CASE STUDY 1: FEATURE ENGINEERING

Name: Sruthi Merlin Thomas ID: 2496855T

Introduction

- This case study focusses on different feature engineering techniques applied on a given data set. Feature engineering is the process of applying various techniques that improves the accuracy and the predictive power of the classification process. A feature is a measurable property or building blocks of data set used for the purpose of analysis of data. Feature engineering involves cleaning and preprocessing the raw data making it fit for the classification purpose. Different feature engineering techniques are applied to the data to improve the efficiency of the model depending on the data domain, problem to be solved and the predicative model. The feature selection for instance selects the most important features for classification thus reducing the *Curse of Dimensionality* problem. Further it prevents overfitting of the data as less redundant means less noise. Various methods can be defined as a part of Feature selection Strategies. This report specifically discusses the **Recursive Feature Elimination with cross validation technique.** This report also provides an outline on other strategies used in this case study which was eliminated due to reduction in accuracy of the classifier.

- In this case study, we explore various factors associated with electroencephalogram (EEG) in predicting the Central Neuropathic Pain in patients having a spinal cord injury. Central neuropathic pain is defined as 'pain caused by a lesion or disease of the central somatosensory nervous system'. It is most common is patients with Spinal cord injury, brain injury etc. The chances of getting a Central Neuropathic Pain is 50 percent. As a part of this case study, 18 patients with spinal cord injury are identified of which 10 patients develop a Central Neuropathic Pain, while 8 patients do not develop a pain. The main aim of the experiment is to predict if a patient with a spinal cord injury gets a Central Neuropathic pain so that appropriate preventive measures could be provided. The raw data represents voltage measured 250 times per second at each electrode location. It contains information on patients who will develop a neuropathic pain and who will not. This defined by classes 1 and 0 respectively. The data also contains features on various frequency spectrum collected with eyes opened and closed. In total each subject has 10 data samples for 432 features sets. The final output should have a classifier that predicts whether a patient will develop a pain (i.e. 1) or not (i.e. 0). The main aim of the case study is to compare various feature engineering and find out the best one based on the cross-validation score, accuracy score, sensitivity and specificity of the classifier.
-

Methods

1. Basic Task definition:

1) Calculate Cross validation score, Accuracy, Sensitivity and the Specificity on the base classifier before applying the feature engineering techniques.
2) Apply feature engineering techniques on the classifier to reduce the features.

3) Calculate Cross validation score, Accuracy, Sensitivity and the Specificity on the classifier after applying diiferent feature engineering techniques.
4) Compare the results and choose and choose an optimal strategy based on the comparison results.

- **Logistic Regression Classifier:**

  The logistic regression function has categorical dependent variables and independent features. Hence it is used to classify binary classes. In this case study, Logistic Regression classifier predicts whether the patient with spinal cord injury develops a pain or not based on the frequency spectrum. In other words, it predicts either 0 or 1 based on independent input features.

  The Scikit Learn has the following inbuilt functions to perform Logistic regression classification:
  sklearn.linear_model.LogisticRegression():
  This class implements the logistic regression classifier.
  LogisticRegression().fit(self, X, y, groups=None):
  This functionality fits the training data into the model.
  LogisticRegression(). predict(X_test):
  This function the outputs the label value based on the trained data.

- **Group K Fold Cross Validation:**

  The cross validation is used to test the effectiveness of the model or the classifier and identify the possibility of overfitting. It is preferred over the normal train and test data split as it gives an opportunity to train multiple train/test splits. In this case study, since each subject is grouped into 10, GroupKFold cross validator is used. The GroupKFold is variant of KFold cross validation strategy where non- overlapping groups are divided into K The data set is split into train data and test data depending on the value of K. The cross-validation score is used to evaluate the model.
  Scikit learn provides following inbuilt functions to perform cross validation and calculate the score:
  class sklearn.model_selection.GroupKFold(n_splits):
  The n_splits parameter determines how the data should be split.
  cross_val_score(logmodel, X, y, groups=groups, cv=group_kfold, scoring='accuracy')
  This function calculates the cross-validation score which is used as measure to evaluate the model and test the effectiveness of the model.

  Parameters of Cross Validation Score:
  - Estimator: The object used to fit the data.
  - X: The data to fit.
  - Y: The target data to be predicted.
  - groups: Group labels used in samples while splitting the data into train/test.
  - cv: The cross-validator technique used to split the data into train and test data.
  - Scoring: This parameter is a string or a function calling the object of the function score(X,y). In this case the score is assigned with the string ‘accuracy’ since the problem is a classification problem.

