

# Case Study 2

## Feature Engineering

Group 104

Garima Ashok Devnani (2653434D)

Pooja Anil Kurup (2708790K)

Rajnish Kumar Verma (2694285V)

Tejas Kundu (2647799K)

# Contents

Table of Figures .....	3
Chapter 1     Introduction .....	4
1.1     Central Neuropathic Pain .....	4
1.2     Electroencephalogram (ECG) .....	4
1.3     Data Set .....	4
1.3.1     Participation .....	4
1.3.2     Data Collection .....	5
Chapter 2     Methodology .....	5
2.1     Feature Engineering .....	5
2.2     Feature Selection .....	5
2.2.1     Wrapper Method .....	5
2.2.2     Embedded Method .....	5
2.3     Classification .....	6
2.3.1     Support Vector Machine .....	6
2.3.2     Logistic Regression .....	6
2.4     Cross-Validation .....	6
2.4.1     Leave-One-Subject Cross-Validation .....	7
2.4.2     K-Fold Cross-Validation .....	7
2.5     Steps for Implementation .....	7
2.6     Performance Metrics .....	7
Chapter 3     Results .....	9
3.1     No Feature Engineering .....	9
3.2     Embedded Method .....	9
3.3     Wrapper Method .....	9
Chapter 4     Discussion .....	10
References .....	11
Appendix .....	12

# Table of Figures

1. Feature Engineering
2. Wrapper Method
3. Embedded Method
4. SVM
5. Confusion Matrix
6. Performance measures when no feature selection is applied
7. Performance Score for each fold using Logistic Regression Classifier
8. Performance Score for each fold using Linear SVM Classifier
9. Performance Score for each classifier when no features are selected
10. Performance measures when Embedded (L1 Regularization) method is applied
11. Performance Score for each fold using Lasso (L1) Feature Selection and Logistic Regression Classifier
12. Performance Score for each fold using Lasso (L1) Feature Selection and Linear SVM Classifier
13. Performance Score for each classifier using Lasso (L1) Feature Selection
14. Performance measures when Wrapper (RFE) method is applied
15. Performance Scores for each fold using RFE Feature Selection and Logistic Regression Classifier
16. Performance Score for each fold using Lasso RFE Feature Selection and Linear SVM Classifier.
17. Performance Scores for each Classifier using RFE Feature Selection

# Chapter 1 Introduction

The purpose of this case study is to describe a feature engineering strategy and provide evidence for why the chosen technique is superior to an alternative strategy in Feature engineering. Predicting whether someone may have Central Neuropathic Pain with Spinal Cord injury using Electroencephalogram data.

## 1.1 Central Neuropathic Pain

"Central neuropathic pain" is a phrase used to describe pain that is originated or induced by a primary lesion or malfunction in the central nervous system, and it is a subset of "neuropathic pain". Neuropathic pain differs from nociceptive pain (non-neuronal tissue injury), which occurs when receptors are sensitive to tissue damage (nociceptors) and are stimulated by a suitable stimulus. Neuropathic pain is a type of chronic pain that is consistently created and serves no useful purpose for the person experiencing it. Although several treatments, including pharmacological and nonpharmacological (including motor cortical stimulation) therapies, as well as cell and gene therapies, have been tried, there is no cure. (MD, 2000)

## 1.2 Electroencephalogram (EEG)

An electroencephalogram (EEG) is a type of brain activity recording. Small sensors are placed on the scalp during the test to pick up the electrical signals created as brain cells communicate with one another. A machine records these signals, which are then examined by a clinician to see whether they are uncommon. [2]

An EEG can be used to help diagnose and monitor a variety of brain diseases. It may aid in determining the reason for specific symptoms, such as seizures (fits) or memory issues, or it may provide further information regarding a condition a patient has been diagnosed with.

## 1.3 Data Set

The case study has a total of 18 individuals. They were all evaluated to see if they would develop CNP within 6 months of collecting data. Eight subjects did not acquire CNP within six months of collecting data. They were termed PNP. Within 6 months of data collection, 10 subjects acquired CNP. The ten people were termed as PDP.

Individuals were instructed to relax with eyes closed (EC) and eyes opened (EO). Snippets of data of 5-second duration were captured with 10 repeats per subject. 180 categorized data points were collected in total (18 participants with 10 repetitions each). Every individual had a 48 electrode EEG placed onto their head, which recorded electrical activity in the brain at 250 Hz.

# Chapter 2                      Methodology

## 2.1 Feature Engineering

The process of changing raw data into features that better describe the underlying problem to predictive models, resulting in enhanced model accuracy on unseen data, is known as feature engineering.

Refer Fig 1

## 2.2 Feature Selection

When creating a predictive model, feature selection is the process of minimizing the number of input variables. It is preferable to limit the number of input variables to reduce modelling computational costs and, in certain situations, increase model performance. (Brownlee, 2019)

There are three types of Feature selection: Wrapper, Filter, and Embedded method.

### 2.2.1 Wrapper Method

Wrapper methods attempt to utilize a subset of characteristics to train a model. We determine whether to add or delete features from the subset based on the inferences drawn from the prior model. The challenge has been simplified to a simple search problem. These approaches are often quite computationally costly. [4]

#### 2.2.1.1 Recursive feature elimination

It is a greedy optimization method that seeks the highest-performing feature subset. It constructs models frequently and saves the best or worst performing feature at each iteration. It builds the next model using the left features until all the features are used up. It then rates the traits in the order in which they were eliminated.

### 2.2.2 Embedded Method

Embedded techniques combine the benefits of filtering and wrapping. Algorithms with their built-in feature selection techniques implement it.

