Worksheet_set_1

MACHINE LEARNING

In Q1 to Q7, only one option is correct, Choose the correct option:

1. What is the advantage of hierarchical clustering over K-means clustering?
A) Hierarchical clustering is computationally less expensive
B) In hierarchical clustering you don't need to assign number of clusters in beginning
C) Both are equally proficient
D) None of these

Ans: B

2. Which of the following hyper parameter(s), when increased may cause random forest to over fit the data?

A) max_depth B) n_estimators
C) min_samples_leaf D) min_samples_splits

Ans: A

3. Which of the following is the least preferable resampling method in handling imbalance datasets?

A) SMOTE B) RandomOverSampler
C) RandomUnderSampler D) ADASYN

Ans: D

4. Which of the following statements is/are true about "Type-1" and "Type-2" errors?
1. Type1 is known as false positive and Type2 is known as false negative.
2. Type1 is known as false negative and Type2 is known as false positive.
3. Type1 error occurs when we reject a null hypothesis when it is actually true.

A) 1 and 2 B) 1 only
C) 1 and 3 D) 2 and 3

Ans: C

5. Arrange the steps of k-means algorithm in the order in which they occur:
1. Randomly selecting the cluster centroids
2. Updating the cluster centroids iteratively
3. Assigning the cluster points to their nearest center

A) 3-1-2 B) 2-1-3
C) 3-2-1 D) 1-3-2

Ans: D

6. Which of the following algorithms is not advisable to use when you have limited CPU resources and time, and when the data set is relatively large?

A) Decision Trees B) Support Vector Machines
C) K-Nearest Neighbors D) Logistic Regression

Ans: B

7. What is the main difference between CART (Classification and Regression Trees) and CHAID (Chi Square Automatic Interaction Detection) Trees?

A) CART is used for classification, and CHAID is used for regression.
B) CART can create multiway trees (more than two children for a node), and CHAID can only create binary trees (a maximum of two children for a node).
C) CART can only create binary trees (a maximum of two children for a node), and CHAID can create multiway trees (more than two children for a node)
D) None of the above

Ans: C

In Q8 to Q10, more than one options are correct, Choose all the correct options:

8. In Ridge and Lasso regularization if you take a large value of regularization constant(lambda), which of the following things may occur?

A) Ridge will lead to some of the coefficients to be very close to 0
B) Lasso will lead to some of the coefficients to be very close to 0
C) Ridge will cause some of the coefficients to become 0
D) Lasso will cause some of the coefficients to become 0.

Ans: A,B

9. Which of the following methods can be used to treat two multi-collinear features?

A) remove both features from the dataset
B) remove only one of the features
C) Use ridge regularization
D) use Lasso regularization

Ans: C,D

10. After using linear regression, we find that the bias is very low, while the variance is very high. What are the possible reasons for this?

A) Overfitting B) Multicollinearity
C) Underfitting D) Outliers

Ans: A,C

Q10 to Q15 are subjective answer type questions, Answer them briefly.

11. In which situation One-hot encoding must be avoided? Which encoding technique can be used in such a case?

Ans: One-hot Encoding is a feature encoding technique to convert categorical features into a numerical . For each feature value, the one-hot transformation creates a new feature demarcating the presence or absence of feature value.

- When the categorical features present in the dataset are ordinal .For e.g Junior, Senior, Executive, Owner.
- When the number of categories in the dataset is quite large. One Hot Encoding should be avoided ,which can lead to high memory consumption.
- Time-based features such as **day of month**, **day of week**, **day of year**, etc have a cyclic nature and have many feature values. One-hot encoding **day of month** feature results in 30 dimensionality vector, **day of year** results in 365 dimension vector.

The solution to encode these cyclic features can be using mathematical formulation and trigonometry. In this article, we will encode the cyclic features using the basic formulation of trigonometry, by computing the sin and cosine of the features.

12. In case of data imbalance problem in classification, what techniques can be used to balance the dataset? Explain them briefly.

Ans: Classification problems are quite common in the machine learning world. As we know in the classification problem we try to predict the class label by studying the input data or predictor where the target or output variable is a categorical variable in nature.