Feature Engineering Techniques used in the case study

- **Recursive Feature Elimination with Cross Validation:**

  The Recursive Feature Elimination is one of the feature selection methods in which iteratively removes the weaker features until he optimal set of features is reached. The features are ranked by the model's coef_ of feature_importances_. To find the optimal numbers of features RFE uses cross validation score. This method is known as Recursive Feature Elimination with cross validation (RFECV). This method finds the optimal number of features based on the cross-validation score. The features with high rank are selected.

  Scikit Learn has inbuilt functionality for performing RFECV:

  RFECV (estimator=log_rfecv, step=1, cv=group_kfold, scoring='accuracy')

  Parameters for Recursive Feature Elimination with Cross Validation:
  - Estimator: The object used to fit the data and provides information on the relative importance of the features.
  - Step: this parameter refers to number of features to eliminate at each iteration.
  - Cv: The cross validation splitting strategy used. In this case study, the groupKFold object is passed as the parameter.
  - Scoring: This parameter is a string or a function calling the object of the function score(X,y). In this case the score is assigned with the string 'accuracy' since the problem is a classification problem.

- **Chi Squared method:**

  Chi Squared test is one of the feature filtering techniques in which the best features are selected based on the rank of the features. This method is particularly useful when the features are categorical in nature. The Chi-squared test is given by the mathematical formula:

$$\approx^2 = \frac{(\boldsymbol{Observed\ Frequency - Expected\ Frequncy})^2}{\boldsymbol{Expected\ Frequency}}$$

  where:

  Observed Frequency is the Number of observations of the class
  Expected Frequency is Number of Expected observations of the class.

  Scikit learn uses an inbuilt functionality to select the best K features. This function returns an array with chi2 statistics of each feature and the p values.

  sklearn. feature_selection.chi2(X, y)

  Parameters for selecting the best features using Chi Squared test:
  - X: Sample Input Vectors
  - Y: Target vector or the class labels.

- **Mutual Information method.**
  Mutual Information between two random variables X and Y is a non-negative value that measures the dependency between the two variables. Higher the values hogher the dependency. In mathematical notation Mutual Information is described as follows:

$$I(X;Y) = H(X) - H(X|Y)$$

  where:
  I(X;Y) is the mutual information for X and Y.
  H(X) is the entropy for X.
  H(X|Y) is the conditional entropy for X given Y.

  Feature Selection in a NP- Complete problem, we use a greedy approach to select the features in an incremental manner. Optimization of the results involve trading the relevance of the feature with respect to the target label against the redundancy of that information compared to the information stored in that variable.

  Scikit learn uses the following function to for Mutual Information feature selection:

  sklearn.feature_selection.mutual_info_classif(X, y, discrete_features='auto', n_neighbors=3, copy=True, random_state=None)

  The function relies on nonparametric methods based on entropy estimation from k-nearest neighbors' distances.

  Parameters for selecting the best features using Mutual Information:
  - X: Sample Input Vectors
  - Y: Target vector or the class labels.
  - n_neighbors: Number of neighbors for Mutual Information Estimation.

- **ANOVA F Test:**

  Analysis of Variance is a statistical approach in defining best features to be selected by considering the significant difference between two or more groups. ANOVA uses F-Test to determine if there is any significant difference between the groups. If there is no difference the result is close to 1. This means that the feature has no impact on the target output and hence can be eliminated.

  F-Value:
  The F- value is calculated by comparing the variance between the groups and within the groups. In statistical notation, it can be described as follows:

$$\boldsymbol{F - Value = \frac{SBB/dfb}{SSW/dfw}}$$

  where:

SSB is the total sum of squares between the groups. It is defined as squared sum of all the distance between each group average from the mean value of all the data points or the grand mean.

SSW is the total sum of squares within the group.It is defined as the sum of squares of distance from each observed value with the group to the group mean.

dfb is the degree of freedom between the groups.

dfw is the degree of freedom within the groups.

Scikit learn uses an inbuilt functionality to calculate the F-value:

sklearn.feature_selection.f_classif(*X*, *y*)

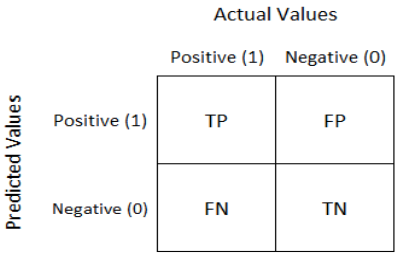
Parameters for calculating the F value:
- X: Sample Input Vectors
- Y: Target vector or the class labels.