#### 2.2.2.1 L1 Regularization using Lasso CV

The Least Absolute Shrinkage and Selection Operator (Lasso) is an effective approach for performing regularization and feature selection on provided data. (Singh, 2020)

L1 regularization is used in Lasso regression, which imposes a penalty equal to the absolute value of the magnitude of the coefficients. (Kaushik, 2016)

## 2.3 Classification

The technique of guessing the class of given data points is known as classification. Classes are sometimes known as goals, labels, or categories. The job of estimating a mapping function from discrete input variables to discrete output variables is known as classification predictive modeling. (Asiri, 2018)

Classification is a type of supervised learning in which the input data is also given to the objectives. Classification has several uses in a variety of fields, including credit approval, medical diagnosis, and target marketing.

### 2.3.1 Support Vector Machine (SVM)

The support vector machine is a model that may be used for both classification and regression tasks. The algorithm generates a hyperplane or line (decision boundary) that divides data into classes. It implements the kernel technique to choose the optimum line separator (decision boundary that has the same distance from the boundary point of both classes). It is a more direct and powerful method of learning complicated nonlinear functions.

In the Case study, we are using “linear” value for the parameter of the kernel.

### 2.3.2 Logistic Regression

Logistic regression is a classification algorithm that is used to solve classification issues. It is a method of predictive analysis in which data is described and the link between variables is explained. (Singh, 2020)

$$0 \leq h_{\theta}(x) \leq 1$$

The output variable (y) is a discrete number that ranges from 1 (yes) to 0 (no), whereas the input variable (X) is continuous (no). It finds the association between variables using the logistic (sigmoid) function. Any real-valued integer may be mapped to a value between 0 and 1, but never exactly within those boundaries, using the sigmoid function.

## 2.4 Cross-Validation

Cross-validation is a statistical technique for estimating the ability of machine learning models. It is often used in applied machine learning to compare and select a model for a specific predictive modeling issue since it is simple to grasp, simple to implement, and produces skill estimates with lower bias than other approaches. The k-fold cross-validation process is used to estimate the model's skill on new data. (Brownlee, 2018)

### 2.4.1 Leave-One-Subject Cross-Validation

We use KFold to divide the data into 18 groups consecutively for each classification model run - with or without Feature selection. The data is preserved in a sequential order to guarantee that each group contains data pertaining to a single subject/patient, as needed by our experiment.

The Leave-One-Subject cross approval ensures that the element determination is applied exclusively to the train information and that the test information stays obscure and doesn't add to the choice of the best elements. this technique additionally guarantees that the classifier predicts "inconspicuous" information, which assists us with testing and checking the exhibition of the prepared model.

### 2.4.2 K-Fold Cross-Validation

K-fold cross-validation is a hyperparameter tuning process that allows you to prepare the model with the optimal hyperparameter values. It is a non-substitution resampling method. The advantage of this method is that each example is only used once for preparation and approval. (Kumar, 2020)

## 2.5 Steps for Implementation

1. Divide data into 18 equal sets using KFold.
2. Iterate over each combination of 17 vs 1 persons' data using a for loop, where the data points of 17 subjects are regarded for training and the data points of 1 subject is the unseen data to be tested to gauge performance.
3. We now input the training data to the feature selection model in each iteration (fold) to obtain the best-picked features. In this phase, we use cross-validation once more by dividing the training data into 16 vs 1 participants' data and passing this KFold object to the feature selection model for results.
4. Based on the outcomes of the feature selection, we obtain the reduced data and use it to train the Classifiers (described above) and forecast the values for the test data.
5. We compute the fold's confusion matrix and store it after comparing the anticipated results to the true values (to be used later to compute mean performance scores).

After all folds have been conducted, we total the confusion matrices and calculate the mean scores for assessment and comparison.

## 2.6 Performance Metrics

Evaluating our machine learning algorithm is a critical component. We are going to measure the performance of the feature engineering with the respective 2 classifications. The performance metrics are used Accuracy, Sensitivity, Recall, and Precision.

A Confusion matrix is a  $N \times N$  matrix that is used to assess the effectiveness of a classification model. The  $N$  refers to target classes. The matrix compares the actual target values to the machine learning model's anticipated values. (Bhandari, 2020)

There are 2 predicted classifiers, Positive(P) and Negative(N). There are also 4 outcomes that are possible for classification. The four outcomes are True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN).

Performance Metrics:

1. Accuracy: It is computed by dividing the total number of correct predictions by the total number of observations in the dataset. The highest accuracy is 1.0, while the lowest is 0.0. (Saito, 2015)

$$Accuracy = (TP+TN)/(P+N) \quad \text{where, } P = TP + FN, \text{ and } N = FP + TN$$

We use two accuracy modes, the first one is mean accuracy where we calculate the accuracy of each iteration. Secondly, the standard deviation is calculated for all accuracy denotes the amount of variation in the accuracy of each iteration.

2. Precision: The number of correct positive predictions divided by the total number of positive predictions yields precision. Positive predictive value is another name for it. (Saito, 2015)

$$Precision = TP/(TP+FP)$$

3. Sensitivity: The number of correct positive predictions divided by the total number of positives yields Sensitivity. (Saito, 2015)

$$Sensitivity = TP/P \quad \text{where, } P = TP + FN$$

4. Specificity: The number of valid negative predictions divided by the total number of negatives yields Specificity. (Saito, 2015)

$$Specificity = TN/N \quad \text{where, } N = FP + TN$$

5. Recall: Recall is a statistic that measures how many correct positive predictions were produced out of all possible positive predictions. (Saito, 2015)

$$Recall = TP/(TP+FN)$$

## Chapter 3 Result

### 3.1 No Feature Engineering

Refer Fig 6, 7, 8 and 9.

