1.a

2.b

3.c

4.a

5

6.b

7.b

8.ab

9.abd

10.d

11. **Outliers:**
The outliers may suggest experimental errors, variability in a measurement, or an anomaly. The age of a person may wrongly be recorded as 200 rather than 20 Years. Such an outlier should definitely be discarded from the dataset.
However, not all outliers are bad. Some outliers signify that data is significantly different from others. For example, it may indicate an anomaly like bank fraud or a rare disease.
Significance of outliers:

- Outliers badly affect mean and standard deviation of the dataset. These may statistically give erroneous results.
- Most machine learning algorithms do not work well in the presence of outlier. So it is desirable to detect and remove outliers.
- Outliers are highly useful in anomaly detection like fraud detection where the fraud transactions are very different from normal transactions.

What is Interquartile Range IQR?

IQR is used to **measure variability** by dividing a data set into quartiles. The data is sorted in ascending order and split into 4 equal parts. Q1, Q2, Q3 called first, second and third quartiles are the values which separate the 4 equal parts.

- Q1 represents the 25th percentile of the data.
- Q2 represents the 50th percentile of the data.
- Q3 represents the 75th percentile of the data.

If a dataset has *2n / 2n+1* data points, then
Q1 = median of the dataset.
Q2 = median of n smallest data points.
Q3 = median of n highest data points.
IQR is the range between the first and the third quartiles namely Q1 and Q3: *IQR = Q3 – Q1*. The data points which fall below *Q1 – 1.5 IQR* or above *Q3 + 1.5 IQR* are outliers.

Example:
Assume the data 6, 2, 1, 5, 4, 3, 50. If these values represent the number of chapatis eaten in lunch, then 50 is clearly an outlier.
Step by step way to detect outlier in this dataset using **Python**:
Step 1: Import necessary libraries.

```
Q1 = np.percentile(data, 25, interpolation = 'midpoint')
Q2 = np.percentile(data, 50, interpolation = 'midpoint')
Q3 = np.percentile(data, 75, interpolation = 'midpoint')

print('Q1 25 percentile of the given data is, ', Q1)
print('Q1 50 percentile of the given data is, ', Q2)
print('Q1 75 percentile of the given data is, ', Q3)

IQR = Q3 - Q1
print('Interquartile range is', IQR)
```

12. Boosting is an ensemble modeling technique that attempts to build a strong classifier from the number of weak classifiers. It is done by building a model by using weak models in series. Firstly, a model is built from the training data. Then the second model is built which tries to correct the errors present in the first model. This procedure is continued and models are added until either the complete training data set is predicted correctly or the maximum number of models is added.

Boosting Algorithms
There are several boosting algorithms. The original ones, proposed by **Robert Schapire** and **Yoav Freund** were not adaptive and could not take full advantage of the weak learners. Schapire and Freund then developed AdaBoost, an adaptive boosting algorithm that won the prestigious Gödel Prize. AdaBoost was the first really successful boosting algorithm developed for the purpose of binary classification. AdaBoost is short for Adaptive Boosting and is a very popular boosting technique that combines multiple “weak classifiers” into a single “strong classifier”

Boosting is an ensemble modeling technique that attempts to build a strong classifier from the number of weak classifiers. It is done by building a model by using weak models in series. Firstly, a model is built from the training data. Then the second model is built which tries to correct the errors present in the first model. This procedure is continued and models are added until either the complete training data set is predicted correctly or the maximum number of models is added.

Boosting Algorithms
There are several boosting algorithms. The original ones, proposed by **Robert Schapire** and **Yoav Freund** were not adaptive and could not take full advantage of the weak learners. Schapire and Freund then developed AdaBoost, an adaptive boosting algorithm that won the prestigious Gödel Prize. AdaBoost was the first really successful boosting algorithm developed for the purpose of binary classification. AdaBoost is short for Adaptive Boosting and is a very popular boosting technique that combines multiple “weak classifiers” into a single “strong classifier”

Algorithm:

1. *Initialise the dataset and assign equal weight to each of the data point.*
2. *Provide this as input to the model and identify the wrongly classified data points.*
3. *Increase the weight of the wrongly classified data points and decrease the weights of correctly classified data points. And then normalize the weights of all data points.*
4. *if (got required results)*
   Goto step 5
   else
   Goto step 2
5. *End*

Bagging

Bootstrap **A**ggregating, also known as bagging, is a machine learning ensemble meta-algorithm designed to improve the stability and accuracy of machine learning algorithms used in statistical classification and regression. It decreases the variance and helps to avoid overfitting. It is usually applied to decision tree methods. Bagging is a special case of the model averaging approach.

Suppose a set D of d tuples, at each iteration i, a training set D_i of d tuples is selected via row sampling with a replacement method (i.e., there can be repetitive elements from different d tuples) from D (i.e., bootstrap). Then a classifier model Mi is learned for each training set D < i. Each classifier M_i returns its class prediction. The bagged classifier M* counts the votes and assigns the class with the most votes to X (unknown sample).

Implementation Steps of Bagging

- **Step 1:** Multiple subsets are created from the original data set with equal tuples, selecting observations with replacement.
- **Step 2:** A base model is created on each of these subsets.
- **Step 3:** Each model is learned in parallel with each training set and independent of each other.
- **Step 4:** The final predictions are determined by combining the predictions from all the models.

Example of Bagging

The Random Forest model uses Bagging, where decision tree models with higher variance are present. It makes random feature selection to grow trees. Several random trees make a Random Forest.

