COMPSCI5100 Machine Learning & Artificial Intelligence for Data Scientists(M) :

Case Study 1 - Feature Engineering

Jiahuan Mai - 2549379M

Vaishnavi Balaji - 2549408B

Xinli Wang - 2519768W

School of Computing Science

Sir Alwyn Williams Building

University of Glasgow

G12 8RZ

Friday 2nd April 2021

Table of Contents

Introduction:

Approximately 50% of people with Spinal Cord Injury (SCI) have Central Neuropathic Pain (CNP). Pain in response to non-painful stimuli, episodic (electric shock), “pins and needles”, numbness. There is currently no treatment, only prevention Preventative medications have strong side-effects Predicting whether a patient is likely to develop pain is useful for selective treatment Manual assessment is time-consuming, error-prone, and somewhat subjective There is some evidence that brain Electroencephalogram (EEG) data has characteristic markers. We have a (small) dataset with EEG from SCI patients, of which some later developed CNP. The data is extremely high-dimensional, so it is very hard for a classifier to tell them apart. To overcome this curse of dimensionality problem we are doing some feature engineering and feature selection techniques and extracting a small subset of useful features that will result in better performance of our machine learning model.

Methods:

High Dimensional datasets are very challenging to work with. As the number of features increases, the error rate also increases while the performance decreases. It refers that algorithms are harder to design in the high dimensional datasets and often have a running time exponential in the dimensions. A fixed number of data points occupy a fraction of the space that decreases exponentially with the number of dimensions. The portion of a volume near the surface increases in the number of dimensions. Distances between points become increasingly indistinguishable. In our dataset, we have a lot of features and if fed to the model usually it gets confusing because it is learning too much of the data. In order to resolve that situation, we apply various feature selection techniques and select those features to the model, increasing the accuracy and time constraint.

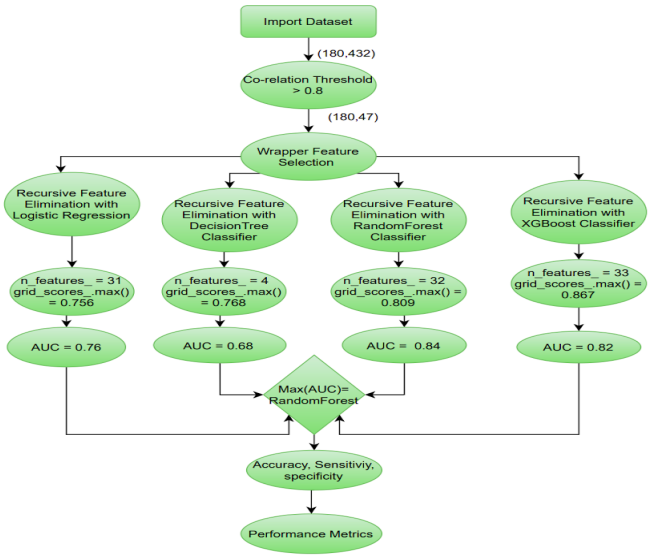

Fig 2.1 Flowchart

Dataset Collection and Preparation:

A total of 18 participants and their brain signals under Electroencephalogram (EEG) is observed. Out of these 18 participants, 8 participants did not develop Central Neuropathic Pain (CNP) within 6 months after the data collection (PNP). The other 10 participants developed Central Neuropathic Pain (CNP) within 6 months after the data collection. The dataset consists of the raw data of all the single electrodes with 9 features each, a total of 43 electrodes with 9 features. The Frequency Band power is calculated for PNP eyes closed and PNP eyes opened and observations are made. The Theta, Alpha, and Beta Bandpower for PNP and PDP are also observed.

