WORKSHEET SOLUTIONS SET 8

Machine Learning Solutions:

1. d) None of these
2. a) max_depth
3. c) RandomUnderSampler
4. c) 1 and 3
5. a) 3-1-2
6. c) K-Nearest Neighbors
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)
8. b) and d)
9. b), c) and d)
10. a) Overfitting
11. One hot encoding must be avoided when target variables have a large number of classes. In such cases, Label Encoder can be used.
12. The main objective of balancing classes is to either increasing the frequency of the minority class or decreasing the frequency of the majority class. This is done in order to obtain approximately the same number of instances for both the classes. Various methods to handle imbalance dataset are as follows :

    Random Under-Sampling:
    Random Undersampling aims to balance class distribution by randomly eliminating majority class examples. This is done until the majority and minority class instances are balanced out.
    Advantages
    It can help improve run time and storage problems by reducing the number of training data samples when the training data set is huge.
    Disadvantages
    It can discard potentially useful information which could be important for building rule classifiers.
    The sample chosen by random under sampling may be a biased sample. And it will not be an accurate representative of the population. Thereby, resulting in inaccurate results with the actual test data set.

    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.
    Advantages:
    Unlike under sampling this method leads to no information loss.
    Outperforms under sampling

Disadvantages:
It increases the likelihood of overfitting since it replicates the minority class events.

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.
Advantages:
This clustering technique helps overcome the challenge between class imbalance. Where the number of examples representing positive class differs from the number of examples representing a negative class.
Also, overcome challenges within class imbalance, where a class is composed of different sub clusters. And each sub cluster does not contain the same number of examples.
Disadvantages:
The main drawback of this algorithm, like most oversampling techniques is the possibility of over-fitting the training data.

Informed Over Sampling: Synthetic Minority Over-sampling Technique for imbalanced data (SMOTE):
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.
Advantages:
Mitigates the problem of overfitting caused by random oversampling as synthetic examples are generated rather than replication of instances
No loss of useful information
Disadvantages:
While generating synthetic examples SMOTE does not take into consideration neighboring examples from other classes. This can result in increase in overlapping of classes and can introduce additional noise
SMOTE is not very effective for high dimensional data

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.

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. The latter generates the same number of synthetic samples for each original minority sample.
14. GridSearchCVis the process of performing hyperparameter tuning in order to determine the optimal values for a given model. The performance of a model significantly depends on the value of hyperparameters. There is no way to know in advance the best values for hyperparameters so ideally, we need to try all possible values to know the optimal values. Doing this manually could take a considerable amount of time and resources and thus we use GridSearchCV to automate the tuning of hyperparameters.

With small data sets and lots of resources, Grid Search will produce accurate results. However, with large data sets, the high dimensions will greatly slow down computation time and be very costly. In this instance, it is advised to use Randomized Search since the number of iterations is explicitly defined by the data scientist.

15. Evaluation Metrics for Regression models are as follows:

M.A.E (Mean Absolute Error)

It is the simplest & very widely used evaluation technique. It is simply the mean of difference b/w actual & predicted values. Below, is the mathematical formula of the Mean Absolute Error.

$$MAE = \frac{1}{n}\sum_{j=1}^{n}|y_j - y_j|$$

M.S.E (Mean Squared Error)

Another evaluation technique is the Mean Squared Error. It takes the average of the square of the error. Here, the error is the difference b/w actual & predicted values. Below is the mathematical formula of the Mean Squared Error.

$$\mathrm{MSE} = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2$$

Well, A problem with the above function is that It changes the units. To avoid, that problem, we will use another technique, called, R.M.S.E (Root Mean Squared Error)

R.M.S.E (Root Mean Squared Error):

Root mean squared Error is another technique that is being used these days. First of all, it solves the problem in the above technique.

It squares the error & then it takes the square root of the total average function. Below is the mathematical function :

$$\mathrm{MSE} = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2.$$

But there is a limitation of this method. In Example 1, we can see that the error is very large. The actual value is 1 & the predicted value is 401. In example 2, we can see that, if we compare actual to predicted, the predicted is giving us a good result. For example 1 and 2, the error is 400, but actually, the ML model in example 2 is giving us better results. But, according to RMSE, the error is the same.

Therefore, to solve this problem, we use another similar, but yet modified method, which is discussed below.

R.M.S.L.E (Root Mean Squared Log Error) :

The mathematical function of this technique is displayed below.

$$\sqrt{\frac{1}{n}\sum_{i=1}^{n}(\log(x_i+1) - \log(y_i+1))^2}$$

Now, if we take the above case in the RMSLE, then, the RMSLE of Ex 1 is greater than Ex 2. & therefore, RMSLE solves the problem which occurred in RMSE (Root Mean Squared Error).This method actually, scales down the values & thus, it avoids the above error.

