MACHINE LEARNING WORKSHEET – 5

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

**Answer 1:** D) None of these

**Answer 2:** A) max\_depth

**Answer 3:** A) SMOTE

**Answer** **4:** C) 1 and 3

**Answer 5:** A) 3-1-2

**Answer 6:** B) Support Vector Machines

**Answer 7:** 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)

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

**Answer 8:** B) Lasso will lead to some of the coefficients to be very close to 0

D) Lasso will cause some of the coefficients to become 0

**Answer 9:** B) remove only one of the features

D) use Lasso regularisation

**Answer 10:** A) Overfitting

**Q11 to Q15 are subjective answer type questions, Answer them briefly.**

**Answer 11**: If there are only three or perhaps even four classes, one-hot encoding may not be that bad a choice, but chances are it’s worth exploring the alternatives, depending on the relative size of the dataset.

**Target encoding** is a very effective way to represent a categorical column and only takes up the space of one feature. Also known as mean encoding, each value in the column is replaced with the mean target value for that category.

**Answer 12: Imbalanced data** typically refers to a classification problem where the number of observations per class is not equally distributed.

**Approach to handling Imbalanced Data:**

**1.1 Data Level approach: Resampling Techniques**

**1.1.1 Random Under-Sampling-** Random Under sampling aims to balance class distribution by randomly eliminating majority class examples. This is done until the majority and minority class instances are balanced out.

**1.1.2 Random Over-Sampling-** Over-Sampling increases the number of instances in the minority class by randomly replicating them in order to present a higher representation of the minority class in the sample.

**1.1.3 Cluster-Based Over Sampling-** In this case, the K-means clustering algorithm is independently applied to minority and majority class instances. This is to identify clusters in the dataset. Subsequently, each cluster is oversampled such that all clusters of the same class have an equal number of instances and all classes have the same size.

**1.1.4 Informed Over Sampling: Synthetic Minority Over-sampling Technique for imbalanced data-** This technique is followed to avoid overfitting which occurs when exact replicas of minority instances are added to the main dataset. A subset of data is taken from the minority class as an example and then new synthetic similar instances are created. These synthetic instances are then added to the original dataset. The new dataset is used as a sample to train the classification models.

**1.1.5 Modified synthetic minority oversampling technique (MSMOTE) for imbalanced data-** It is a modified version of SMOTE. SMOTE does not consider the underlying distribution of the minority class and latent noises in the dataset. To improve the performance of SMOTE a modified method MSMOTE is used.

This algorithm classifies the samples of minority classes into 3 distinct groups – Security/Safe samples, Border samples, and latent nose samples. This is done by calculating the distances among samples of the minority class and samples of the training data.

Security samples are those data points which can improve the performance of a classifier. While on the other hand, noise are the data points which can reduce the performance of the classifier. The ones which are difficult to categorize into any of the two are classified as border samples.

While the basic flow of MSOMTE is the same as that of SMOTE (discussed in the previous section). In MSMOTE the strategy of selecting nearest neighbors is different from SMOTE. The algorithm randomly selects a data point from the k nearest neighbors for the security sample, selects the nearest neighbor from the border samples and does nothing for latent noise.

**1.2 Algorithmic Ensemble Techniques:**

**1.2.1. Bagging Based techniques for imbalanced data-** Bagging is an abbreviation of Bootstrap Aggregating. The conventional bagging algorithm involves generating ‘n’ different bootstrap training samples with replacement. And training the algorithm on each bootstrapped algorithm separately and then aggregating the predictions at the end. Bagging is used for reducing Overfitting in order to create strong learners for generating accurate predictions. Unlike boosting, bagging allows replacement in the bootstrapped sample.

**1.2.2. Boosting-Based techniques for imbalanced data-** Boosting is an ensemble technique to combine weak learners to create a strong learner that can make accurate predictions. Boosting starts out with a base classifier / weak classifier that is prepared on the training data.

**Answer 13:** The key difference between **ADASYN** and **SMOTE** is that the former uses a density distribution, as a criterion to automatically decide the number of synthetic samples that must be generated for each minority sample by adaptively changing the weights of the different minority samples to compensate for the skewed distributions.

**Answer 14:** **GridSearchCV** is a library function that is a member of sklearn’s model\_selection package. It helps to loop through predefined hyperparameters and fit your estimator (model) on your training set. So, in the end, you can select the best parameters from the listed hyperparameters.

If n\_jobs was set to a value higher than one, the data is copied for each point in the grid (and not n\_jobs times). This is done for efficiency reasons if individual jobs take very little time, but may raise errors if the dataset is large and not enough memory is available. A workaround in this case is to set pre\_dispatch. Then, the memory is copied only pre\_dispatch many times. A reasonable value for pre\_dispatch is 2 \* n\_jobs.

**Answer 15:** **Regression accuracy metrics:**

The **MSE, MAE, RMSE**, and **R-Squared** are mainly used metrics to evaluate the prediction error rates and model performance in regression analysis.

**MAE (Mean absolute error)** represents the difference between the original and predicted values extracted by averaged the absolute difference over the data set.

**MSE (Mean Squared Error)** represents the difference between the original and predicted values extracted by squared the average difference over the data set.

**RMSE (Root Mean Squared Error)** is the error rate by the square root of MSE.

**R-squared (Coefficient of determination)** represents the coefficient of how well the values fit compared to the original values. The value from 0 to 1 interpreted as percentages. The higher the value is, the better the model is.