selection on scaled data. Like RFECV we tried two scoring methods F1 and accuracy for ANOVA based filtering with Logistic Regression and Linear SVM models.

```
Number of feature selected:  369
```

	model	mean_accuracy	std_accuracy	sensitivity	specificity	auc
0	LogReg	0.938889	0.089062	0.9125	0.96	0.93625

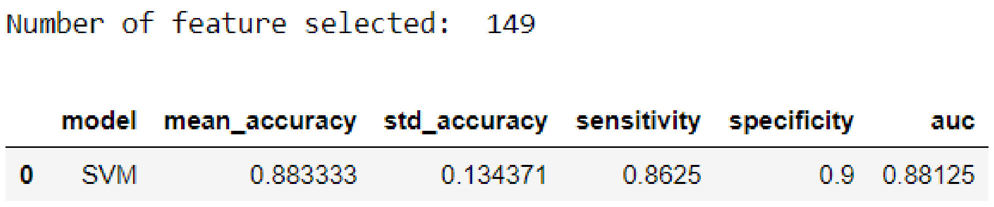
Table 3 (a): Logistic Regression Scoring ‘F1’

```
Number of feature selected:  182
```

	model	mean_accuracy	std_accuracy	sensitivity	specificity	auc
0	LogReg	0.911111	0.093624	0.875	0.94	0.9075

Table 3 (b): Logistic Regression Scoring ‘Accuracy’

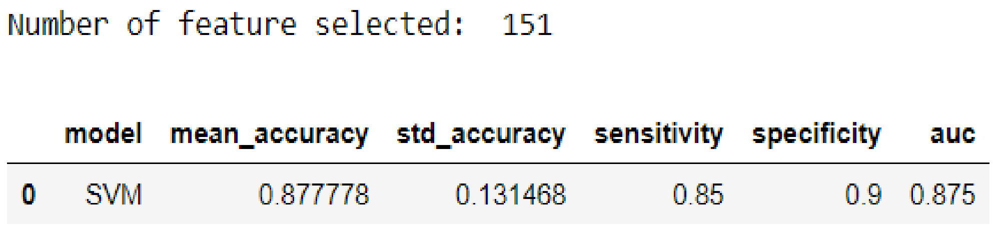
Similar to RFECV, the number of features selected by F1 scoring (Table 3 (a)) is more than that by accuracy method as detailed in Table 3 (b). In terms of performance, F1 scoring has a slight improvement over the baseline scores however, the mean_accuracy has decreased.



```
Number of feature selected:  149
```

	model	mean_accuracy	std_accuracy	sensitivity	specificity	auc
0	SVM	0.883333	0.134371	0.8625	0.9	0.88125

Table 4 (a): SVM Scoring ‘F1’



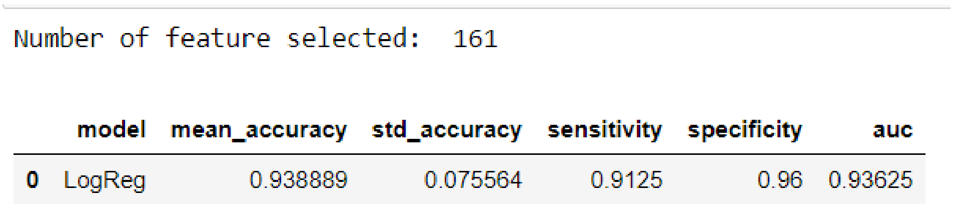
```
Number of feature selected:  151
```

	model	mean_accuracy	std_accuracy	sensitivity	specificity	auc
0	SVM	0.877778	0.131468	0.85	0.9	0.875

Table 4 (b): SVM Scoring ‘Accuracy’

In the case of SVM, there is little difference in terms of selected features for both scoring methods as can be observed in the tables 4 (a) and (b) but the performance of AUC is still less than the baseline scores.

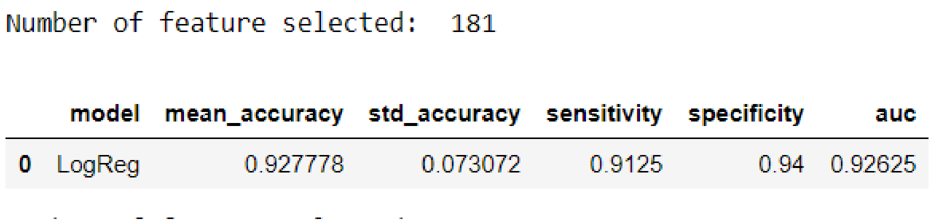
Mutual Information on Scaled Data-
The second filtering method that we investigated was the Mutual Information filter method which gave us the best performance on the AUC metric for both the Logistic Regression and Linear SVM models.



```
Number of feature selected:  161
```

	model	mean_accuracy	std_accuracy	sensitivity	specificity	auc
0	LogReg	0.938889	0.075564	0.9125	0.96	0.93625

