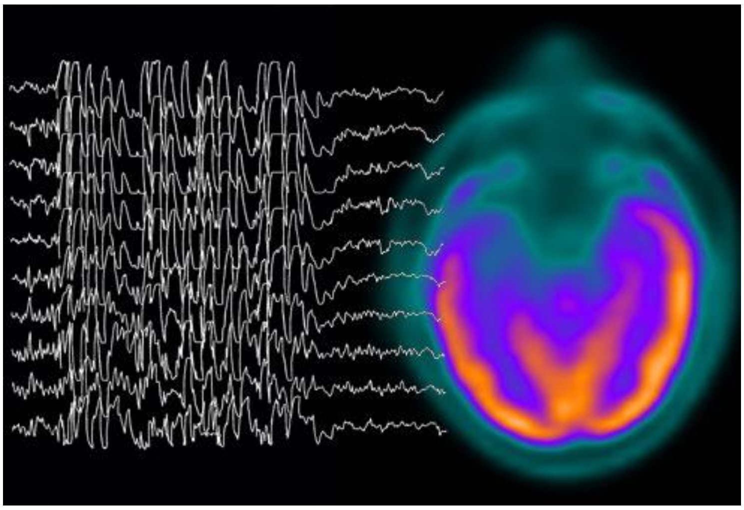


CA1-Epilepsy Classification

29-Sep-2019 | Version 1.0

Team Name: A22G

Alfred Tay Wenjie
Wang Zilong
Wong Yoke Keong

Prepared for NUS-ISS Master of Technology in Intelligent Systems ISY5002 Problem Solving Using Pattern Recognition Group Project

Table of Contents

1. Introduction

According to the Singapore Epilepsy Foundation, Epilepsy is a neurological disorder that causes sudden electrical surges in the brain that are not controllable by the patient. These brief interruptions in the brain activities cause periods of altered awareness, more commonly known as seizures whose nature and intensity levels vary in each person. In 2017, The Straits Times reported that more than 20,000 Singaporeans live with Epilepsy. Moreover, it is also estimated by SingHealth in a HealthHub article that there are approximately 150 new cases of epilepsy diagnosed in children each year. Hence the team believe that Epileptic Seizure Recognition is an area where we can apply Pattern Recognition to detect epileptic seizure accurately in order to provide timely treatment and improve the lives of these patients.

A suitable dataset that can be accessed practically is the Epileptic Seizure Recognition data set studied by Andrzejak et al and provided to the UCI Machine Learning Repository by Wu et el. This particular dataset contains the Electroencephalography (EEG) readings from 500 anonymous individuals. The data has been pre-processed from the raw signal readings by dividing and shuffling the EEG signals into 23 one-second samples from each person for a total of 11,500 observations with 178 features (X1 to X178) The target variable is 'y' with the below labels.

Target Label	Description
1	Recording of seizure activity
2	They recorded the EEG from the area where the tumor was located
3	Yes they identify where the region of the tumor was in the brain and recording the EEG activity from the healthy brain area
4	eyes closed, means when they were recording the EEG signal the patient had their eyes closed
5	eyes open, means when they were recording the EEG signal of the brain the patient had their eyes open

The original aims of the study is to detect the presence of seizure activity and hence collapsed the target variable into two broad categories: 1 - representing seizure activity as per the above table, and 0 - representing no seizure activity, collapsing categories 2 to 5 into a single class.

In the following sections, the binary classification results will be reported in alignment with original research goals. In addition, believing that the remaining labels are also highly relevant to the medical study of Epilepsy and brain electrical activity at large, the team has also further attempted 5-class classification as a challenge and documented in the respective sections.

2. Data Pre-processing

<u>Understanding the Data</u>

The dataset contains 11,500 records with 178 features. All 178 features are numerical data with no missing values. The dataset is balanced according to the 5 target labels.

Data Partitioning

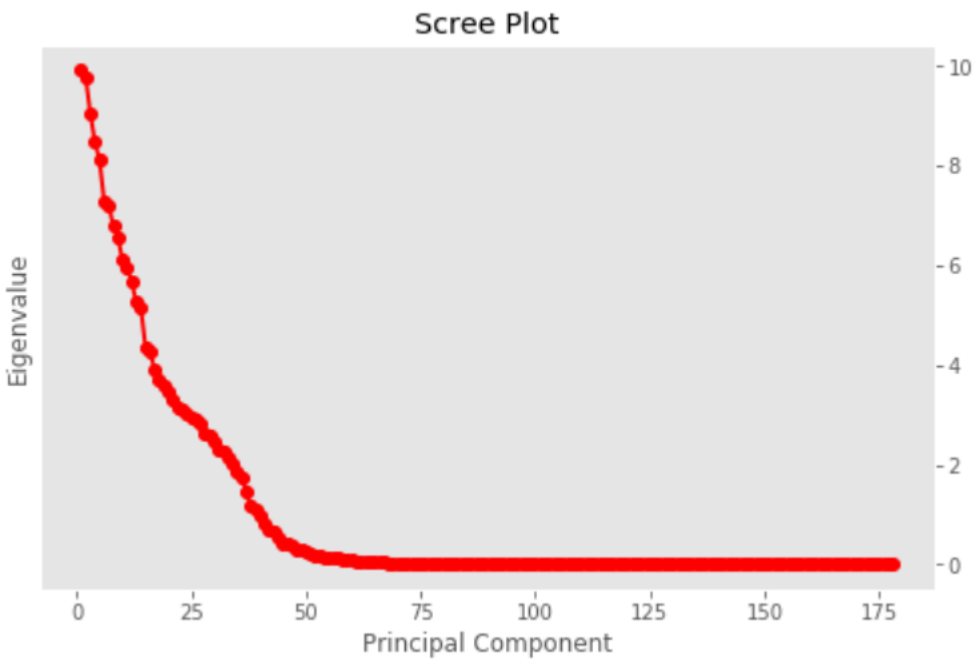
Usually split the dataset into 50/25/25, 60/20/20, 70/15/15 as the split for training/validation/test samples; this also depends on how many samples, if have an extremely huge dataset (hundreds of millions of rows) then can even use a split such as 98/1/1. This dataset has 11,500 rows and we find it appropriate to split the dataset into 70/15/15.

The dataset is partitioned and created separately for binary class classification and 5-class classification.