The Fig 9. shows the five mean performance scores - accuracy, precision, recall, specificity, and sensitivity (which is the same as recall) - for each fold run between the two classifiers. It's easy to conclude that both classifiers performed equally based on the findings (in terms of accuracy). When using all 432 features over 18 folds to train the models, the 'Linear SVM' Classification model seems to have somewhat better accuracy and precision.

Performance measures when no feature selection method is applied

```
In [35]: no_feature_selection_scores_df
```

Out[35]:

	Accuracy (Mean)	Accuracy (Std)	Precision	Recall	Specificity	Sensitivity
Classifier						
Logistic Regression	0.855556	0.116625	0.875	0.890909	0.8	0.890909
Linear SVM	0.872222	0.0762574	0.894309	0.916667	0.783333	0.916667

Fig 6. Performance measures when no feature selection is applied

### 3.2 Embedded Method

Refer Fig 11. The accuracy of each fold run, as well as the number of best features obtained, are shown in the graph. The figure reveals that folds with range of 105-110 best features picked have accuracy closer to 1, marking this as the optimal range to use for a Logistic Regression classifier.

Refer Fig 12. While the plot has a random character that does not allow us to determine a range of an optimal number of features, most folds with number of best features between 90 and 110 had accuracy closer to 1. Thus, for Linear SVM Classifier, this range can be regarded ideal.

Refer Fig 13. It's clear that both classifiers performed nearly equally (in terms of accuracy), but with lower accuracy than when the classifiers were run across folds without any feature selection.

Performance measures when Embedded (L1 Regularisation) method is applied

```
In [36]: embed_scores_df
```

Out[36]:

	Accuracy (Mean)	Accuracy (Std)	Precision	Recall	Specificity	Sensitivity
Classifier						
Logistic Regression	0.811111	0.107838	0.858491	0.827273	0.785714	0.827273
Linear SVM	0.805556	0.0970236	0.815534	0.84	0.7625	0.84

Fig 10. Performance measures when Embedded (L1 Regularization) method is applied

### 3.3 Wrapper Method

Refer Fig 15. Most folds with number of best features between 150 and 310 had accuracy near to 1. As a result, using the Logistic Regression Classifier employing RFE Feature Selection, this greater range might be regarded as ideal.

Refer Fig 16. Most folds with number of best features between 110 and 210 and 290 and 420 demonstrated accuracy near to 1. As a result, with the Linear SVM Classifier with RFE Feature Selection, this greater range can be regarded ideal.

Refer Fig 17. It's clear to deduce from the findings that both classifiers performed equally (in terms of accuracy), but with lower accuracy than when the classifiers were run across folds without any feature selection.

Performance measures when Wrapper (RFE) method is applied

```
In [37]: wrapper_scores_df
```

Out[37]:

	Accuracy (Mean)	Accuracy (Std)	Precision	Recall	Specificity	Sensitivity
Classifier						
Logistic Regression	0.822222	0.13317	0.846939	0.83	0.8125	0.83
Linear SVM	0.844444	0.107509	0.870968	0.9	0.733333	0.9

Fig 14. Performance measures when Wrapper (RFE) method is applied

## Chapter 4 Discussion

Refer Fig 9, 13 and 17.

While the classifiers trained using feature selection models have lower accuracy across folds than the classifiers trained with all 432 features, we can see that these values of accuracy and precision indicate that the models have been well trained and that when used to test additional unseen data, they will produce similar results as the default models (trained with all features).

All the measured metrics tend to be closer to 1 and in the neighborhood of the default model's performance, indicating the models' efficiency and improved performance.

While RFE feature selection (wrapper technique) takes longer, in general, to obtain the optimal features to utilize (due to individual feature removal procedure), it performed better than lasso regularization (embedded method) for the Linear SVM classifier.

We also notice that, regardless of feature selection, the Linear SVM classifier outperforms the Logistic Regression classifier.

# References

1. Asiri, S., 2018. *Machine Learning Classifiers*. [Online]  
Available at: <https://towardsdatascience.com/machine-learning-classifiers-a5cc4e1b0623>
2. Bhandari, A., 2020. *Confusion Matrix – Not So Confusing!*. [Online]  
Available at: <https://www.analyticsvidhya.com/blog/2020/04/confusion-matrix-machine-learning/>
3. Brownlee, J., 2018. *A Gentle Introduction to k-fold Cross-Validation*. [Online]  
Available at: <https://machinelearningmastery.com/k-fold-cross-validation/>
4. Brownlee, J., 2019. *How to Choose a Feature Selection Method For Machine Learning*. [Online]  
Available at: <https://machinelearningmastery.com/feature-selection-with-real-and-categorical-data/>
5. Kaushik, S., 2016. *Introduction to Feature Selection methods with an example (or how to select the right variables?)*. [Online]  
Available at: <https://www.analyticsvidhya.com/blog/2016/12/introduction-to-feature-selection-methods-with-an-example-or-how-to-select-the-right-variables/>
6. Kumar, A., 2020. *K-Fold Cross Validation – Python Example*. [Online]  
Available at: <https://vitalflux.com/k-fold-cross-validation-python-example/>
7. MD, K. K. J., 2000. *Central neuropathic pain*. [Online]  
Available at: <https://www.medlink.com/articles/central-neuropathic-pain>
8. Singh, A., 2020. *Feature Selection For Dimensionality Reduction(Embedded Method)*. [Online]  
Available at: <https://medium.com/analytics-vidhya/feature-selection-for-dimensionality-reduction-embedded-method-e05c74014aa>
9. Takaya Saito and Marc Rehmsmeier (2015) The precision-recall plot is more informative than the ROC plot when evaluating binary classifiers on imbalanced datasets. *PLoS One*. 10(3):e011843