Table 5 (a): Logistic Regression Scoring ‘F1’

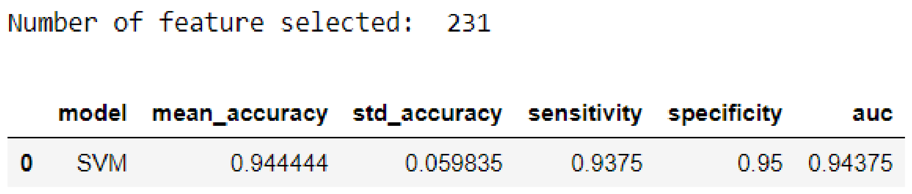


```
Number of feature selected:  181
```

	model	mean_accuracy	std_accuracy	sensitivity	specificity	auc
0	LogReg	0.927778	0.073072	0.9125	0.94	0.92625

Table 5 (b): Logistic Regression Scoring ‘Accuracy’

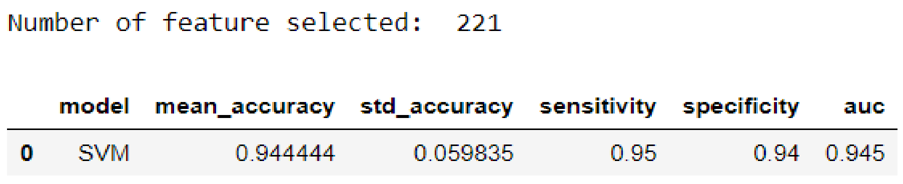
As can be observed from the above tables, the best AUC on the Logistic Regression model is **0.93625** same as ANOVA using F1 score but in this case, only using 161 features are used to get this performance, while in the case of ANOVA 369 features were selected. Reducing the number of features from 432 to 161 features with increased performance is a significant improvement for the model as it will be easier to interpret, and it can also be observed that there is a decrease in std_accuracy across folds meaning the model is more robust now.



```
Number of feature selected:  231
```

	model	mean_accuracy	std_accuracy	sensitivity	specificity	auc
0	SVM	0.944444	0.059835	0.9375	0.95	0.94375

Table 6 (a) SVM Scoring ‘F1’



```
Number of feature selected:  221
```

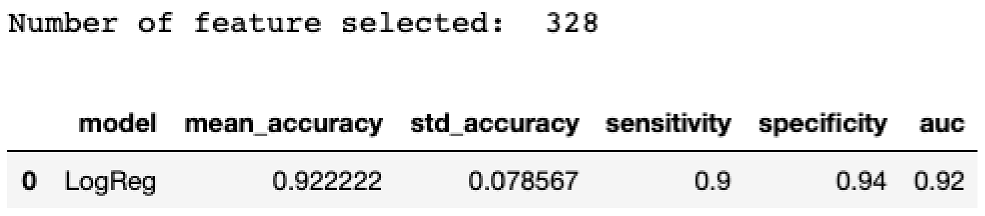
	model	mean_accuracy	std_accuracy	sensitivity	specificity	auc
0	SVM	0.944444	0.059835	0.95	0.94	0.945

Table 6 (b): SVM Scoring ‘Accuracy’

Similar trends can be observed with SVM models as well. In Table 6 (b), Mutual Information with accuracy as a scoring method resulted in the highest AUC score of **0.945** as compared to other feature selection techniques and models. In this case, a total of 221 features were selected out of 432 with lowest std_accuracy.

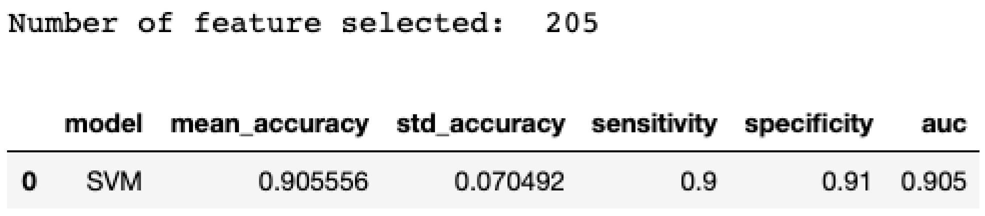
Chi-Squared test on Scaled Data-

We then performed Chi-squared tests over the scaled data and calculated scores using accuracy and F1 methods. The tests were carried out on Logistic regression and SVM models.



```
Number of feature selected:  328
```

	model	mean_accuracy	std_accuracy	sensitivity	specificity	auc
0	LogReg	0.922222	0.078567	0.9	0.94	0.92