Dimensionality Reduction

PCA was applied to the dataset and it was discovered that 39 to 40 Principal Components could explain more than 95% of the variance in the data. Using the Kaiser rule, 39 Principal Components was used for the binary classification dataset and 40 Principal Components was used the 5-class classification dataset. Checking against the Scree Plot below, the elbow of the graph is between 35 to 45.

PCA was fitted using the training data and transformation was applied to the validation data and testing data.



Data Normalization

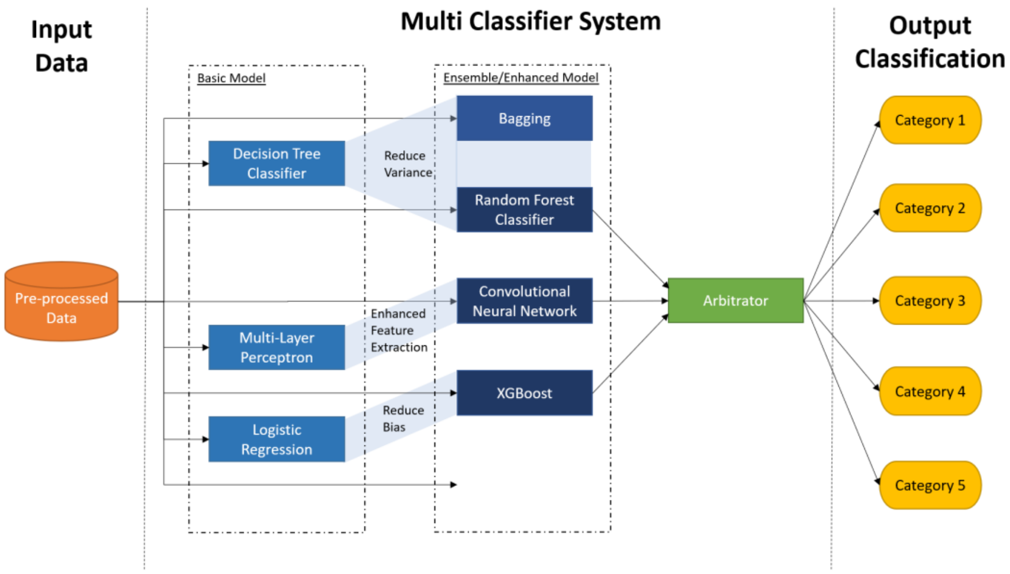
Data was normalized using Standardization (Z-score Normalization)

$$z = \frac{x - \mu}{\sigma}$$

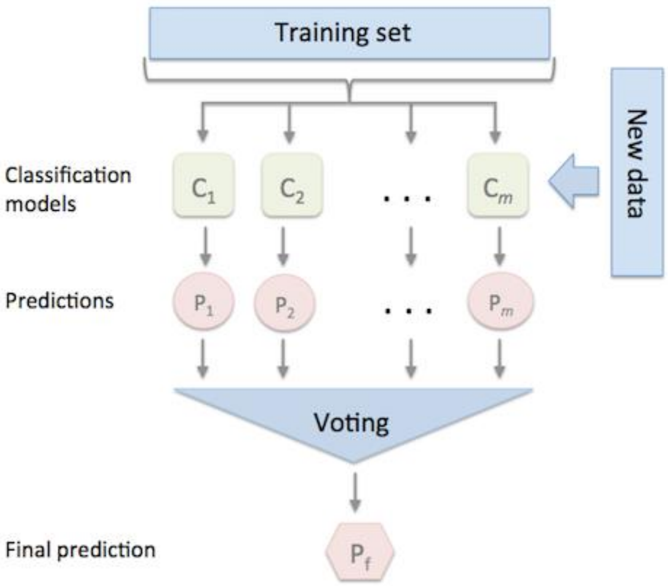
Since PCA works to maximize the variance of each principal component, it is crucial to have the variables be of the same scale. Thus this step has been applied before PCA.

3. Multi-Classifier

The final model design is a multi-classifier consisting of Random Forest, XGBoost and Convolutional Neural Network models. In the process of building up the final model, four other models have been built and tested as a comparison.



The multi-classifier is a meta-classifier for combining homogenous or heterogenous classifiers for classification as illustrated in the diagram below that takes advantages of stacking different solution experts. Stacking refers to combining predictions of several other learning algorithms. The diagram below illustrates the voting approach used for combination.



Source:https://rasbt.github.io/mlxtend/user_guide/classifier/EnsembleVoteClassifier/

'Soft' voting has been chosen in the team's implementation. In soft voting, we predict the class labels by averaging the class-probabilities according to the formula below, where w_j is the weight that can be assigned to the j^{th} classifier. Each classifier has been assigned equal weights and have the average probabilities computed

$$\hat{y} = \arg\max_i \sum_{j=1}^{m} w_j p_{ij},$$

Taking this approach allowed the team to achieve an accuracy of 74.43% that is higher than all individual models. Looking at the f1-score, there is also a general improvement from the individual models. Hence it has been determined that the voting ensemble has higher model quality in terms of accuracy and f1-score.

Best accuracy (on validation dataset): 74.43%

	precision	recall	f1-score	support
1	0.9509	0.9536	0.9522	345
2	0.6574	0.5507	0.5994	345
3	0.6313	0.6551	0.6430	345
4	0.8451	0.7275	0.7819	345
5	0.6621	0.8348	0.7385	345
accuracy			0.7443	1725
macro avg	0.7494	0.7443	0.7430	1725
weighted avg	0.7494	0.7443	0.7430	1725

```
[[329   8    4    4    0]
 [ 14  190  110   2   29]
 [  3   75  226  12   29]
 [  0    4    1  251   89]
 [  0   12   17   28  288]]
```

Hyper Parameter Tuning

Parameter Name	**Parameters for Multi-Classifier**
Number and Type of Estimators	3; Random Forest, Gradient Boosted Trees (Xgboost) and Convolution (1D) Neural Network
Voting Type	Soft
Weights	Uniform weights

Findings and Understanding

When building the multi-classifier, additional combinations have been attempted. It has been observed empirically that only some combinations work. In particular, ensembles of models with large difference in quality (in terms of accuracy and f1-score), the multi-classifier will often perform much worse than the best individual classifier.

4. Decision Tree Classifier

Decision Trees are a non-parametric supervised learning method used for classification and regression. The goal is to create a model that predicts the value of a target variable by learning simple decision rules inferred from the data features.