## Appendix

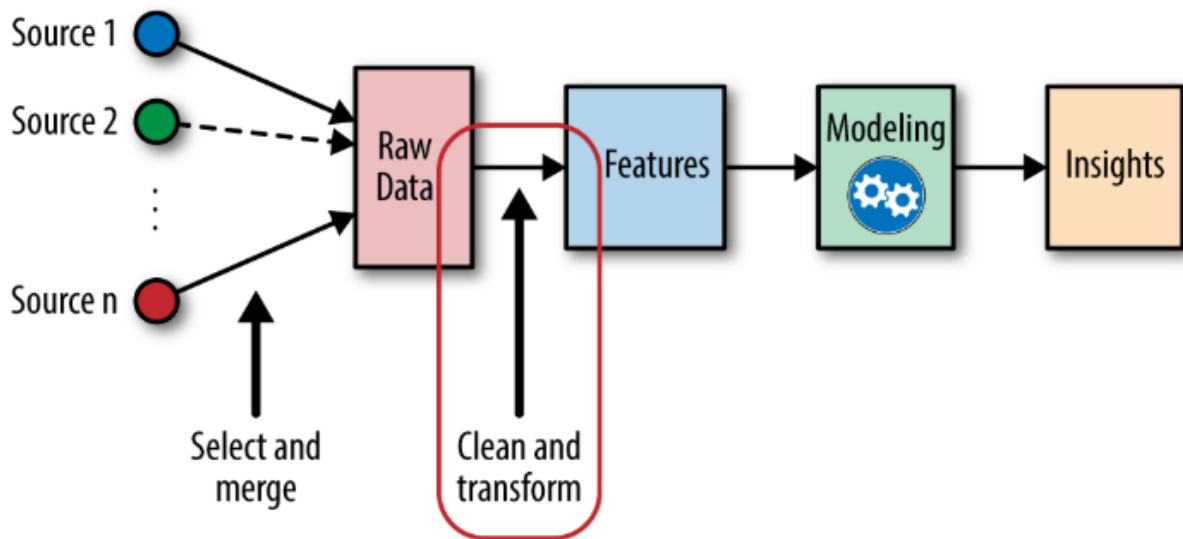


Fig 1. Feature Engineering

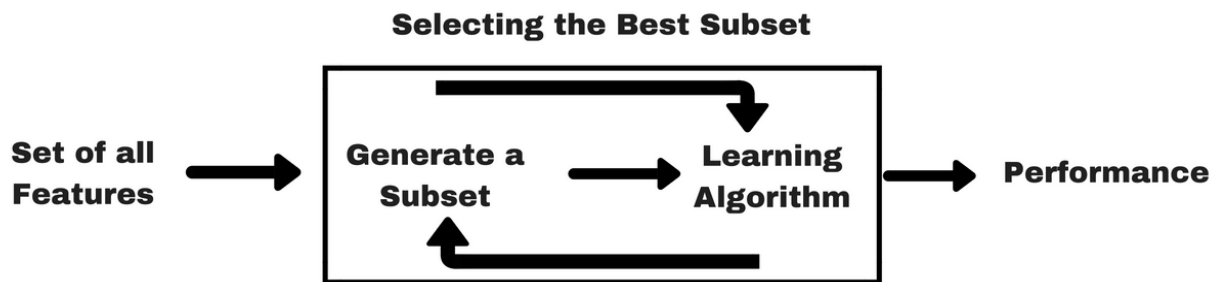


Fig 2. Wrapper Method

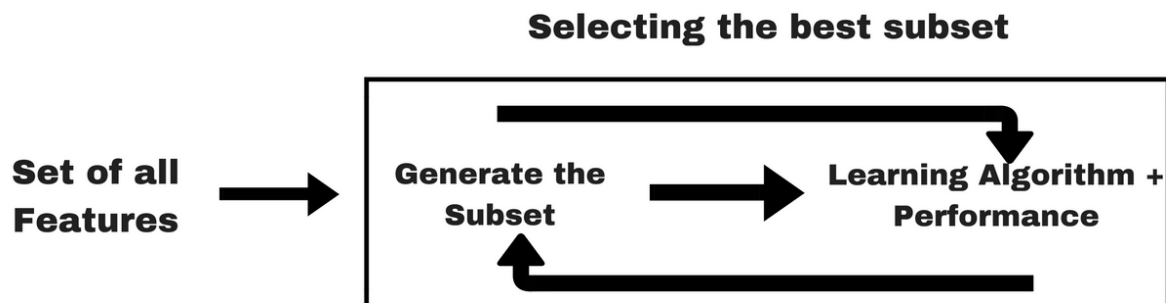


Fig 3. Embedded Method

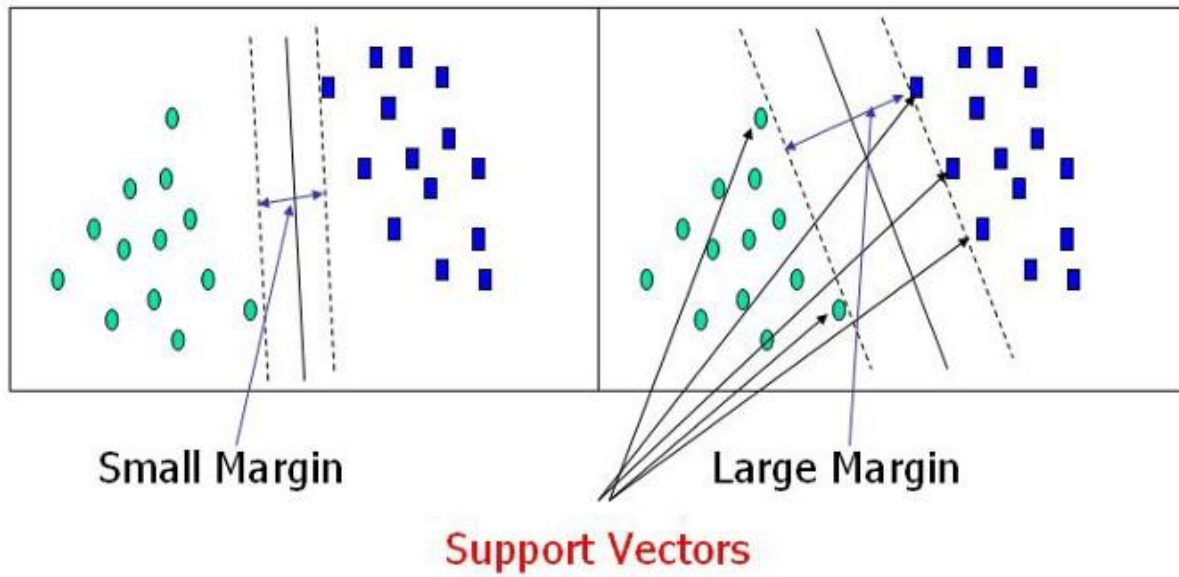


Fig 4. SVM

		ACTUAL VALUES	
		POSITIVE	NEGATIVE
PREDICTED VALUES	POSITIVE	TP	FP