R — Squared (Relative Squared Error) :

This method helps us to calculate the relative error. This technique helps us to judge, which algorithm is better based on their mean squared errors. The mathematical formula of the R — Squared method is given below.

$$R^2 = 1 - \frac{\sum(y - \hat{y})^2}{\sum\left(y - \bar{y}\right)^2}$$

If x >1, this means that, the MSE of the numerator is greater than the MSE of the baseline model which in turn means that, the new model is worse than the baseline model. Higher is the R — Squared, better is the model. The limitation is that R-Squared value either increases or doesn't change upon adding more features. Regardless of how features impact the model. To overcome this limitation, there is another evaluation technique called Adjusted R — Squared.

Adjusted R — Squared

The mathematical formula is displayed below.

$$R^2_{adj} = 1 - \left[\frac{(1 - R^2)(n - 1)}{n - k - 1}\right]$$

Here, n: number of samples & k: number of features.

There is no inbuilt function on scikit-learn to calculate Adjusted R-Squared but we can find R-Squared & just calculate the Adjusted R-Squared.

Statistics Solutions:

1. b) The probability of failing to reject H0 when H1 is true
2. b) null hypothesis
3. b) Type II error
4. b) the t distribution with n - 1 degrees of freedom
5. c) rejecting Ho when it is false
6. d) two-tailed test

7. b) the probability of committing a Type I error
8. a) the probability of committing a Type II error
9. a) $z > z\alpha$
10. d) All of the above are needed.
11. b) critical value
12. d) All of the above.
13. Analysis of Variance, i.e. ANOVA in SPSS, is used for examining the differences in the mean values of the dependent variable associated with the effect of the controlled independent variables, after taking into account the influence of the uncontrolled independent variables. Essentially, ANOVA in SPSS is used as the test of means for two or more populations.ANOVA in SPSS must have a dependent variable which should be metric (measured using an interval or ratio scale). ANOVA in SPSS must also have one or more independent variables, which should be categorical in nature. In ANOVA in SPSS, categorical independent variables are called factors. A particular combination of factor levels, or categories, is called a treatment.
14. There are three primary assumptions in ANOVA:
    a) The responses for each factor level have a normal population distribution.
    b) These distributions have the same variance.
    c) The data are independent.
15. The only difference between one-way and two-way ANOVA is the number of independent variables. A one-way ANOVA has one independent variable, while a two-way ANOVA has two.

Python Solutions:

1. c) %
2. b) 0
3. c) 24
4. a) 2
5. d) 6
6. c) the finally block will be executed no matter if the try block raises an error or not.
7. a) It is used to raise an exception.
8. c) in defining a generator
9. a) & c)
10. a) & b)
11. Program to find factorial of a number:

```
num = int(input("Enter a number: "))
factorial = 1
if num < 0:
   print(" Factorial does not exist for negative numbers")
elif num == 0:
```

```
  print("The factorial of 0 is 1")
else:
  for i in range(1,num + 1):
    factorial = factorial*i
  print("The factorial of",num,"is",factorial)
```

12. Program to find whether a number is prime or composite :

```
num = int(input("Enter a number: "))
if num > 1:
  for i in range(2,num):
    if (num % i) == 0:
      print(num,"is not a prime number")
      break
    else:
      print(num,"is a composite number")
elif (num ==0 or num ==1):
  print(num, “is neither Prime nor Composite.”)
else:
  print(num,"is not a prime number")
```

13. Program to find if a string is a palindrome or not:

```
def isPalindrome(input):
  return input == input[::-1]

user_input = input(“Enter a string: ”)
ans = isPalindrome(user_input)

if ans:
  print("Yes")
else:
  print("No")
```

14. Program to get the third side of a triangle given the other two sides of a triangle :

```
def pythagoras(opposite_side,adjacent_side,hypotenuse):
    if opposite_side == str("x"):
      return ("Opposite = " + str(((hypotenuse**2)-(adjacent_side**2))**0.5))
    elif adjacent_side == str("x"):
      return ("Adjacent = " + str(((hypotenuse**2)-(opposite_side**2))**0.5))
    elif hypotenuse == str("x"):
      return ("Hypotenuse = " + str(((opposite_side**2) +
(adjacent_side**2))**0.5))
```

```
        else:
            return "You know the answer!"

print(pythagoras(3,4,'x'))
print(pythagoras(3,'x',5))
print(pythagoras('x',4,5))
print(pythagoras(3,4,5))
```

15. Program to print the frequency of each of the characters present in a given string:

```
user_input= input(“Enter a string”)

all_freq = {}

for i in user_input:
    if i in all_freq:
        all_freq[i] += 1
    else:
        all_freq[i] = 1

# printing result
print ("Count of all characters in”,user_input,” is :\n "
                                 +  str(all_freq))
```