Suitability for Dataset

- Able to handle numerical data
- Able to handle multiple output

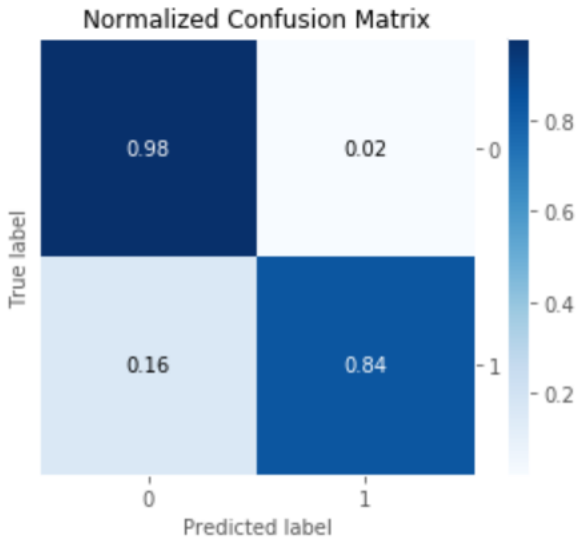
Hyper Parameter Tuning

Parameter Name	Value for Binary Classification	Value for 5-Class Classification
criterion	Entropy	Entropy
max_depth	8	None
min_samples_split	2	2
min_samples_leaf	1	2
max_features	None	None
max_leaf_nodes	None	None
min_impurity_decrease	0	0

Model Performance for Binary Classification

Best accuracy (on validation dataset): 95.30%

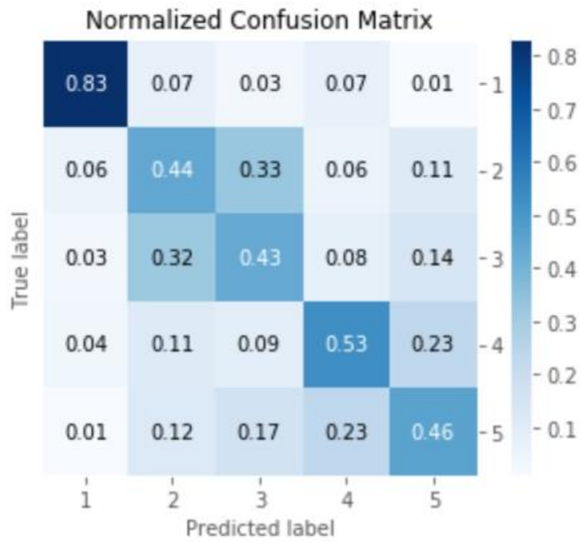
```
              precision    recall  f1-score   support

           0     0.9616    0.9804    0.9709      1380
           1     0.9151    0.8435    0.8778       345

    accuracy                         0.9530      1725
   macro avg     0.9384    0.9120    0.9244      1725
weighted avg     0.9523    0.9530    0.9523      1725
```



Model Performance for 5 Class Classification

Best accuracy (on validation dataset): 53.97%

	precision	recall	f1-score	support
1	0.8507	0.8261	0.8382	345
2	0.4169	0.4435	0.4298	345
3	0.4144	0.4348	0.4243	345
4	0.5463	0.5304	0.5382	345
5	0.4908	0.4638	0.4769	345
accuracy			0.5397	1725
macro avg	0.5438	0.5397	0.5415	1725
weighted avg	0.5438	0.5397	0.5415	1725



Findings and Understanding

The simple Decision Tree Classifier has achieved remarkably high accuracy for binary classification on the imbalanced epilepsy dataset. This suggest that the features in the dataset contains highly distinguishable pattern for Epileptic Seizure.

As the Decision Tree Classifier accepts both numerical and categorical data and is relatively easy to setup, it can be deployed as an early model to get a sensing on the dataset.

The maximum depth of the tree and the minimum number of samples required to split an internal node has been set to 8 and 2 respectively for the binary classification and the 5-class classification to avoid overfitting. The value of this hyper parameter was discovered using grid search.

5. Bagging Classifier

A Bagging Classifier is an ensemble meta-estimator that fits base classifiers each on random subsets of the original dataset and then aggregate their individual predictions (either by voting or by averaging) to form a final prediction. Such a meta-estimator can typically be used as a way to reduce the variance of a black-box estimator (e.g., a decision tree), by introducing randomization into its construction procedure and then making an ensemble out of it.

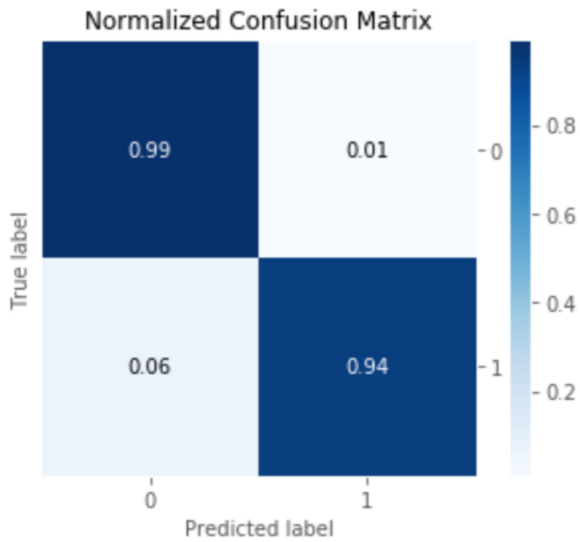
Hyper Parameter Tuning

Parameter Name	**Value for Binary Classification**	**Value for 5-Class Classification**
base_estimator	Decision Tree	Decision Tree
n_estimators	100	100
bootstrap	True	True
bootstrap_features	True	True
oob_score	False	True
warm_start	False	False

Model Performance for Binary Classification

Best accuracy (on validation dataset): 97.80%

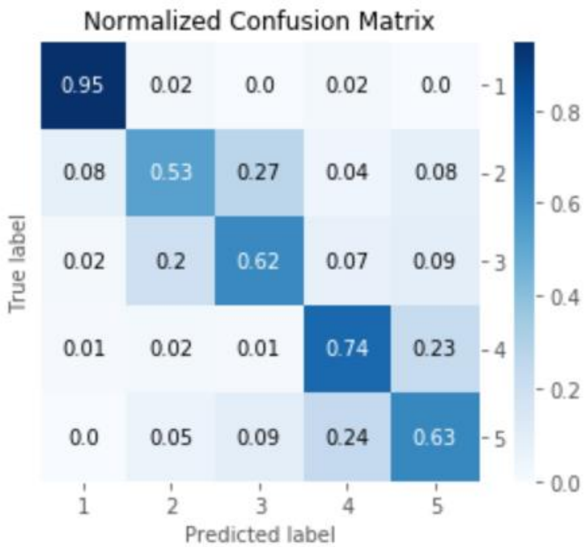
```
              precision    recall  f1-score   support

           0     0.9848    0.9877    0.9863      1380
           1     0.9501    0.9391    0.9446       345

    accuracy                         0.9780      1725
   macro avg     0.9675    0.9634    0.9654      1725
weighted avg     0.9779    0.9780    0.9779      1725
```