	NEGATIVE	FN	TN

Fig 5. Confusion Matrix

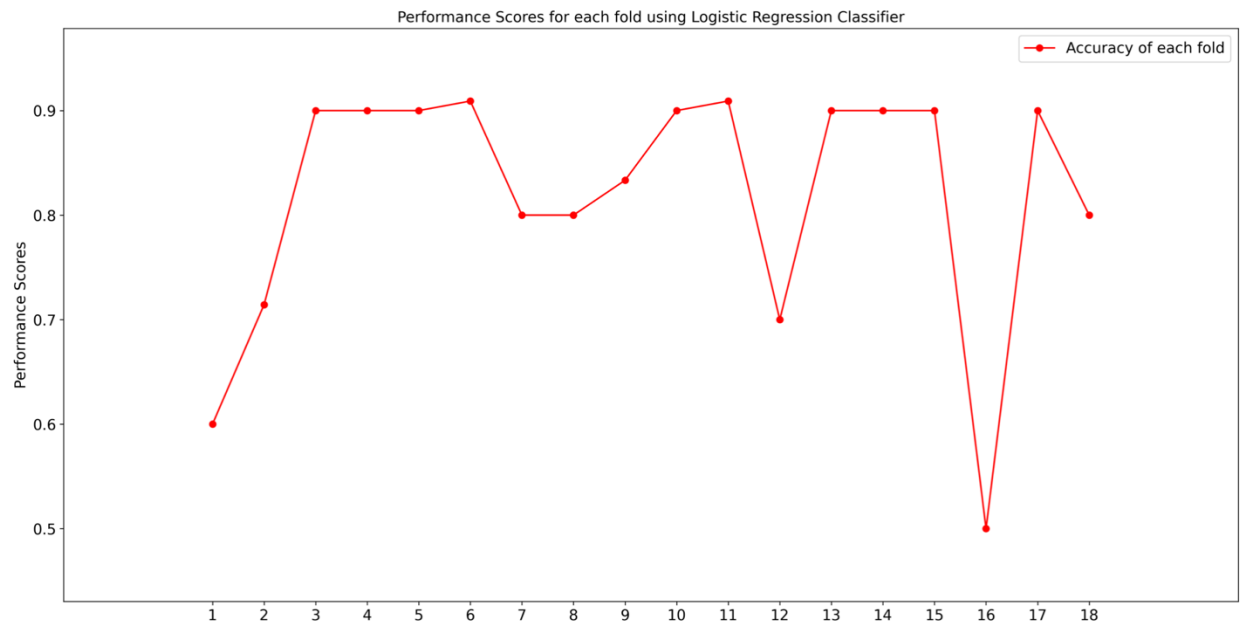


Fig 7. Performance Score for each fold using Logistic Regression Classifier



Fig 8. Performance Score for each fold using Linear SVM Classifier

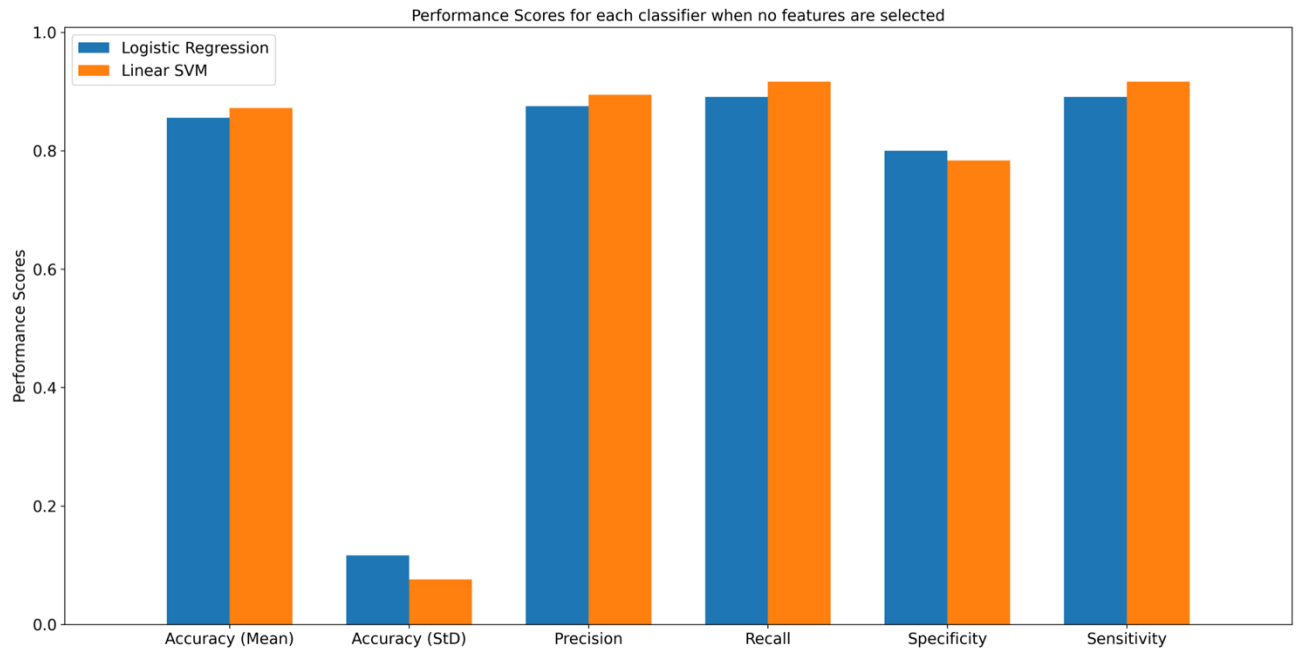


Fig 9. Performance Score for each classifier when no features are selected

