## Task 2: Explain your analysis of the code. Make a detailed analysis that can also cover the following questions: (Submit the PDF of the Report)

## Code Analysis:

1. Program for understanding Overfitting and Underfitting

Firstly, we import all the libraries needed. A true\_fun is defined which is a cosine function. Then a random sample is taken and sorted and passed through the true\_fun function. Then apply polynomial features and linear regression model to the data. Then pipeline the polynomial features and linear regression. Finally, evaluate the model using cross-validation. Plot the graphs for degree 1, degree 4, and degree 15 using MSE. The function is not sufficient to fit the training samples for degree 1, this is called **underfitting**. Degree 4 polynomial fits perfectly without underfitting or overfitting. When the degree increases, the model will **overfit** the training data by learning the noise of the training data. This is called overfitting.

Graphical user interface, application

Description automatically generated

Graphical user interface, text, application

Description automatically generated

Graphical user interface, chart, application, histogram

Description automatically generated

1. Overfitting (Printing accuracy at different steps)

Here make\_classification function is used for problems with 10000 samples and 20 features. Dataset is split into training and testing data. A decision tree is used as a classifier, tree depth is adjusted. Test and training data are evaluated, and accuracies are printed at different depths. As the tree depth increases the performance increases, and the shallow trees have low performance. That is the reason the shallow trees do not overfit and extremely deeper trees overfit. As shown below, until depth 5 the accuracy of the test set increases, and later it decreases but the accuracy of the training sets increases until the maximum depth, this is called overfitting. This can be solved by decreasing the depth of the tree.

Graphical user interface, text, application, email

Description automatically generated

Graphical user interface, text, application, email

Description automatically generatedA picture containing text

Description automatically generated

1. Cross Validation

Here, we are using dataset load\_iris. We use cross-validation to evaluate overfitting and underfitting.

A picture containing rectangle

Description automatically generated

Firstly, we use the basic method to compute the score using svm.

Graphical user interface, text, application

Description automatically generated

Here, the accuracy is estimated by splitting the data, fitting the model, and evaluating the score 5 times. The score is calculated at each iteration.

Graphical user interface, text, application, email

Description automatically generated

Here, cross\_val\_score is used to cross-validate the score.

Graphical user interface, text, application, chat or text message

Description automatically generated

Here, the cross-validate iterator is used to calculate the score.

Graphical user interface, text, application, chat or text message, Teams

Description automatically generated

Graphical user interface, text

Description automatically generated

1. Different types of Cross validations: Below are a few cross-validation types.

K-fold divides all the samples into k groups of samples(folds) of equal sizes. The prediction function is learned using k−1 folds, and the fold left out is used for testing.

Text

Description automatically generated

Repeated K-fold repeats K-Fold n times. It can be used when one requires to run kfold, n times, producing different splits in each iteration.

A picture containing scatter chart

Description automatically generated

In LOO, learning sets are created by taking all the samples leaving only one, which is the test set.

A picture containing calendar

Description automatically generated

LPO creates all the possible training/test sets by removing p samples from the complete set. For n samples, this produces (np) train-test pairs.

Calendar

Description automatically generated with medium confidence

Stratified k-fold is a variation of k-fold which returns stratifiedfolds, each set contains approximately the same percentage of samples of each target class as the complete set.

Graphical user interface, text

Description automatically generated

Stratified Shuffle Split is a variation of Shuffle Split, which returns stratified splits i.e., which creates splits by preserving the same percentage for each target class as in the complete set

A picture containing scatter chart

Description automatically generated

A picture containing calendar

Description automatically generated

1. Validation Curve

The influence of a single hyperparameter on the training score and the validation score is used to find out if the estimator is overfitting or underfitting for some hyperparameter values. If the training score and the validation score are both low, the estimator will be underfitting. If the training score is high and the validation score is low, the estimator is overfitting.

Here, the train and test data CSV files are trained using a random forest classifier and data is fit, then used to draw a validation curve.

Graphical user interface, application

Description automatically generated

Graphical user interface, text, application

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated

Graphical user interface, application

Description automatically generated

Graphical user interface, text, application

Description automatically generated

A picture containing diagram

Description automatically generated

1. ROC

The two main metrics of true positive rate and two negative rates are visualized using a ROC.

Data is fetched from the GitHub URL; a logical regression model is used, and data is fit. The metrics are defined, and the ROC curve is drawn. If the curve touches the top left corner of the plot, the model is considered better at classifying the data into categories.

Graphical user interface, text, application

Description automatically generated

Graphical user interface, application

Description automatically generated

1. According to you, why do overfitting and underfitting occur, and how resolve them? What is the difference between them?

When the model tries to capture the relation between the input and target values very efficiently leading to the memorization of the data, the model does not perform well on the testing data but performs well on the training data.

When the model performs poorly on training data, it is called underfitting. This is because the model is unable to capture the relationship between the input examples and the target values.

There are a few common ways to control the overfitting of data i.e., by Adding more data, data augmentation, regularization, and removing features from the data. In the code, we have regulated the depth of the decision tree to control overfitting. Or for the KNN classifier, we can change the neighbor number to control overfitting.

The underfitting problem can be solved by increasing the model complexity, reducing regularization, and adding features to training data.

**A model that is under fitted will have high training and high testing error while an overfit model will have extremely low training error but a high testing error.**

1. What kind of pattern did you analyze in the Train and Test score while running the code of overfitting?

Test and training data are evaluated, and accuracies are printed at different depths. As the tree depth increases the performance increases, and the shallow trees have low performance. That is the reason why the shallow trees do not overfit and extremely deeper trees overfit, until depth 5 the accuracy of the test set increases, and later it decreases but the accuracy of the training sets increases until the maximum depth, this is called overfitting. This can be solved by decreasing the depth of the tree.

1. What is cross-validation, and what did you analyze in a different type of validation that you performed?

To avoid overfitting, sometimes available data is divided into three parts training, testing, and validation data, as the size of the samples used for learning decreases, the results depend on a particular random pair of training and validation data, to solve this problem

cross-validation is used.

* K-fold divides all the samples into k groups of samples(folds) of equal sizes. The prediction function is learned using k−1 folds, and the fold left out is used for testing.
* Repeated K-fold repeats K-Fold n times. It can be used when one requires to run kfold, n times, producing different splits in each iteration.
* In LOO, learning sets are created by taking all the samples leaving only one, which is the test set.
* LPO creates all the possible training/test sets by removing p samples from the complete set. For n samples, this produces (np) train-test pairs.
* Stratified k-fold is a variation of k-fold which returns stratifiedfolds, each set contains the same percentage of samples of each target class as the complete set.
* Stratified Shuffle Split is a variation of Shuffle Split, which returns stratified splits i.e., which creates splits by preserving the same percentage for each target class as in the complete set.

1. Explain the analysis from generated ROC and validation curve and what they represent?

Validation Curve:

To estimate the generalization accurately we must compute the score on another test set. If the training score and the validation score are both low, the estimator will be underfitting. If the training score is high and the validation score is low, the estimator is overfitting and otherwise, it is working very well. A low training score and a high validation score are usually not possible.

ROC Curve:

The two main metrics, true positive rate, and true negative rates are visualized using a ROC. If the curve touches the top left corner of the plot, the model is considered better at classifying the data into categories.