Model Performance for 5-Class Classification

Best accuracy (on validation dataset): 69.22%

	precision	recall	f1-score	support
1	0.8965	0.9536	0.9242	345
2	0.6408	0.5275	0.5787	345
3	0.6265	0.6174	0.6219	345
4	0.6667	0.7362	0.6997	345
5	0.6119	0.6261	0.6189	345
accuracy			0.6922	1725
macro avg	0.6885	0.6922	0.6887	1725
weighted avg	0.6885	0.6922	0.6887	1725



Findings and Understanding

The Bagging Classifier has outperformed the Decision Tree Classifier for both binary and 5 category classification. This is likely to be due to Decision Tree Classifier's tendency to be overfitted. The Bagging Classifier overcame this problem by training many models with random samples (which are subset of the training data) to create diversity and generalisation. For this dataset, the bagging classifier achieved higher accuracy when features are drawn with replacement.

6. Random Forest Classifier

A random forest is a meta estimator that fits a number of decision tree classifiers on various sub-samples of the dataset and uses averaging to improve the predictive accuracy and control over-fitting.

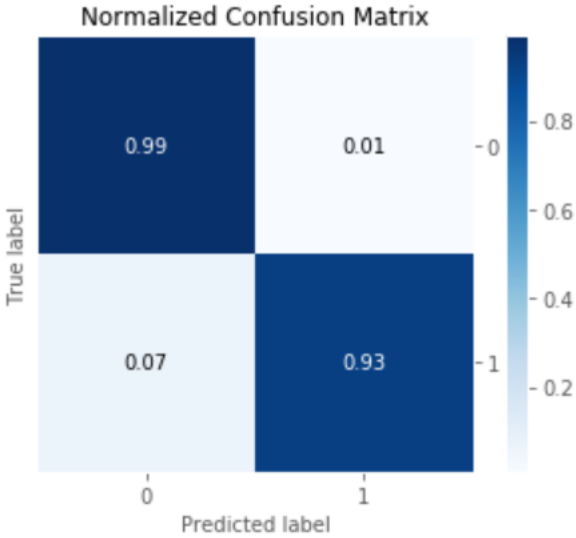
Hyper Parameter Tuning

Parameter Name	**Value for Binary Classification**	**Value for 5-Class Classification**
n_estimators	190	190
criterion	Entropy	Entropy
max_depth	None	None
min_samples_split	2	2
min_samples_leaf	2	2
max_features	None	None
max_leaf_nodes	None	None
min_impurity_decrease	0	0
bootstrap	False	False
oob_score	False	False
warm_start	False	False

Model Performance for Binary Classification

Best accuracy (on validation dataset): 97.51%

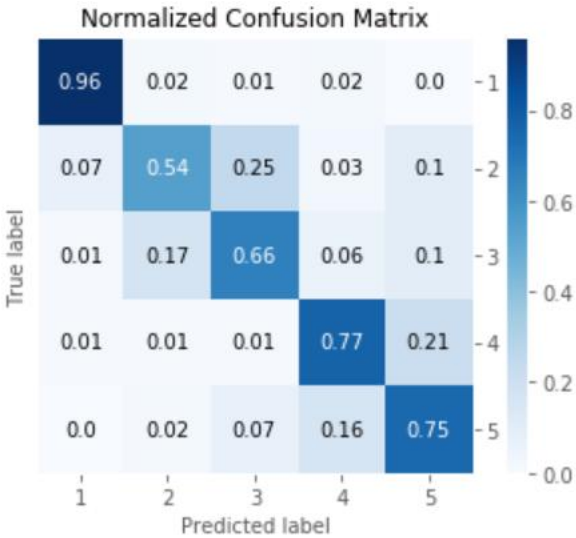
```
              precision    recall  f1-score   support

           0     0.9827    0.9862    0.9844      1380
           1     0.9441    0.9304    0.9372       345

    accuracy                         0.9751      1725
   macro avg     0.9634    0.9583    0.9608      1725
weighted avg     0.9750    0.9751    0.9750      1725
```



Model Performance for 5 Class Classification

Best accuracy (on validation dataset): 73.39%

	precision	recall	f1-score	support
1	0.9118	0.9594	0.9350	345
2	0.7083	0.5420	0.6141	345
3	0.6608	0.6551	0.6579	345
4	0.7437	0.7652	0.7543	345
5	0.6434	0.7478	0.6917	345
accuracy			0.7339	1725
macro avg	0.7336	0.7339	0.7306	1725
weighted avg	0.7336	0.7339	0.7306	1725



Findings and Understanding

For 5-class classification, the Random Forest Classifier achieved the highest accuracy among the 3 Decision Tree based Classifiers. Compared with the basic Decision Tree Classifier, it has achieved 19% higher accuracy. This implies that the Random Forest Classifier has successfully removed significant percentage of error due to variance.

7. Logistic Regression

Logistic regression, despite its name, is a linear model for classification rather than regression. Logistic regression is also known in the literature as logit regression, maximum-entropy classification (MaxEnt) or the log-linear classifier. In this model, the probabilities describing the possible outcomes of a single trial are modelled using a logistic function.

