MACHINE LEARNING WORKSHEET – 8

1. What is the advantage of hierarchical clustering over K-means clustering?
Ans:

- B) In hierarchical clustering you don't need to assign number of clusters in beginning

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

- A) max_depth

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

- C) RandomUnderSampler

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.
Ans:

- C) 1 and 3

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
Ans:

- D) 1-3-2

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?
Ans:

- C) K-Nearest Neighbors

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

- 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:

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

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

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

- B) remove only one of the features
- C) Use ridge regularization
- D) use Lasso regularization

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?
Ans:

- A) Overfitting
- D) Outliers

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 transforms categorical features to a format that works better with classification and regression algorithms.

- We know that we prefer to using **One-Hot Encoding** not **Label Encoding** when processing with non-ordinal data. And I real a blog which give the difference between **Label Encoding** and **One-Hot Encoding**. So I am wondering why **One-Hot Encoder** can avoid the situation that the model will misunderstand the data to be in some kind of order, 0<1<20<1<2 if the data has been **Label Encoding**.

- One hot encoding resolves the issue by explicitly showing that 1 category is true, while all others are false (1 vs 0). One column is turned into 3 columns which describe all categories as either true or false. As opposed to letting a model see one column with 0, 1, and 2. One column to show this data with label encoding does make it seem as if the data is numerical so 0 < 1 < 2.

- Label Ecoder be used for that.

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

Ans:

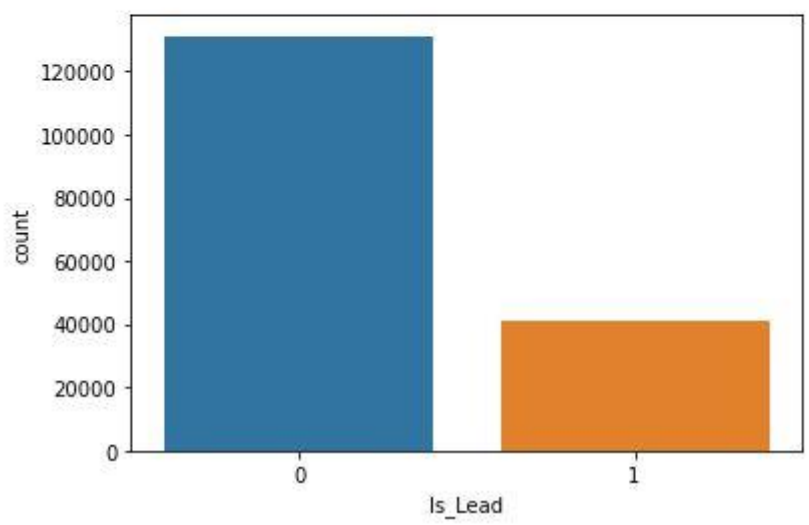
- Resampling (Oversampling and Undersampling)
- This technique is used to upsample or downsample the minority or majority class. When we are using an imbalanced dataset, we can oversample the minority class using replacement. This technique is called oversampling. Similarly, we can randomly delete rows from the majority class to match them with the minority class which is called undersampling.
- An example of this technique using the **sklearn** library's **resample()** is shown below for illustration purposes. Here, Is_Lead is our **target** variable. Let's see the distribution of the classes in the target.

```
df_train['Is_Lead'].value_counts()
```

```
0    131177
1     40830
Name: Is_Lead, dtype: int64
```

```
import seaborn as sns
sns.countplot(df_train['Is_Lead'])
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x27c0bdbf430>
```



- It has been observed that our target class has an imbalance. So, we'll try to upsample the data so that the minority class matches with the majority class.

```
from sklearn.utils import resample
#create two different dataframe of majority and minority class
df_majority = df_train[(df_train['Is_Lead']==0)]
df_minority = df_train[(df_train['Is_Lead']==1)]
# upsample minority class
df_minority_upsampled = resample(df_minority,
```

```
                                    replace=True,    # sample with replacement
                                    n_samples= 131177, # to match majority class
                                    random_state=42)  # reproducible results
# Combine majority class with upsampled minority class
df_upsampled = pd.concat([df_minority_upsampled, df_majority])
```