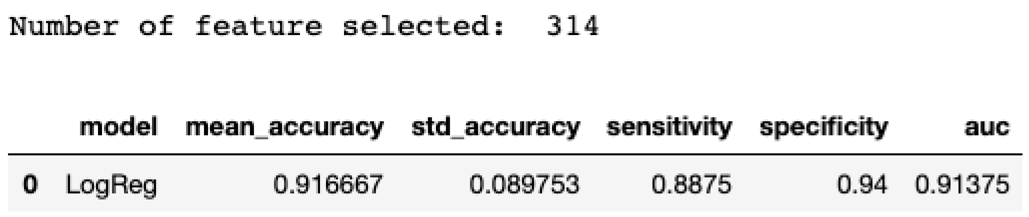


```
Number of feature selected:  205
```

	model	mean_accuracy	std_accuracy	sensitivity	specificity	auc
0	SVM	0.905556	0.070492	0.9	0.91	0.905

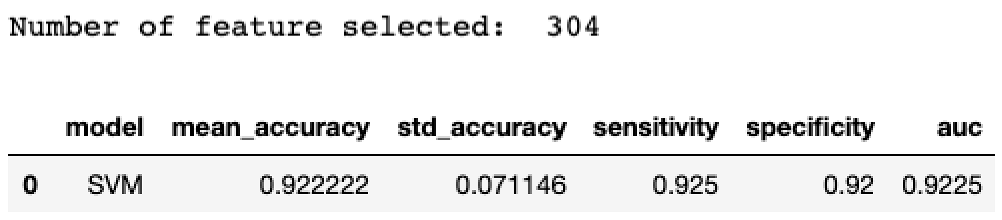
Table 7 (a): Model performances for Chi-squared test on scoring ‘Accuracy’

Although AUC score for Logistic regression is comparable to that of ANOVA, the number of features it used to get the same score is much more than ANOVA. SVM performs similar to ANOVA but used much more features.



```
Number of feature selected:  314
```

	model	mean_accuracy	std_accuracy	sensitivity	specificity	auc
0	LogReg	0.916667	0.089753	0.8875	0.94	0.91375



```
Number of feature selected:  304
```

	model	mean_accuracy	std_accuracy	sensitivity	specificity	auc
0	SVM	0.922222	0.071146	0.925	0.92	0.9225

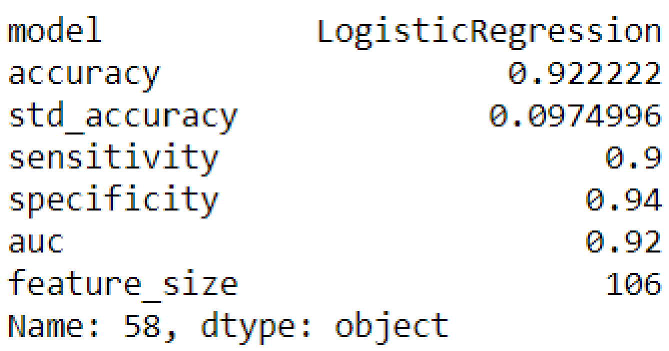
Table 7 (b): Model performances for Chi-squared test on scoring ‘F1’

Under the F1 method, mean accuracy score for Logistic regression is slightly lower than the accuracy scoring method. SVM manages to perform better under F1 compared to the accuracy method and outperforms ANOVA but uses significantly more features compared to the other filter methods used.

Embedded Method (Lasso) on Scaled Data-

Finally, we experimented with an Embedded method using LASSO or L1 regularized model for feature selection. As was the case with the previous experiments we tried this technique on Logistic Regression and Linear SVM models for consistency.

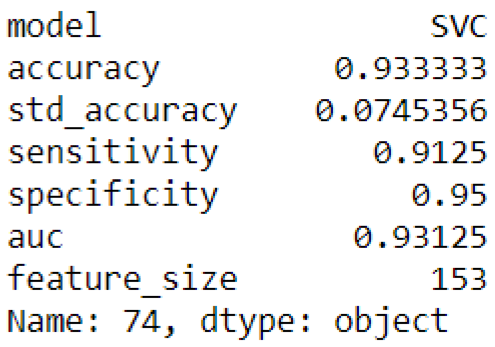
It was a little tricky to perform this experiment because we had to grid search over values of alpha parameter for Lasso as it has quite an impact on the performance of the model. Same alpha cannot be optimized for both the models. We first fit the scaled data on LassoCV() which returned some alpha values that we used to perform grid search over. Next, we implemented a custom Lasso Transformer which would facilitate pipelining of scaling, fitting the Lasso model on one of the alpha values, then filtering features based on non-zero coefficient and finally fitting one of the models on a selected feature set for each fold.



```
model           LogisticRegression
accuracy                  0.922222
std_accuracy             0.0974996
sensitivity                    0.9
specificity                   0.94
auc                           0.92
feature_size                   106
Name: 58, dtype: object
```