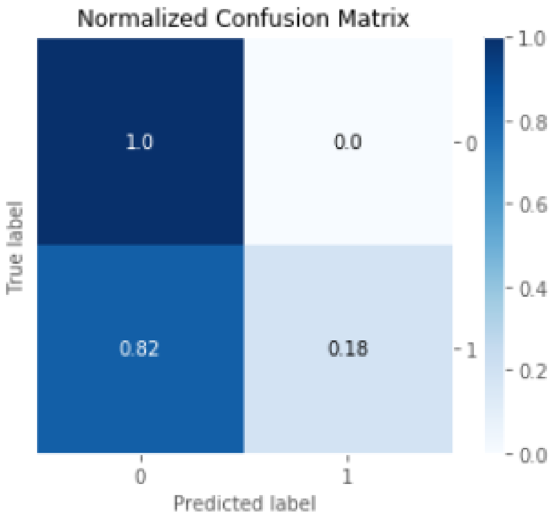
Hyper Parameter Tuning

Parameter Name	**Value for Binary Classification**	**Value for 5-Class Classification**
penalty (regularization)	L2	L2
tol (Tolerance for stopping criteria)	1e-4	1e-4
C (Inverse of regularization strength)	1e-3	1e-3
solver	liblinear	lbfgs
max_iter	100	yes
multi_class	N.A.	multinomial

Model Performance for Binary Classification

Accuracy on validation dataset: 83.30%

	precision	recall	f1-score	support
0	0.8293	0.9964	0.9052	1380
1	0.9254	0.1797	0.3010	345
accuracy			0.8330	1725
macro avg	0.8773	0.5880	0.6031	1725
weighted avg	0.8485	0.8330	0.7844	1725

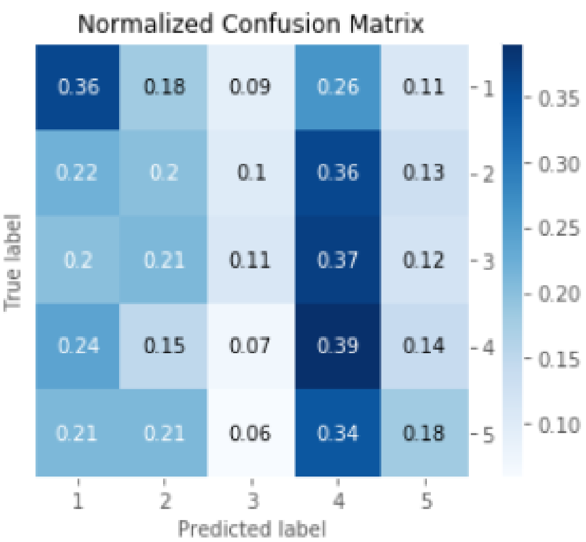


Model Performance for 5-Class Classification

Accuracy on validation dataset: 24.81%

Best accuracy (on valid dataset): 24.81%

	precision	recall	f1-score	support
1	0.2948	0.3623	0.3251	345
2	0.2061	0.1971	0.2015	345
3	0.2635	0.1130	0.1582	345
4	0.2288	0.3913	0.2888	345
5	0.2618	0.1768	0.2111	345
accuracy			0.2481	1725
macro avg	0.2510	0.2481	0.2369	1725
weighted avg	0.2510	0.2481	0.2369	1725



Findings and Understanding

Parameter C (Inverse of regularization strength) - The effect of using different regularization values is obvious. Bigger C values make model increase its complexity and adjust better to the data. Small C values will increase the regularization strength.

Parameter multi_class - Multinomial logistic regression is a form of logistic regression used to predict a target variable have more than 2 classes. It is a modification of logistic regression using the Softmax function instead of the sigmoid function. The training algorithm uses the one-vs-rest (OvR) scheme if the 'multi_class' option is set to 'ovr', and uses the cross-entropy loss if the 'multi_class' option is set to 'multinomial'. The issue with Softmax function is that it blows small differences out of proportion which makes our classifier biased towards a particular class which is not desired.

Logistic regression is essentially a linear classifier, it can't handle the problem of feature correlation. When the feature space is large, the performance is not good and it is easy to underfit and accuracy is not high as seen in the 5-class classification.

8. Gradient Boost

Gradient boosting is a machine learning technique for regression and classification problems, which produces a prediction model in the form of an ensemble of weak prediction models, typically decision trees. It builds the model in a stage-wise fashion like other boosting methods do, and it generalizes them by allowing optimization of an arbitrary differentiable loss function.

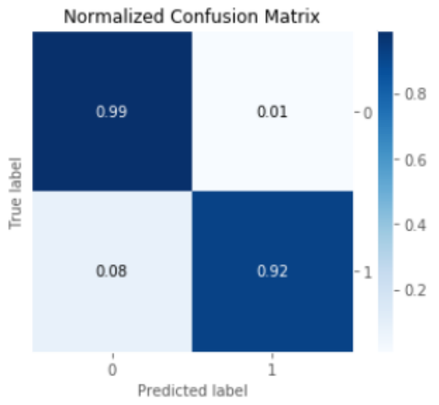
Hyper Parameter Tuning

Parameter Name	Value for Binary Classification (tree booster)	Value for 5-Class Classification (tree booster)	Value for 5-Class Classification (linear booster)
n_estimator	100	1000	1000
learning_rate	0.1	0.1	0.1
max_depth	5	5	5
objective	binary:logistic	Multi: softmax	Multi: softmax
Booster	gbtree	gbtree	gblinear

Model Performance for Binary Classification

Accuracy on validation dataset: 97.39%

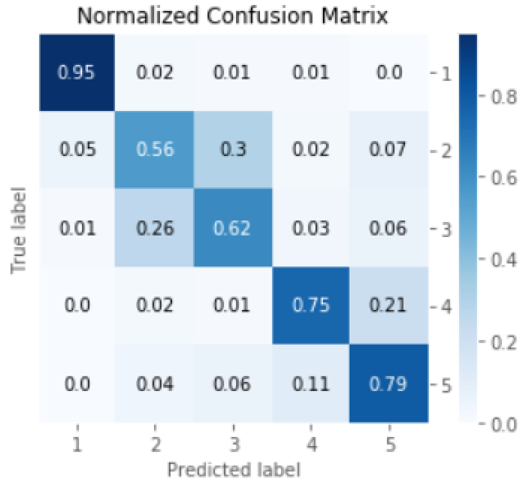
```
              precision    recall  f1-score   support

           0     0.9799    0.9877    0.9838      1380
           1     0.9491    0.9188    0.9337       345

    accuracy                         0.9739      1725
   macro avg     0.9645    0.9533    0.9587      1725
weighted avg     0.9737    0.9739    0.9738      1725
```



Model Performance for 5-Class Classification (gbtree)