For simplicity's purpose, the given three datasets for this case study namely data.csv, features.csv, and labels.csv are combined into a single dataset as **spinal.csv.** All the methods needed for this process are imported and the dataset is loaded. Here, the dataset has 432 independent features which are a preprocessed data of signal denoising, normalization, temporal segmentation, frequency band power estimation. Overall, it comprises 180 rows (18 subjects x 10 repetitions) x 432 columns (9 features x 48 electrodes). And the dependent variable tells us whether the patient with the Spinal Cord Injury (SCI) developed Central Neuropathic Pain (CNP) or not.

Correlation Matrix:

The major issue of the Recursive Feature Elimination method is that it is very expensive to run. So before feeding all of our features to the model, the correlation between the variables is checked. Highly correlated features in our dataset provide similar information, so removing one of those features will increase our efficiency of the model. The correlation coefficient threshold is set to 0.8 and we can eliminate 385 features. Basically, they convey the same information the other 47 features convey. These features are dropped from the dataset. Finally, the correlation matrix is constructed with the 47 features as follows in Fig. 2.2.

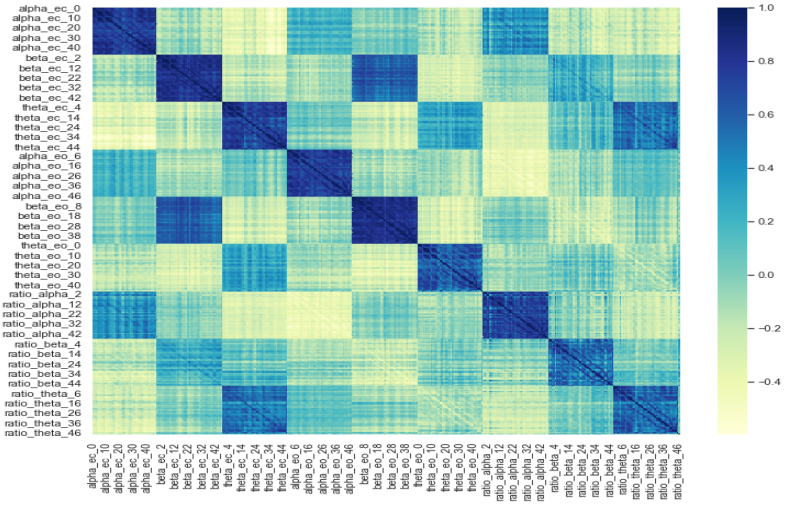

Fig 2.2 Correlation Matrix

Recursive Feature Elimination:

Recursive feature elimination is a wrapper-type feature selection algorithm, that if a machine learning algorithm is given and used in the core, is wrapped by RFE to select the useful features. RFE performs filter-based feature selection within wrapper style algorithms. RFE selects the subset of features from the given dataset according to their importance and removes the less important features. This is accomplished by fitting our machine learning classifier in the model, ranking our features by their significance, eliminating the least important features, and re-fitting the model. This process is recursively done until maximum optimization is achieved (Brownlee, 2020).

Arguments of RFECV include estimator - the model instance that is used, step - the number of features to be eliminated at each iteration, cv - the cross-validation score, use StratifiedKFold and the number of folds is selected, scoring - performance metric to optimize. In this case study, four RFECV is executed with Logistic Regression, Decision Tree classifier, Random Forest Classifier, and XGBoost classifier. Each classifier when wrapped with RFECV gives a different number of optimum features and different accuracy. After analyzing all the execution RFECV with Random Forest gives us 32 optimum features to be selected with the maximum grid score of 0.806.