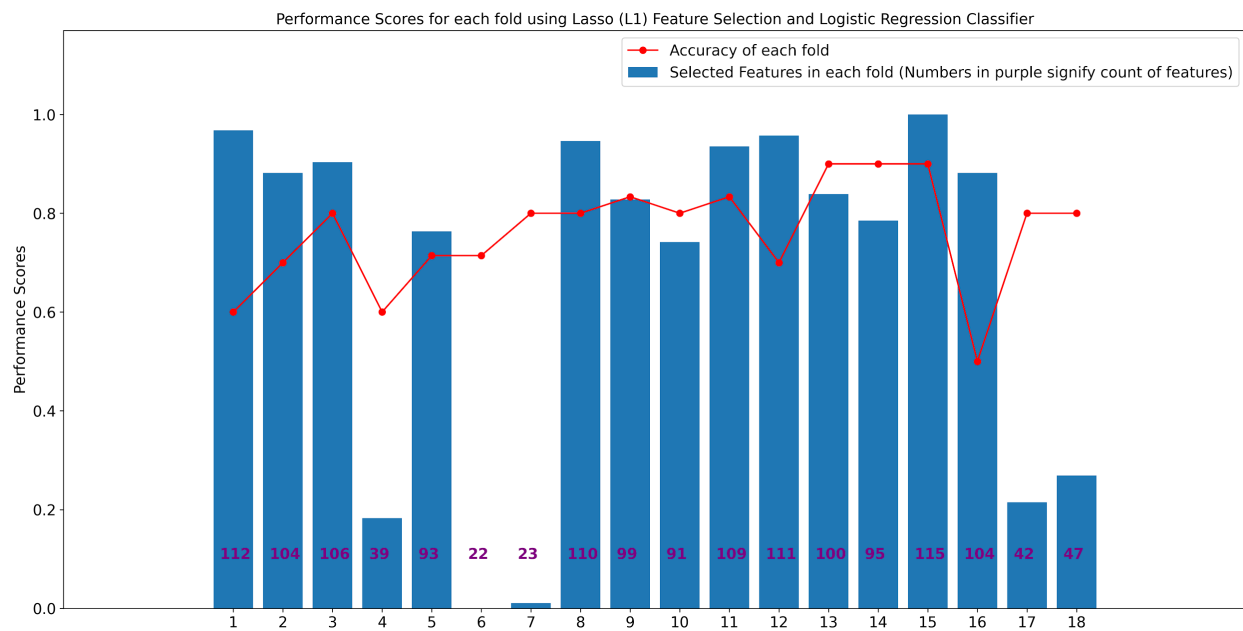


Fig 11. Performance Score for each fold using Lasso (L1) Feature Selection and Logistic Regression Classifier

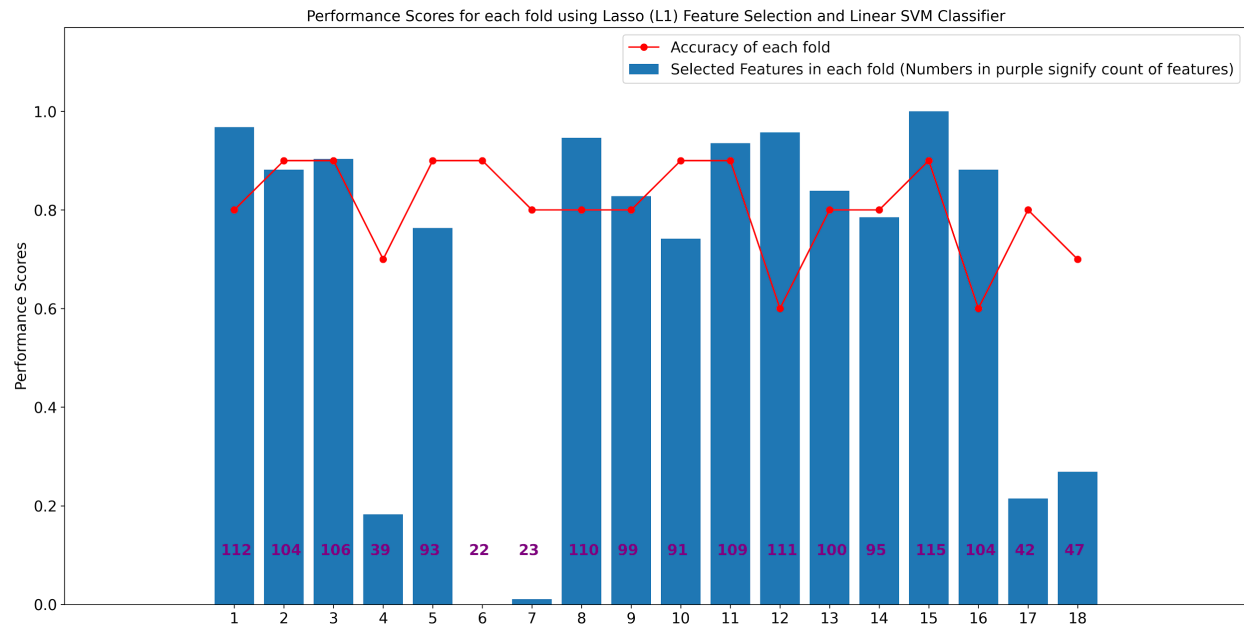


Fig 12. Performance Score for each fold using Lasso (L1) Feature Selection and Linear SVM Classifier