Accuracy on validation dataset: 73.57%

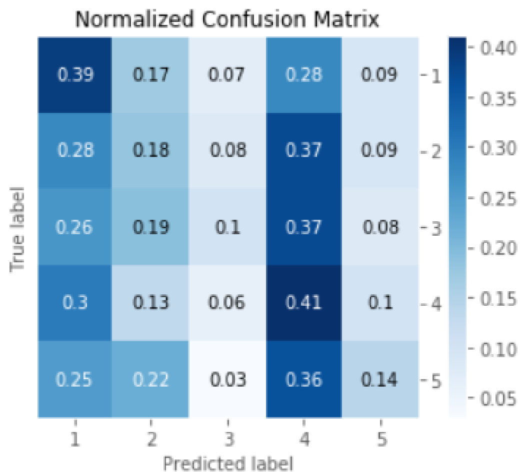
	precision	recall	f1-score	support
1	0.9316	0.9478	0.9397	345
2	0.6166	0.5594	0.5866	345
3	0.6214	0.6232	0.6223	345
4	0.8100	0.7536	0.7808	345
5	0.6954	0.7942	0.7415	345
accuracy			0.7357	1725
macro avg	0.7350	0.7357	0.7342	1725
weighted avg	0.7350	0.7357	0.7342	1725



Model Performance for 5-Class Classification (gblinear)

Accuracy on validation dataset: 24.23%

	precision	recall	f1-score	support
1	0.2633	0.3884	0.3138	345
2	0.1981	0.1768	0.1868	345
3	0.3000	0.1043	0.1548	345
4	0.2265	0.4058	0.2908	345
5	0.2765	0.1362	0.1825	345
accuracy			0.2423	1725
macro avg	0.2529	0.2423	0.2258	1725
weighted avg	0.2529	0.2423	0.2258	1725



<u>Findings and Understanding</u>

XGBoost is similar to gradient boosting except:

- Trees have a varying number of terminal nodes
- Leaf weights of the trees that are calculated with less evidence is shrunk more heavily
- Newton Boosting provides a direct route to the minima than gradient descent
- Extra randomization parameter is used to reduce the correlation between trees
- Uses a more regularized model to control over-fitting since standard GBM has no regularization, which gives it better performance over GBM.
- XGB implements parallel processing, and it is much faster than GBM

Tree Booster VS Linear Booster

- gbtree booster uses version of regression tree as a weak learner
- gblinear uses (generalized) linear regression with l1&l2 shrinkage. But since it's an additive process, and since linear regression is an additive model itself, only the combined linear model coefficients are retained

9. Multi-Layer Perceptron

The Multi-Layer Perceptron (MLP) is considered a deep, artificial neural network [1] that is composed of 1 input layer, 1 output layer and multiple hidden layers consisting of multiple perceptron units.

A perceptron is a linear classifier i.e. providing classification based on a linear combination of input units according to the formula below, where **w** denotes the vector of weights, **x** is the vector of inputs, **b** is the bias and **φ** is the non-linear activation function.

$$y = \varphi\left(\sum_{i=1}^{n} w_i x_i + b\right) = \varphi(\mathbf{w}^T \mathbf{x} + b)$$

MLP is capable of learning complex non-linear functions typically applied for supervised learning such as classification and regression. It learns by calculating the error difference between the predicted value and ground truth value during the forward pass and propagates error gradients through backpropagation during the backward pass and allows the optimizer to update weights accordingly.

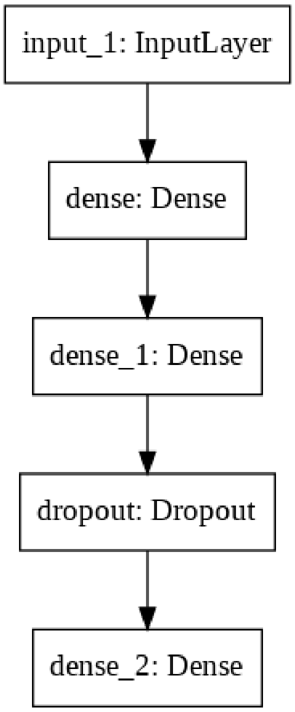
The model is implemented using Keras rather than scikit-learn. This is to take advantage of GPU processing speed-ups for neural networks, that enables rapid prototyping iterations.

Hyper Parameter Tuning

Parameter Name	**Value for Binary Classification**	**Value for 5-Class Classification**
Input Layer Features	39	40
Hidden Layers	Dense	Dense
Hidden Layer Activation	relu	relu
Output Layer	Units: 1	Units: 5
Output Layer Activation	sigmoid	softmax
Regularization used	Dropout	Dropout
Loss	cross entropy	cross entropy
Optimizer	adam	adam
Learning Rate	0.001	0.001
Epochs	40	50
Batch Size	64	32

Model Performance for Binary Classification

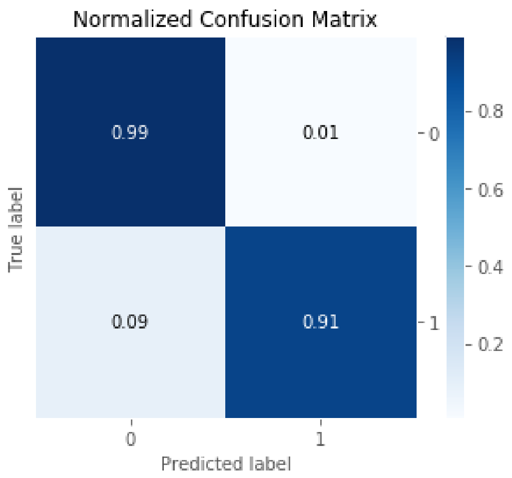
The below diagram shows the network architecture for the binary classifier.



Using the above architecture, an accuracy of 97.68% is achieved with ROC score of 0.9935118672547784.

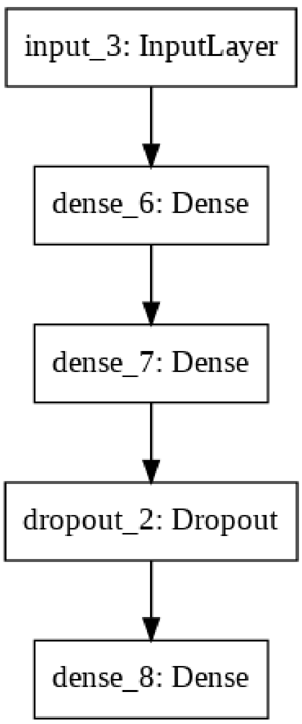
```
Best accuracy (on validation dataset): 97.68%
              precision    recall  f1-score   support

           0     0.9841    0.9870    0.9855      1380
           1     0.9472    0.9362    0.9417       345

    accuracy                         0.9768      1725
   macro avg     0.9657    0.9616    0.9636      1725
weighted avg     0.9767    0.9768    0.9768      1725

[[1362   18]
 [  22  323]]
```

The normalized confusion matrix shows the proportion of correctly and incorrectly classified classes.