- **Projection through Principal Component Analysis:**

  Principal component analysis (PCA) is the process of transforming high dimensional data to lower dimensions while retaining the necessary information and patterns. The main goal of this process is to find the minimum number of principal components to fit the best summary of the data. The principal components are chosen in such a way that it lies in the direction of the maximum variation in the data. To perform projection, we first compute the covariance matrix that shows the correlation between the features in for of a matrix. It defines both variance (spread) and the covariance(orientation) of the data The diagonal of the covariance matrix gives the variance of the features. On this matrix, we perform the eigen decomposition to get the eigen vectors and eigen values. The eigen vectors represent the direction of the principal components and the eigen values represent the magnitude.

- **Metrics to evaluate the effectiveness of the model:**

  - **Confusion Matrix:**
    The confusion matrix is one of the most popular metrics from which most of the performance metrics are based to evaluate the accuracy and the correctness of the model. It is used in most of the classification problems where there are two or more classes as output. The confusion matrix is a tabular structure with 2 dimensions i.e. Actual and predicted with the output classes along each dimension. The basic structure of the confusion matrix is as follows:



Matrix Terminologies:

1) **True Positive (TP):** When positive value is predicted, and it is true.
2) **True Negative (TN):** When negative value is predicted, and it is true.
3) **False Positive (FP):** When positive value is predicted but the actual value is **negative.**
4) **False Negative (FN):** When negative value is predicted but the actual value is positive.

- **Accuracy Score:** The Accuracy Score of a classifier is the ratio of the correct predictions to the total number of the input samples.

$$\boldsymbol{Accuracy} = \frac{\boldsymbol{TP + TN}}{\boldsymbol{TP + TN + FP + FN}}$$

- **Sensitivity:** The sensitivity also known as the true positive rate is the ratio of the positive values correctly identified to the total number of positive values.

$$\boldsymbol{Sensitivity} = \frac{\boldsymbol{TP}}{\boldsymbol{TP + FN}}$$

- **Specificity:** The sensitivity also known as the true negative rate is the ratio of the negative values correctly identified to the total number of negative values.

$$\boldsymbol{Specificity} = \frac{\boldsymbol{TN}}{\boldsymbol{TN + FP}}$$

- **Cross Validation Score:**
  Although the accuracy, sensitivity and specificity are used for measuring the correctness of the model, there are chances of overfitting or underfitting especially when the data is distributed in an uneven manner. To prevent this problem, cross validation can be used as an evaluation measure by splitting and shuffling the data, thus distributing the data evenly.

Results

Input Data

In this case study, we use preprocessed data that contains information on 18 patients. The experiment is repeated 10 times for each patient i.e. 10 rows for each patient. The data in total has 18 *10= 180 rows. The rows are arranged in a subject major order. There are 9 features which include normalized alpha, beta, theta band power while eyes closed, eyes open and their ratio for each of the 48 electrodes which. Hence there are 9 * 48=432 columns. The dataset can be represented as an array of shape 180 x 432. The labels are represented in 2 classes, patient will develop neuropathic pain, or the patient will not develop neuropathic pain i.e. 0 or 1. Out of 18 patients, 10 patients develop pain while 8 patients do not develop pain. The feature identifiers for all the columns are stored in feature_names.csv. The labels.csv contains the label class for each row of data.

GroupKFold Cross Validation:

Since each patient has 10 records, the input data is split into test data and train data based on groups. The groups value is defined for each patient. The groupKFold cross validator splits the data into 3 folds based on the parameter n_splits=3. The groups value is passed as parameter to the groupKFold function. After cross validation, the data is split into X_train, X_test, y_train and y_test.

Logistic Regression:

Logistic regression is a form of binary classifier that estimates either 0 or 1. The classifier is trained on the X_train and y_train data and tested on X_test data using the predict function. The results are used to build the classification report and confusion matrix to calculate the accuracy score, sensitivity and specificity of the classifier.