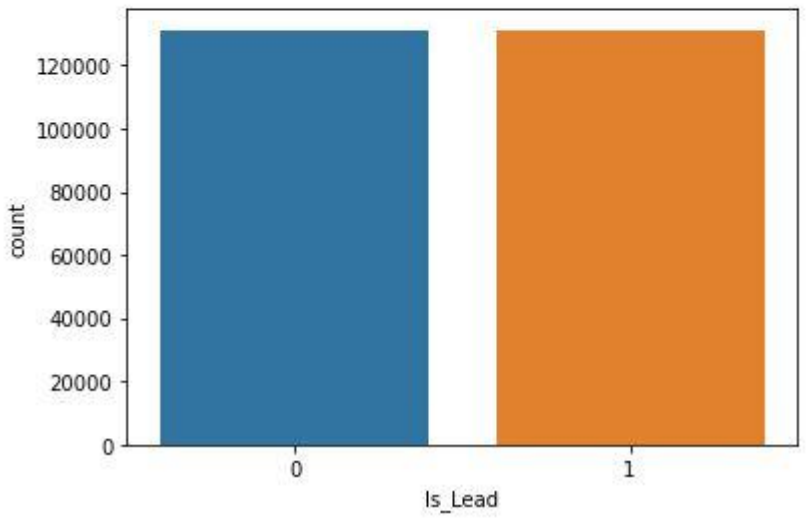
After upsampling, the distribution of class is balanced as below –

```
# Display new class counts
df_upsampled['Is_Lead'].value_counts()
```

```
1    131177
0    131177
Name: Is_Lead, dtype: int64
```

```
sns.countplot(df_upsampled['Is_Lead'])
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x27c0b83feb0>
```



- **Sklearn.utils resample** can be used for both undersamplings the majority class and oversample minority class instances.

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

Ans:

- 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. The latter generates the same number of synthetic samples for each original minority sample.

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 machine learning library for python. We have an exhaustive search over the specified parameter values for an estimator. An estimator object needs to provide basically a score function or any type of scoring must be passed. There are 2 main methods which can be implemented on GridSearchcv they are fit and predict. There are other also predict_proba,decision_function etc. But the two mentioned are frequently used. According to the type of algorithm which is been used for the dataset at hand for analysis it has its own different parameters. The user needs to give a different set of values for the important parameters. Gridsearchcv by cross-validations will find out the best value for the parameters mentioned. There are default values set for the parameters which can be also taken into consideration.

- GridSearchCv will calculate the average of out of fold recall for each combination of parameters, the set of parameters with best score, will be chosen by Grid search CV. It is fine to use the entire dataset, as you are using Cv method, which will check the score on out of fold set, hence you are not evaluating performance on Training data (on which model is trained) for parameter selection. for example: we want to tune max depth of tree, let's say maximum depth parameter we want to test are 5,10,15. Grid Search Cv will calculate recall score on out of fold set for all three value. The max depth value corresponding to best score on out of fold set will be chosen.

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

Ans:

There are several different metrics used to evaluate regression models. It also depends on which kind of regression model you are using i.e., if it is a linear or non-linear regression, is it logistic regression is it a simple linear/non-linear or a mixed effects regression etc. I would limit my answer to linear and non-linear regression without mixed effects. If you are looking at a particular regression model then some of the following metrics would be useful

1. Metrics like correlation coefficient and coefficient of determination. Root mean square error (RMSE) is a common metric for comparisons
2. goodness of fit plots which include Pred vs. Observed, Pred vs Residuals (which include plots like residuals and weighted residuals). In many instances you may also want to look at Pred vs Independent Variable as well as residuals vs. independent variable.
3. When comparing models you may look for all of the above plus use statistical metrics and tests for selection of a better fit. These depend on whether the two models that are being compared are nested or non-nested. For nested models, comparison can be made with a chi-squared test. We can compare two models (nested or non-nested) using goodness of fit criteria like the Akaike information criteria (AIC) or the Bayesian information criterion (BIC). In general for non-nested models, the lower the AIC value the better is the fit to the data.
4. Other diagnostic plots are also used to evaluate the underlying distribution assumptions (assumptions of normality) which include plots like Q-Q plots.

There are 3 main metrics for model evaluation in regression:

- Mean Square Error(MSE): MSE is calculated by the sum of square of prediction error which is real output minus predicted output and then divide by the number of data points. It gives an absolute number on how much our predicted results deviate from the actual number. We cannot interpret much insights from one single result but it gives a real number to compare against other model results and helps to select the best regression model.

- Root Mean Square Error(RMSE) is the square root of MSE. It is used more commonly than MSE because firstly sometimes MSE value can be too big to compare easily. Secondly, MSE is calculated by the square of error, and thus square root brings it back to the same level of prediction error and make it easier for interpretation.

- Mean Absolute Error (MAE): Mean Absolute Error (MAE) is similar to Mean Square Error (MSE). However, instead of the sum of square of error in MSE, MAE is taking the sum of absolute value of error. Compared to MSE or RMSE, MAE is a more direct representation of sum of error terms. MSE gives larger penalisation to big prediction error by squaring it while MAE treats all errors the same.