Model Performance for 5-Class Classification

The below diagram shows the network architecture for the 5-class classifier. This is the same architecture used for binary classification.

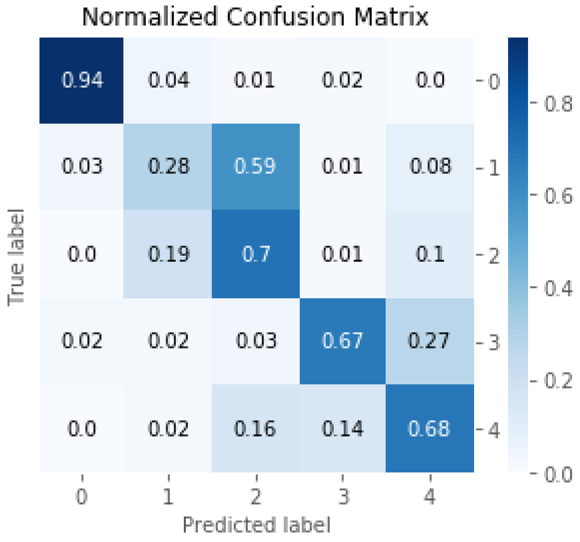


Using the above architecture, the accuracy dropped to 65.22%.

```
Best accuracy (on validation dataset): 65.22%
              precision    recall  f1-score   support

           0     0.9472    0.9362    0.9417       345
           1     0.5131    0.2841    0.3657       345
           2     0.4697    0.6957    0.5607       345
           3     0.7884    0.6696    0.7241       345
           4     0.5990    0.6754    0.6349       345
```

```
    accuracy                                0.6522      1725
   macro avg      0.6635     0.6522     0.6454      1725
weighted avg      0.6635     0.6522     0.6454      1725

[[323  14   2   6   0]
 [ 12  98 204   3  28]
 [  0  66 240   3  36]
 [  6   7   9 231  92]
 [  0   6  56  50 233]]
```

As shown below, the normalized confusion matrix shows that there are more incorrectly classified classes, in particular for label 1, only 28% is correctly classified while 59% is incorrectly classified as label 2. Given this is a 5-class prediction, it can be considered a random guess, which can be further improved.



<u>Findings and Understanding</u>

In the case of binary classification, in addition to the high accuracy score obtained, it is observed that it also has the highest ROC AUC score of 99.3% amongst the other binary classifiers despite the imbalanced dataset.

Given that the ROC AUC score measures the performance of the classifier across various True Positive Rate vs False Positive Rate thresholds, it can be deduced that the Multi-Layer Perceptron has the best model quality compared to the other models tested. Hence it is chosen for final testing in the upcoming section on Test Set scores.

In the case of 5-class classification, the same MLP model has a much lower accuracy. It can be deduced that in this multi-class scenario, MLP is unable to discern patterns very well. Hence an intuition can be to improve feature extraction, which can be introduced with Convolution 1D layers.

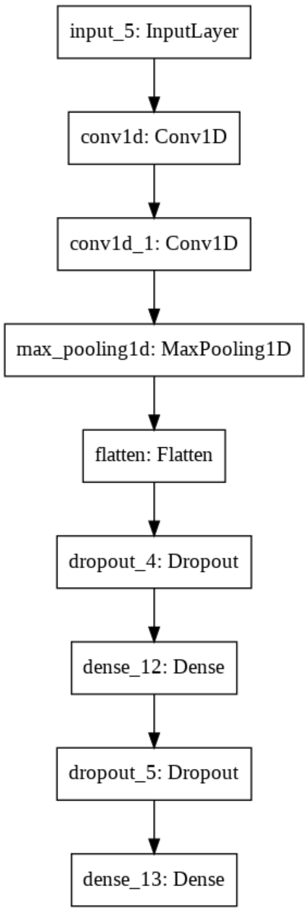
10. Convolutional Neural Network

Compared to the MLP model in the previous section, the Convolution Neural Network (CNN) here has additional Convolution and Max Pooling layers that is used to learn features from data set before feeding these learned features into the remaining hidden units.

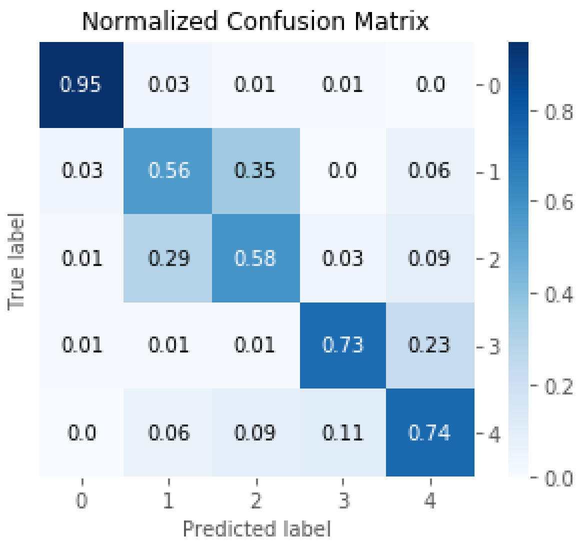
Hyper Parameter Tuning

Parameter Name	Value for 5-Class Classification
Input Layer Features	40,1
Hidden Layers and other functions	Dense, Convolution 1D, MaxPooling1D
Hidden Layer Activation	relu
Output Layer	Units: 5
Output Layer Activation	softmax
Regularization used	Dropout
Loss	cross entropy
Optimizer	adam
Learning Rate	0.001
Epochs	60
Batch Size	32

Model Performance for 5-Class Classification



```
Best accuracy (on validation dataset): 71.48%
              precision    recall  f1-score   support

           0     0.9563    0.9507    0.9535       345
           1     0.5897    0.5623    0.5757       345
           2     0.5568    0.5826    0.5694       345
           3     0.8322    0.7333    0.7797       345
           4     0.6624    0.7449    0.7012       345

    accuracy                         0.7148      1725
   macro avg     0.7195    0.7148    0.7159      1725
weighted avg     0.7195    0.7148    0.7159      1725

[[328  10   5   2   0]
 [  9 194 122   0  20]
 [  3 101 201  10  30]
 [  3   5   3 253  81]
 [  0  19  30  39 257]]
```



<u>Findings and Understanding</u>

With the introduction of additional Convolution-1D and Maxpooling-1D layers, the accuracy increased from 65.22% to 71.48%. This shows that some useful feature extraction has been learned. The proportion of correctly classified class 1 has increased to 56%, which is better than random guess. Looking at the f1-score, all classes have improved with the exception of class 3, indicating better model quality. Hence this model is considered for voting ensemble mentioned in the early sections.