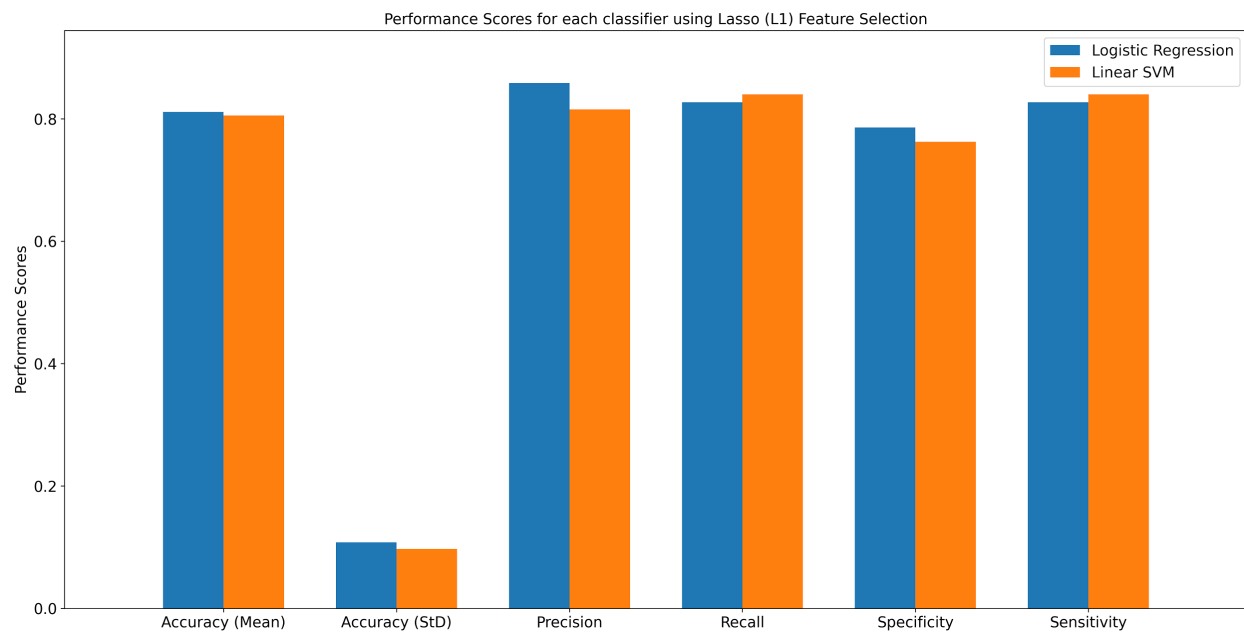


Fig 13. Performance Score for each classifier using Lasso (L1) Feature Selection

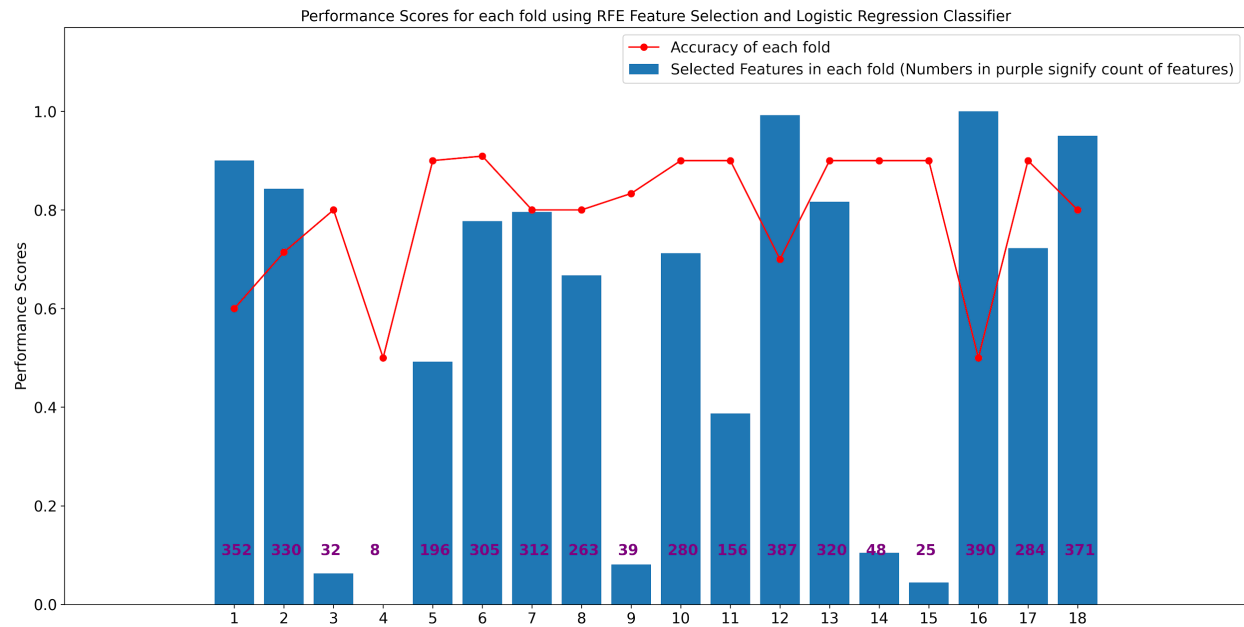


Fig 15. Performance Scores for each fold using RFE Feature Selection and Logistic Regression Classifier