To read more refer to this article: Bagging classifie

13. **Adjusted R-Squared:-**It measures the proportion of variation explained by only those independent variables that really help in explaining the dependent variable. It penalizes you for adding independent variable that do not help in predicting the dependent variable.

Adjusted R-Squared can be calculated mathematically in terms of sum of squares. The only difference between R-square and Adjusted R-square equation is degree of

freedom.

$$\bar{R}^2 = 1 - \frac{SS_{\text{res}}/\text{df}_e}{SS_{\text{tot}}/\text{df}_t}$$

Adjusted R-Squared Equation

In the above equation, dft is the degrees of freedom n– 1 of the estimate of the population variance of the dependent variable, and dfe is the degrees of freedom n – p – 1 of the estimate of the underlying population error variance.

Adjusted R-squared value can be calculated based on value of r-squared, number of independent variables (predictors), total sample size.

$$R^2_{\text{adjusted}} = 1 - \frac{(1 - R^2)(N - 1)}{N - p - 1}$$

where

R^2 = sample R-square

p = Number of predictors

N = Total sample size.

Adjusted R-Squared Equation 2

Ex:-n = 10
p = 3
*adj.r.squared = 1 - (1 - R.squared) * ((n - 1)/(n-p-1))*
print(adj.r.squared)

14. **Normalization?**

The process of arranging the data in a database is known as Normalization. It is a scaling technique used to reduce redundancy in which the values are shifted and scaled in a range of 0 and 1. Normalization is used to remove the unwanted characteristics from the dataset, and it is useful when there are no outliers as it can not handle them. Mathematically, Normalisation is denoted as:

$$x_{\text{stand}} = \frac{x - \text{mean}(x)}{\text{standard deviation } (x)}$$

	Normalization	Standardization
1.	Minimum and maximum value of features are used for scaling	Mean and standard deviation is used for scaling.
2.	It is used when features are of different scales.	It is used when we want to ensure zero mean and unit standard deviation.
3.	Scales values between [0, 1] or [-1, 1].	It is not bounded to a certain range.
4.	It is really affected by outliers.	It is much less affected by outliers.
5.	Scikit-Learn provides a transformer called MinMaxScaler for Normalization.	Scikit-Learn provides a transformer called StandardScaler for standardization.
6.	This transformation squishes the n-dimensional data into an n-dimensional unit hypercube.	It translates the data to the mean vector of original data to the origin and squishes or expands.
7.	It is useful when we don’t know about the distribution	It is useful when the feature distribution is Normal or Gaussian.
8.	It is a often called as Scaling Normalization	It is a often called as Z-Score Normalization.

Standardisation?

Data standardization is a process in which the data is restructured in a uniform format. In statistics, standardization compares the variables by putting all the variables on the same scale. It is done by transforming the features by subtracting from the mean and dividing by the standard deviation. This process is also known as the Z-score. Mathematically, Standardisation is denoted as:

$$x_{\text{stand}} = \frac{x - \text{mean}(x)}{\text{standard deviation }(x)}$$

Implementing Normalisation

```
#importing the Normalizer from sklearn
from sklearn.preprocessing import Normalizer
#Creating a sample data array
X = [[4, 1, 2, 2],[1, 3, 9, 3],[5, 7, 5, 1]]
transformer = Normalizer().fit(X)  # fit does nothing.
transformer
Normalizer()
transformer.transform(X)
```

Output:

```
array([[0.8, 0.2, 0.4, 0.4],
       [0.1, 0.3, 0.9, 0.3],
       [0.5, 0.7, 0.5, 0.1]])
```

Implementing Standardisation

```
#importing the StandardScaler from sklearn
from sklearn.preprocessing import StandardScaler
#Creating a sample data array
data = [[0, 0], [0, 0], [1, 1], [1, 1]]
scaler = StandardScaler()
print(scaler.fit(data))
print(scaler.mean_)
print(scaler.transform(data))
print(scaler.transform([[2, 2]]))
```

15.

Advantages of Cross Validation

1. Reduces Overfitting: In Cross Validation, we split the dataset into multiple folds and train the algorithm on different folds. This prevents our model from overfitting the training dataset. So, in this way, the model attains the generalization capabilities which is a good sign of a robust algorithm.

Note: Chances of overfitting are less if the dataset is large. So, Cross Validation may

not be required at all in the situation where we have sufficient data available.

2. Hyperparameter Tuning: Cross Validation helps in finding the optimal value of hyperparameters to increase the efficiency of the algorithm

15. Cross-validation is **a statistical method used to estimate the performance (or accuracy) of machine learning models**. It is used to protect against overfitting in a predictive model, particularly in a case where the amount of data may be limited

Advantages of Cross Validation

1. Reduces Overfitting: In Cross Validation, we split the dataset into multiple folds and train the algorithm on different folds. This prevents our model from overfitting the training dataset. So, in this way, the model attains the generalization capabilities which is a good sign of a robust algorithm.

Note: Chances of overfitting are less if the dataset is large. So, Cross Validation may not be required at all in the situation where we have sufficient data available.

2. Hyperparameter Tuning: Cross Validation helps in finding the optimal value of hyperparameters to increase the efficiency of the algorithm.

Disadvantages of Cross Validation

1. Increases Training Time: Cross Validation drastically increases the training time. Earlier you had to train your model only on one training set, but with Cross Validation you have to train your model on multiple training sets.

For example, if you go with 5 Fold Cross Validation, you need to do 5 rounds of training each on different 4/5 of available data. And this is for only one choice of hyperparameters. If you have multiple choice of parameters, then the training period will shoot too hig