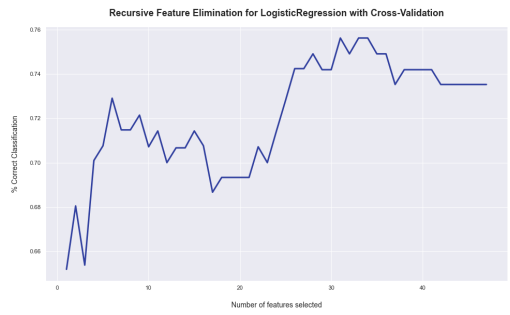

Fig 2.3.1 RFECV with Logistic Regression

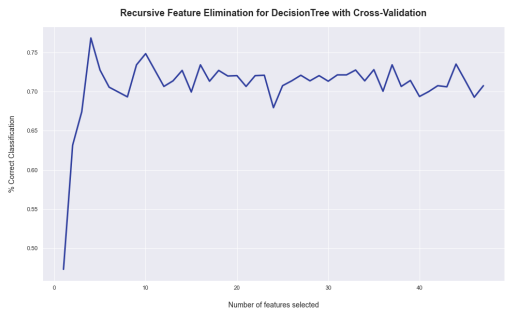

Fig 2.3.2 RFECV with Decision Tree

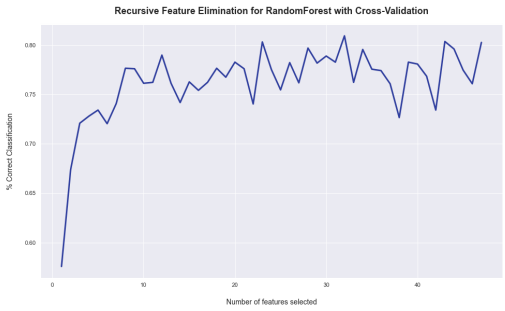

Fig 2.3.3 RFECV with Random Forest

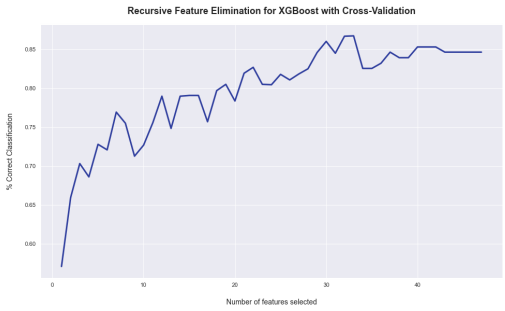

Fig 2.3.4 RFECV with XGBoost

Fig 2.3 RFECV with models

As the random forest classifier when wrapped with RFECV gives us the most optimum 22 features, it can be visualized using a bar chart in regards to their importance. These features are the ones that are the most significant out of all 432 features we had in our original dataset.

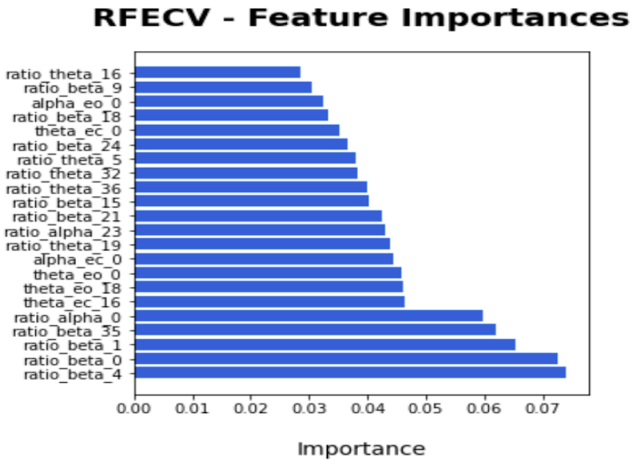

Fig 2.4 Feature Importance

AUC-ROC Curve:

In order to measure the probability of prediction, we use the AUC-ROC curve that is Area Under the curve and Receiver Characteristic Operator. ROC is only used in place of binary classification problems. ROC is the probability curve for True Positive Rate and the False Positive rate basically separates the signal from the noise. The Area Under the Curve (AUC) is basically the summary of the ROC curve which measures the liability of the given classifier model to differentiate between the classes. The higher the AUC value the better our classifier model has performed in distinguishing between the two binary classes (Bhandari, 2020).

In this case study, the AUC-ROC curve is plotted for all four classifier models with RFECV as mentioned above. Out of our observations made, it is clear that RandomForest Classifier with RFECV gives the higher AUC value of around 0.84, which implies this pipeline has a better ability to differentiate between positive and negative classes efficiently while comparing with other classifier models.