In certain classification problems we have faced instances where one of the target class labels' numbers of observation is significantly lower than other class labels. This type of dataset is called an **imbalanced class dataset** which is very common in classification scenarios.

Imbalanced data refers to those types of datasets where the target class has an uneven distribution of observations, i.e one class label has a very high number of observations and the other has a very low number of observations.

For e.g : we are going to predict disease from an existing dataset where for every 100 records only 5 patients are diagnosed with the disease. So, the majority class is 95% with no disease and the minority class is only 5% with the disease. Now, assume our model predicts that all 100 out of 100 patients have no disease.

Sometimes when the records of a certain class are much more than the other class, our classifier may get biased towards the prediction. In this case, the confusion matrix for the classification problem shows how well our model classifies the target classes and we arrive at the accuracy of the model from the confusion matrix. It is calculated based on the total no of correct predictions by the model divided by the total no of predictions. In the above case it is (0+95)/(0+95+0+5)=0.95 or 95%. It means that the model fails to identify the minority class yet the accuracy score of the model will be 95%.

Thus our traditional approach of classification and model accuracy calculation is not useful in the case of the imbalanced dataset.

Techniques used to balance the datasets:

i) **Choose Proper Evaluation Metric** : The accuracy of a classifier is the total number of correct predictions by the classifier divided by the total number of predictions. This may be good enough for a well-balanced class but not ideal for the imbalanced class problem. The other metrics such as precision is the measure of how accurate the classifier's prediction of a specific class and recall is the measure of the classifier's ability to identify a class.

ii) **Resampling (Oversampling and Undersampling)**: This technique is used to upsample or downsample the minority or majority class. When we are using an imbalanced dataset, we can oversample the minority class using replacement. This technique is called oversampling. Similarly, we can randomly delete rows from the majority class to match them with the minority class which is called undersampling. After sampling the data we can get a balanced dataset for both majority and minority classes. So, when both classes have a similar number of records present in the dataset, we can assume that the classifier will give equal importance to both classes

iii) **Synthetic Minority Oversampling Technique** or **SMOTE**: **Synthetic Minority Oversampling Technique** or **SMOTE** is another technique to oversample the minority class. Simply adding duplicate records of minority class often don't add any new information to the model. In SMOTE new instances are synthesized from the existing data. If we explain it in simple words, SMOTE looks into minority class instances and use *k* nearest neighbor to select a random nearest neighbour, and a synthetic instance is created randomly in feature space.

iv) **BalancedBaggingClassifier:** When we try to use a usual classifier to classify an imbalanced dataset, the model favors the majority class due to its larger volume presence. A BalancedBaggingClassifier is the same as a sklearn classifier but with additional balancing. It includes an additional step to balance the training set at the time of fit for a given sampler. This classifier takes two special parameters "sampling_strategy" and "replacement". The sampling_strategy decides the type of resampling required (e.g. 'majority' – resample only the majority class, 'all' – resample all classes, etc) and replacement decides whether it is going to be a sample with replacement or not.

v) **Threshold moving**: In the case of our classifiers, many times classifiers actually predict the probability of class membership. We assign those prediction's probabilities to a certain class based on a threshold which is usually 0.5, i.e. if the probabilities < 0.5 it belongs to a certain class, and if not it belongs to the other class.

For imbalanced class problems, this default threshold may not work properly. We need to change the threshold to the optimum value so that it can efficiently separate two classes. We can use ROC Curves and Precision-Recall Curves to find the optimal threshold for the classifier. We can also use a grid search method or search within a set of values to identify the optimal value.

13. What is the difference between SMOTE and ADASYN sampling techniques?

14. What is the purpose of using GridSearchCV? Is it preferable to use in case of large datasets? Why or why not?

Ans: GridSearchCV is a library function that is a member of sklearn's model_selection package. It helps to loop through predefined hyperparameters and fit our model on training set. So, in the end, we can select the best parameters from the listed hyperparameters.
In addition to that, we can specify the number of times for the cross-validation for each set of hyperparameters.