11. Testing and Comparing Models

After the training for all the models was completed with the training and validation data, the models were run using the reserved test data.

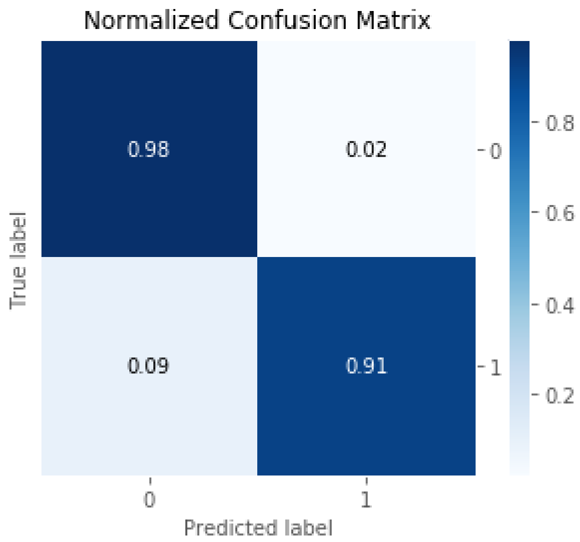
Model Performance for binary Classification

Comparison of the best classifier has been done based on the ROC AUC score. Amongst the classifier types below, the Multi-Layer Perceptron has the highest score, indicating the best model quality amongst these classifiers.

Binary Classifier Type	**ROC AUC Score (3 decimal places)**
Decision Tree	0.912
Bagging	0.963
Random Forest	0.957
Logistic Regression	0.588
Multi-Layer Perceptron	0.994

The classifier is then evaluated on the test set which is put aside as the unseen data set. An accuracy of **96.93%** has been achieved, with high f1-score for class 1, that is the classification that the patient has epilepsy.

```
Best accuracy (on testing dataset): 96.93%
              precision    recall  f1-score   support

           0     0.9777    0.9841    0.9809      1380
           1     0.9345    0.9101    0.9222       345

    accuracy                         0.9693      1725
   macro avg     0.9561    0.9471    0.9515      1725
weighted avg     0.9691    0.9693    0.9691      1725

[[1358   22]
 [  31  314]]
```



Model Performance for 5-Class Classification

In the case of 5-class classification, an accuracy of **74.61%** has been achieved, which is similar to that of the validation set. The f1-score reflects similarly. Hence, it can be deduced that while accuracy is not as high, the model is well-calibrated i.e. performing similarly on the unseen test set.

```
Best accuracy (on testing dataset): 74.61%
              precision    recall  f1-score   support

           1     0.9307    0.9739    0.9518       345
           2     0.6486    0.5565    0.5991       345
           3     0.6017    0.6087    0.6052       345
           4     0.8459    0.7797    0.8115       345
           5     0.6983    0.8116    0.7507       345

    accuracy                         0.7461      1725
   macro avg     0.7451    0.7461    0.7436      1725
weighted avg     0.7451    0.7461    0.7436      1725

[[336   6   1   2   0]
 [ 15 192 115   6  17]
 [  8  79 210  13  35]
 [  2   4   1 269  69]
 [  0  15  22  28 280]]
```

12. Conclusion

The team started out to create several binary classification models for Epilepsy and has shown that the best model for this dataset is the Multi-Layer Perceptron model as measured by the ROC AUC score. This model achieved an accuracy of **96.93%**, with a high f1-score of 0.9222 for the label of interest on the unseen test set. Hence the team have high confidence that the model will work well.

The team subsequently took up the challenge of creating a 5-class classification model and demonstrated that the heterogeneous multi-classifier model out-performs the other homogeneous ensemble models, reaching an accuracy of **74.61%** on the test set.

Throughout the assignment, the team also went through the entire machine learning cycle end-to-end and learned how to apply dimension reduction techniques like PCA successfully.

Reference

Andrzejak RG, Lehnertz K, Rieke C, Mormann F, David P, Elger CE (2001) Indications of nonlinear deterministic and finite dimensional structures in time series of brain electrical activity: Dependence on recording region and brain state, Phys. Rev. E, 64, 061907

UCI Machine Learning Repository, Qiuyi Wu, Ernest Fokoue (School of Mathematical Sciences

Rochester Institute of Technology) (2017, May 24) Epileptic Seizure Recognition Data Set: https://archive.ics.uci.edu/ml/datasets/Epileptic+Seizure+Recognition

The Straits Times (2017, February 21) Optimism in the face of epilepsy: https://www.straitstimes.com/singapore/health/optimism-in-the-face-of-epilepsy

Singapore Epilepsy Foundation (2019, September 15) What is Epilepsy?: http://www.epilepsy.com.sg/epilepsy.html

Healthhub SG (2019, April 18) Epilepsy in Children: https://www.healthhub.sg/a-z/diseases-and-conditions/706/managing-epilepsy-in-children

Skymind AI (2019, September 25) ,Multi-layer Perceptron: https://skymind.ai/wiki/multilayer-perceptron

mlxtend user guide (2019, September 25), EnsembleVoteClassifier: https://rasbt.github.io/mlxtend/user_guide/classifier/EnsembleVoteClassifier/

Scikit-Learn Documentation (2019, September 29) Logistic Regression: https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression

Scikit-Learn Documentation (2019, September 29) Decision Trees: https://scikit-learn.org/stable/modules/tree.html#tree

Scikit-Learn Documentation (2019, September 29) Bagging Classifier: https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.BaggingClassifier.html

Scikit-Learn Documentation (2019, September 29) Random Forest Classifier: https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html#sklearn.ensemble.RandomForestClassifier

Wikipedia (2019, September 2019) Gradient Boosting: https://en.wikipedia.org/wiki/Gradient_boosting