Classification report of the base classifier:

The classification report is an inbuilt function in scikit report that provides the summarized results of accuracy, sensitivity (recall), f1 score and precision. Below figure represents the classification of the classifier.

	precision	recall	f1-score	support
0.0	0.94	0.82	0.88	40
1.0	0.72	0.90	0.80	20
accuracy			0.85	60
macro avg	0.83	0.86	0.84	60
weighted avg	0.87	0.85	0.85	60

Figure 1

Confusion Matrix:

The confusion matrix also known as the error matrix can be used as an alternative to calculate the accuracy, sensitivity and the specificity of the classifier. Below figure shows the confusion matrix of the base classifier:

```
[[33  7]
 [ 2 18]]
```

TRUE POSITVE (TP)	**33**
TRUE NEGATIVE (TN)	**18**
FALSE POSITIVE (FP)	**7**
FALSE NEGATIVE (FN)	**2**

Accuracy Score:

There are 2 ways to calculate the accuracy score.

1) **From the confusion matrix:**

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} = 0.85$$

2) **From Scikit Learn:**

accuracy_score(y_test,predictions)=0.85

Sensitivity:

$$Sensitivity = \frac{TP}{TP+FN} = 0.94$$

Specificity:

$$Specificity = \frac{TN}{TN+FP} = 0.72$$

Cross Validation Score:

Although, Accuracy sensitivity and specificity gives a measure on the correctness of the classifier, there are chances of overfitting especially when the data is not distributed. To prevent overfitting, we use the concept of cross validation score to measure the effectiveness of the model. The cross-validation score can be calculated by using the function cross_val_score. The mean of the cross-validation score to taken to compare the results.

Feature Engineering Techniques:

Various feature engineering methods were applied, and an optimal strategy was chosen based on the comparison of the evaluation scores.

1) **Chi Squared Method**

On applying Chi Squared method, the accuracy and cross validation scores decreased. This is because the chi squared method is mainly applicable for features with categorical data. Therefore, this strategy was eliminated as it reduced the accuracy scores.

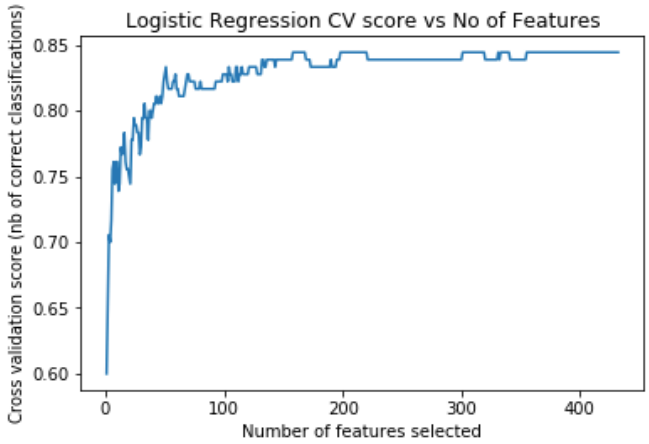
2) **Mutual Information Feature selection**

Similarly, the application of Mutual Information has also decreased the accuracy which as a result was eliminated from the feature engineering strategies in this case study.

3) **ANOVA F- Test**
ANOVA F test calculates the significant difference between the groups and within the groups.

4) **Recursive Feature Elimination**
The optimal features were selected recursively based on the cross-validation score. Based on the below plot, optimal number of features is 158 for which the cross-validation score is highest. The top 158 relevant features are chosen.

```
Optimal number of features : 158
```



5) **Principal Component Analysis**
After applying Recursive Feature Elimination with Cross validation, the top 158 features are projected onto first 2 principal components. This further improved the accuracy of the classifier. The n_components parameter was tuned with a variance value of 0.95 for which the accuracy of the classifier was the highest.

Results Comparison:

Below table compares the different scores of the base classifier and after applying the feature engineering techniques:

Measures	Base Classifier	Chi Squared Method	Mutual Information	ANOVA-F test	RFECV	RFECV with PCA
Accuracy	0.85	0.73	0.73	0.73	0.85	0.9
Sensitivity	0.94	0.75	0.75	0.75	0.82	0.85
Specificity	0.72	0.7	0.7	0.7	0.9	1.0
Cross Validation Score	0.84	0.73	0.73	0.73	0.86	0.86

Discussion

- Based on the comparison of results of Accuracy Score, Sensitivity , Specificity and Cross Validation score, we observe that RFECV with PCA has the highest combination of scores.. Further, it is observed that the 3 Feature Filtering methods i.e Chi-Squared, Mutual Information and ANOVA F-test, not only result in lower scores but also gives the same scores. From this we can conclude that Recursive Feature Elimination with Principal Component Analysis produces the most optimal solution.