Following are the simple steps to carry out:

- The first step is to define the hyperparameters we want to try out. It is depending on the estimator selected. All we need to do is create a dictionary that has the hyperparameters as keys and an iterable that holds the options we need to try out.
- Then all we have to do is create an object of GridSearchCV. Here basically we need to define a few named arguments:
    - ✓ **estimator**: estimator object created
    - ✓ **params_grid**: the dictionary object that holds the hyperparameters we want to try
    - ✓ **scoring**: evaluation metric that we want to use, we can simply pass a valid string/ object of evaluation metric
    - ✓ **cv**: number of cross-validation we have to try for each selected set of hyperparameters
    - ✓ **verbose**: we can set it to 1 to get the detailed print out while we fit the data to GridSearchCV

- ✓ **n_jobs**: number of processes we wish to run in parallel for this task if it is -1, it will use all available processors.

Conclusion:

For a large size dataset, Grid Search CV time complexity increases exponentially, and hence it's not practically feasible. One can shift to **Random Search CV** where the algorithm will randomly choose the combination of parameters.

The grid Search Cross-Validation technique is **computationally expensive**. The complexity of Grid Search CV increases with an increase in the number of parameters in the param grid. Thus Grid Search CV technique is not recommended for large-size datasets or param grids with a large number of components

15. List down some of the evaluation metric used to evaluate a regression model. Explain each of them in brief.

Ans: Some of the evaluation metric used to evaluate regression model are as follows:

1. **R2 Score**

   The R2 score (pronounced R-Squared Score) is a statistical measure that tells us how well our model is making all its predictions on a scale of zero to one.

   As mentioned above, it's not ideal for a model to predict the actual values in a regression problem (as opposed to a classification problem that has discrete levels of value).

   But we can use the R2 score to determine the accuracy of our model in terms of distance or residual. You can calculate the R2 score using the formula below:

Formula

$$R^2 = 1 - \frac{RSS}{TSS}$$

R^2 = coefficient of determination
RSS = sum of squares of residuals
TSS = total sum of squares

$$\text{RSS} = \Sigma\left(y_i - \widehat{y_i}\right)^2$$

Where: y_i is the actual value and, $\widehat{y_i}$ is the predicted value.

$$\text{TSS} = \Sigma\left(y_i - \overline{y}\right)^2$$

Where: y_i is the actual value and $\overline{y}$ is the mean value of the variable/feature

Outcome:

We can use the R2 score to get the accuracy of your model on a percentage scale, that is 0–100, just like in a classification model.

2. Mean Absolute Error (MAE)

The MAE is simply defined as the sum of all the distances/residual s(the difference between the actual and predicted value) divided by the total number of points in the dataset.

It is the absolute average distance of our model prediction.

We can calculate the MAE using the following formula:

$$MAE = \frac{1}{N} \sum_{i=1}^{N} |y_i - \hat{y}_i|$$

We can see that the above formula has two pipelines represented by the absolute symbol. The absolute symbol makes sure that the negative residual (which may be a result where the predicted value is greater than the actual value) is converted to positive so that it doesn't cancel out other positive residuals.

Outcome:

If we want to know the model's average absolute distance when making a prediction, we can use MAE. Simply if we want to know how close the predictions are to the actual model on average.

Also, another important note is that low MAE values indicate that the model is correctly predicting. Larger MAE values indicate that the model is poor at prediction.

3. Root Mean Squared Error (RMSE)

Another commonly used metric is the root mean squared error, which is the square root of the average squared distance (difference between actual and predicted value).

RMSE is defined as the square root of all the squares of the distance divided by the total number of points.

$$\text{RMSE} = \sqrt{\frac{1}{n} \sum_{i=1}^{n} (y_i - \hat{y}_i)^2}$$

RMSE functions similarly to MAE (that is, we use it to determine how close the prediction is to the actual value on average), but with a minor difference.

We use the RMSE to determine whether there are any large errors or distances that could be caused if the model overestimated the prediction (that is the model predicted values that were significantly higher than the actual value) or underestimated the predictions (that is, predicted values less than actual prediction).

Outcome:

RMSE is a popular evaluation metric for regression problems because it not only calculates how close the prediction is to the actual value on average, but it also indicates the effect of large errors. Large errors will have an impact on the RMSE result.