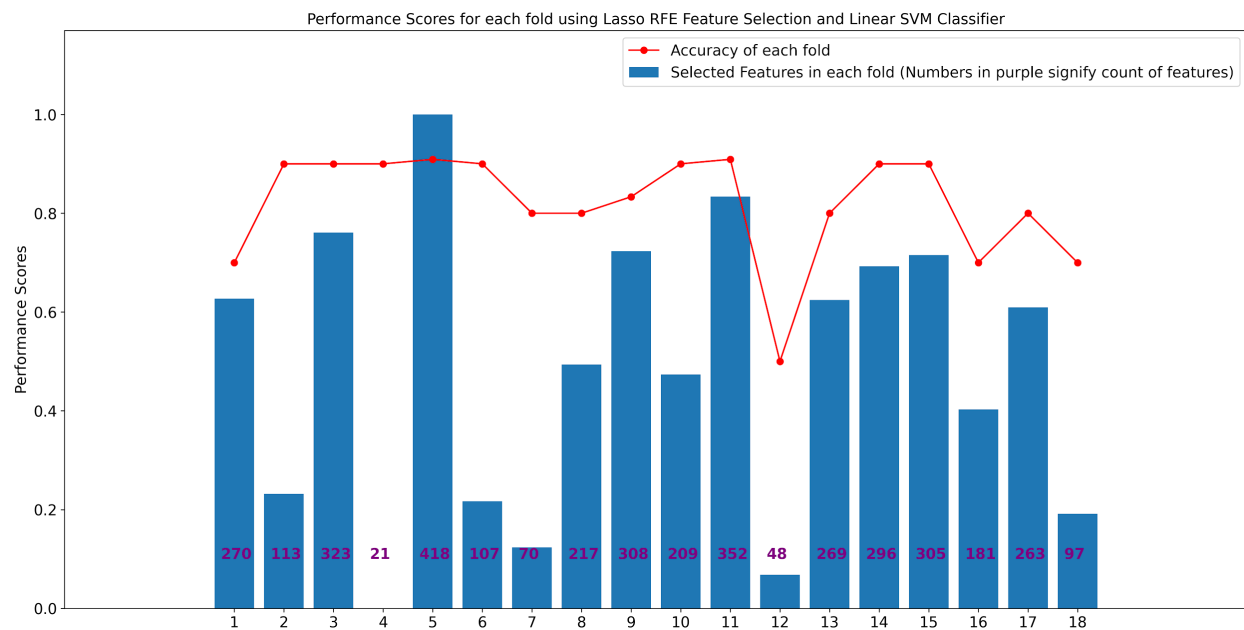


Fig 16. Performance Score for each fold using Lasso RFE Feature Selection and Linear SVM Classifier.

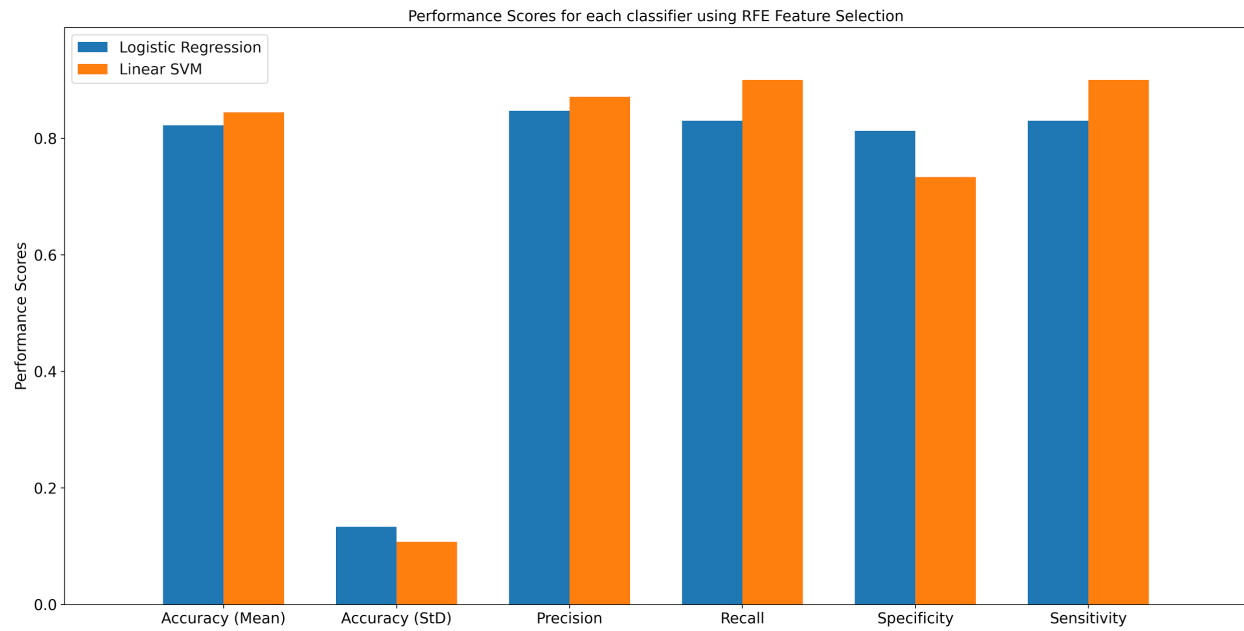


Fig 17. Performance Scores for each Classifier using RFE Feature Selection