Fig 3 (a): Logistic Regression Lasso

In terms of performance, Logistic Regression with Lasso is generally worse than the scaled baseline score, but it selected only 106 features resulting in a 0.92 AUC score.



```
model                SVC
accuracy        0.933333
std_accuracy   0.0745356
sensitivity       0.9125
specificity         0.95
auc              0.93125
feature_size         153
Name: 74, dtype: object
```

Fig 3 (b): SVM Lasso

Overall, the performance of SVM is better than the baseline score but not as good as the results obtained by Mutual Information however, it is only using 153 features to get 0.93125 AUC which is quite good and better than Logistic Regression with Lasso.

Discussion

The different feature selection methods that were explored in this case study depicted a consistent increase over the raw data baseline score bolstering the idea that there is indeed merit in implementing some data transformation and feature engineering pipeline before feeding the data to a classifier and it is worth the time that is spent on implementing them especially when there is very high dimensional data in the play. Starting off with an AUC baseline score of 0.85 and 0.86 for Logistic Regression and Linear SVM respectively, determined on the raw data, there was steady increase in model performance across scaling the data, performing Mutual Information strategy on scaled data and Lasso embedded technique. However, RFECV and ANOVA techniques on scaled data did not show promising results in terms of model performance as compared to the scaled baseline performance.

Mutual Information feature selection method with scaled data on the Linear SVM model was the best performing model with an AUC score of 0.945 and Leave one subject out accuracy of 94%. As Mutual Information can capture any kind of dependency between variables, compared to other feature selection techniques which can only capture linear dependency, it could better select features that had a non-linear dependency with the output variable and contributed more towards the output something which other techniques would be incapable to do due to their limitations. Consequently, Mutual Information selected the best 221 features that were most discriminative towards the output; achieving an accuracy score of 94.44%.

The most discriminative feature selected by Mutual Information 'ratio_beta_5' did not even come up in the top 10 features selected by ANOVA, which consequently reflects on the classifier performance.

References

1. National Organization for Rare Disorders [Internet]. [place unknown]: National Order for Rare Disorders; c2021. Central Pain Syndrome; 2021 [cited 2021 Mar 28]. Available from: https://rarediseases.org/rare-diseases/central-pain-syndrome/

2. Spinal Research [Internet]. [place unknown]: Spinal Research; c2021. Spinal Research announces two new research projects with Stoke Mandeville; 2020 June [cited 2021 Mar 28]. Available from: https://spinal-research.org/june-2020-spinal-research-announces-two-new-research-projects-stoke-mandeville

3. Scikit-learn: Machine Learning in Python [Internet]. [place unknown]: scikit-learn; c2020. sklearn.feature_selection.RFECV; 2020 [cited 2021 Mar 28]. Available from: https://scikit-learn.org/stable/modules/generated/sklearn.feature_selection.RFECV.html

4. Scikit-learn: Machine Learning in Python [Internet]. [place unknown]: scikit-learn; c2020. Univariate feature selection; 2020 [cited 2021 Mar 28]. Available from: https://scikit-learn.org/stable/modules/feature_selection.html#univariate-feature-selection

5. Scikit-learn: Machine Learning in Python [Internet]. [place unknown]: scikit-learn; c2020. sklearn.feature_selection.mutual_info_classif; 2020 [cited 2021 Mar 28]. Available from: https://scikit-learn.org/stable/modules/generated/sklearn.feature_selection.mutual_info_classif.html

6. DataDrivenInvestor [Internet]. [place unknown]: DataDrivenInvestor; c2020. Feature Selection Techniques; 2020 Jul 14 [cited 2021 Mar 28]. Available from: https://medium.datadriveninvestor.com/feature-selection-techniques-1a99e61da222

7. Statistics by Jim [Internet]. [place unknown]: Statistics by Jim; c2021. How F-tests work in Analysis of Variance (ANOVA); 2017 [cited 2021 Mar 28]. Available from: https://statisticsbyjim.com/anova/f-tests-anova/

8. Stack Exchange [Internet]. [place unknown]: Stack Exchange Inc; c2021. How to understand ANOVA-F for feature selection in Python. Sklearn SelectKBest with f_classif; 2020 May 19 [cited 2021 Mar 28]. Available from: https://datascience.stackexchange.com/questions/74465/how-to-understand-anova-f-for-feature-selection-in-python-sklearn-selectkbest-w

9. William G. Cochran. "The $\chi 2$ Test of Goodness of Fit." Ann. Math. Statist. 23 (3) 315 - 345, September, 1952. https://doi.org/10.1214/aoms/1177729380