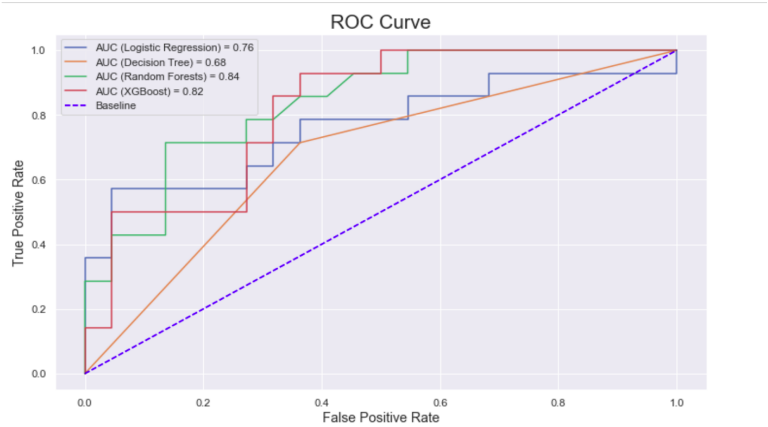

Fig 2.5 ROC_AUC curve

Confusion Matrix:

The confusion matrix is an important metric that will tell us how well our classifier model has performed in real-time. The confusion matrix will tell us about the True Positive Rate, False Positive Rate, True Negative Rate, False Negative Rate. In our binary classifier case study, we may have two different types of errors, that is False Positive rate and False Negative rate that is assigning the data points to different classes. Their confusion matrix is a convenient way to solve and display this information for our machine learning model (Brownlee, 2016).

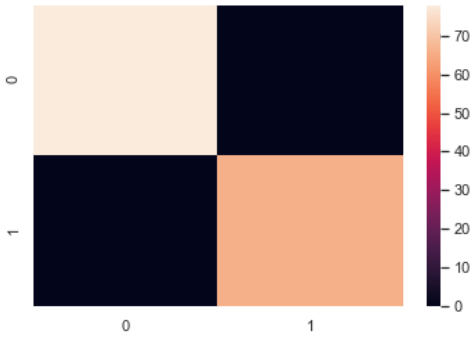

Fig 2.6 Confusion Matrix

Accuracy, Sensitivity, and Specificity:

Accuracy is simply calculated by the ratio of the y_test and the predicted value of the machine learning model. Accuracy is estimated by True positive plus True Negative by True positive to the sum of False Positive False Negative and True Negative. In terms of good evaluation, the AUC curve is more trusted than the accuracy values. Sensitivity is basically the True positive value, it tells what portion of the positive class is correctly classified. Sensitivity is calculated by True positive value divided by True positive and False Negative values. Specificity is the true negative value, revealing what portion of the positive class is incorrectly classified by our model classifier. Specificity is calculated by True Negative values divided by False Positive and True Negative values. Specificity and Sensitivity are the most important performance metrics to be measured and evaluated. The trade-off is often performed to maximize our performance of our model, which is usually visualized by the ROC-AUC curve. Positive Predictive Value gives the right decision that is made by our model, which is usually calculated as True positive divided by True positive and False Positive. Negative Predictive Value gives the unsuccessful classification usually calculated by using True Negative divided by False-negative and True negative (Fricker, n.d.).

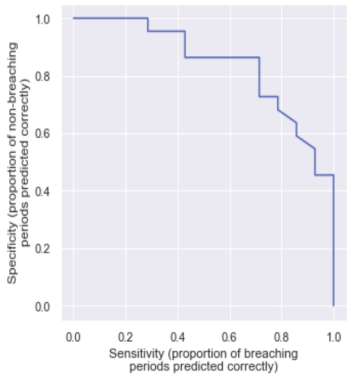

Fig 2.7 Sensitivity vs Specificity

Precision, Recall, and F1 Score:

Precision gives us the proportion of correctly predicted positive observations to the total number of Positive observations predicted. It is calculated by True Positive divided by True Positive and False positive. The recall value is also called Sensitivity. F1 score gives us the weighted average score of the Precision and Recall, thereby taking all the false positive and the false negative value into account. F1 score is more accurate and useful than accuracy calculations. Numerically F1 score is calculated by 2 multiplied by recall and Precision and divided by the sum of Recall and Precision (Joshi, 2016).

Results:

In this Case study, We observed feature selection techniques and machine learning techniques to improve our performance of our model and also to overcome the curse of dimensionality. This machine learning approach is a practical and automatic tool to help diagnose patients who are more likely to develop Central Neuropathic Pain (CNP) after a short period of their Spinal cord Injury (SCI) instead of random guessing. Our feature selection techniques helped us to reduce 432 features to the most important 22 features and Random forest Classifier is used as a machine learning model to predict this automation. Overall our study involves the role of machine learning in medical problems and to improve the classification of the automatic diagnosis on the clinical practice. The performance metrics for our Machine Learning model is observed and shown in Fig 2.10.

```
Machine learning diagnostic performance measures:
-------------------------------------------------
accuracy = 0.750
sensitivity = 0.643
specificity = 0.818
positive_likelihood = 3.536
negative_likelihood = 0.437
false_positive_rate = 0.182
false_negative_rate = 0.357
positive_predictive_value = 0.692
negative_predictive_value = 0.783
precision = 0.692
recall = 0.643
f1 = 0.667
positive_rate = 0.361
```

Discussion:

Our Case Study focuses on the improvement of the Central Neuropathic Pain (CNP) diagnosis from the observation of the Electroencephalogram (EEG) and its characteristic markers applying machine learning techniques. The results obtained from our experiment imply the curse of dimensionality, in this case, can be greatly reduced by removing the highly correlated features and using the Recursive feature elimination wrapper method with the random forest classifier model. We compared the AUC performance for different classifier models and by that conclusion, the random forest gives us optimal performance. After this analysis, we can conclude that feature selection improves the number of instances correctly classified with respect to the improvement in the performance and also reduces time constraints. Although there is no one way to do feature engineering, according to our observation RFECV with Random forest works most efficiently with 22 features and an AUC value of 0.84 which makes the automation of the diagnosis of CNP much easier and efficient.

Bibliography

1. Bhandari, A. (2020, June 16). *AUC-ROC Curve in Machine Learning clearly Explained*. Analytics Vidhya. Retrieved April 2, 2020, from https://www.analyticsvidhya.com/blog/2020/06/auc-roc-curve-machine-learning/
2. Brownlee, J. (2016, November 18). *What is a Confusion Matrix in Machine Learning*? Machine Learning Mastery. Retrieved April 2, 2021, from https://machinelearningmastery.com/confusion-matrix-machine-learning/
3. Brownlee, J. (2020, May 25). *Recursive Feature Elimination (RFE) for Feature Selection in Python*. Machine Learning Mastery. Retrieved April 2, 2021, from https://machinelearningmastery.com/rfe-feature-selection-in-python/#:~:text=Technically%2C%20RFE%20is%20a%20wrapper,until%20the%20desired%20number%20remains.
4. Fricker, T. (n.d.). *Sensitivity, Specificity, and Predictive Values – What is the best way to measure the performance of binary classification models?* Select Statistical services. https://select-statistics.co.uk/blog/sensitivity-specificity-and-predictive-values-what-is-the-best-way-to-measure-the-performance-of-binary-classification-models/
5. Joshi, R. (2016, September 9). *Accuracy, Precision, Recall & F1 Score: Interpretation of Performance Measures*. Exsilio Solutions. Retrieved April 2, 2021, from https://blog.exsilio.com/all/accuracy-precision-recall-f1-score-interpretation-of-performance-measures/