VIVA QUESTION – DSS

PRACTICAL -01

Question: What is the primary advantage of using Python for data science?

Answer: Python is versatile and has extensive libraries, making it suitable for various data science tasks.

Question: Name three popular Python libraries used in data science and describe their primary functions briefly.

Answer: Three popular libraries are NumPy (for numerical computations), Pandas (for data manipulation), and Matplotlib (for data visualization).

Question: How can you install Python libraries using pip?

Answer: You can install libraries using 'pip install library name' in the command line, replacing 'library name' with the desired library.

Question: Explain the key differences between Python 2 and Python 3 in the context of data science.

Answer: Python 3 is recommended for data science due to better support, but Python 2 is still used in some legacy environments.

Question: What is the purpose of a Jupyter Notebook in data science, and how do you create one?

Answer: A Jupyter Notebook is used for interactive data analysis, and you can create one by running 'jupyter notebook' in your terminal.

Question: How would you import a dataset in Python for analysis, and which library is commonly used for this task?

Answer: You can use the Pandas library to import datasets, typically by using 'pd.read csv()' or 'pd.read_excel()' functions.

Question: What is the role of data cleaning in the data science process, and can you provide an example of a data cleaning task?

Answer: Data cleaning ensures data quality by handling missing values, duplicates, and outliers. An example is removing duplicate records from a dataset.

Question: Explain the concept of data exploration in data science and give an example of a data visualization technique used for this purpose.

Answer: Data exploration involves understanding the dataset's characteristics. A common technique is creating histograms to visualize the distribution of a numerical variable.

Question: What is the purpose of statistical measures such as mean, median, and standard deviation in data analysis?

Answer: These measures help summarize and understand the central tendency and variability of data, providing insights into data distributions.

Question: How can you write a basic Python function to perform statistical analysis, like calculating the mean of a dataset?

Answer: You can write a function using Python's basic math operations and looping constructs to calculate the mean of a dataset.

PRACTICAL -02

Question: How do you compute the mean of a dataset manually?

Answer: Add all values and divide by the number of values.

Question: What is the formula for calculating the mean statistically?

Answer: Sum of all values divided by the number of values.

Question: How can an end user compute the mean of a dataset without performing calculations themselves?

Answer: They can use software tools or spreadsheet applications that provide a mean function.

Question: How do you calculate the mode manually?

Answer: Identify the value that occurs most frequently in the dataset.

Question: What is the mode statistically?

Answer: It is the value with the highest frequency in a dataset.

Question: How can an end user find the mode without manual calculation?

Answer: They can use software tools or functions that automatically identify the mode.

Question: What is the manual method to find the median?

Answer: Arrange the data in ascending order and find the middle value (or the average of two middle values).

Question: Statistically, how is the median computed?

Answer: The median is the middle value when data is sorted or the average of two middle values in a dataset.

Question: How can an end user determine the median without manual calculation?

Answer: They can use software or tools that automatically calculate the median from a dataset.

PRACTICAL -03

Question: What is the range of a dataset, and how is it calculated?

Answer: The range measures the spread between the maximum and minimum values and is calculated as Range = Max - Min.

Question: How is the variance of a dataset computed in Python?

Answer: Variance is calculated as the average of the squared differences between each data point and the mean.

Question: Explain the purpose of the standard deviation in statistics.

Answer: Standard deviation measures the degree of variation or dispersion in a dataset.

Question: How do you calculate the standard deviation in Python?

Answer: The standard deviation is the square root of the variance.

Question: What does the interquartile range (IQR) represent in a dataset?

Answer: The IQR measures the spread of the middle 50% of the data, making it robust against outliers.

Question: How can you compute the IQR in Python?

Answer: Calculate the difference between the third quartile (Q3) and the first quartile (Q1) in a dataset.

Question: Name a Python library or module commonly used for statistical calculations.

Answer: NumPy is a popular library for statistical calculations in Python.

Question: What is the main difference between variance and standard deviation?

Answer: Variance provides the average of squared differences, while standard deviation is its square root and provides a measure in the original units of data.

Question: In what scenarios is the interquartile range (IQR) preferred over the range?

Answer: IQR is preferred when data contains outliers, as it's less sensitive to extreme values compared to the range.

Question: How can Python be used to visualize the spread of data, along with statistical calculations like variance and standard deviation?

Answer: Python libraries like Matplotlib can be used to create visual representations, such as histograms or box plots, to visualize data spread and statistics.

PRACTICAL -04

Question: What is Linear Regression in the context of data analysis?

Answer: Linear Regression is a statistical method for modeling the relationship between a dependent variable and one or more independent variables.

Question: What is the objective of linear regression?

Answer: The objective is to find the best-fitting line that represents the relationship between variables.

Question: How is the equation of a linear regression line represented?

Answer: It's represented as $Y = aX + b$, where Y is the dependent variable, X is the independent variable, a is the slope, and b is the intercept.

Question: What is the role of the coefficient 'a' in a linear regression equation?

Answer: 'a' represents the slope of the regression line, indicating the change in the dependent variable for a unit change in the independent variable.

Question: Explain the purpose of the coefficient 'b' in linear regression.

Answer: 'b' is the intercept, representing the predicted value of the dependent variable when the independent variable is zero.

Question: How is the best-fitting line determined in linear regression?

Answer: It's determined by minimizing the sum of squared differences between the observed and predicted values (least squares method).

Question: What is the difference between simple linear regression and multiple linear regression?

Answer: Simple linear regression involves one independent variable, while multiple linear regression includes multiple independent variables.

Question: How can you implement linear regression in Python?

Answer: You can use libraries like scikit-learn or statsmodels to implement linear regression in Python.

Question: What is the coefficient of determination (R-squared) in linear regression?
Answer: R-squared measures the proportion of the variance in the dependent variable explained by the independent variable(s).

Question: How do you assess the goodness of fit in linear regression, and what are some common metrics used?

Answer: Goodness of fit is assessed using metrics like R-squared, mean squared error (MSE), and mean absolute error (MAE).

PRACTICAL -05

Question: What is multilinear regression, and how does it differ from simple linear regression?

Answer: Multilinear regression is an extension of simple linear regression, where there are multiple independent variables, allowing for more complex modeling.

Question: What is the goal of multilinear regression?

Answer: The goal is to model the relationship between the dependent variable and multiple independent variables in a linear fashion.

Question: How is the equation for multilinear regression represented?

Answer: It's represented as Y = aX1 + bX2 + cX3 + ... + nXn + e, where Y is the dependent variable, X1, X2, etc., are independent variables, and a, b, c, etc., are coefficients.

Question: What is the purpose of the coefficients in multilinear regression?

Answer: The coefficients represent the impact of each independent variable on the dependent variable, considering the other variables' effects.

Question: How is the best-fitting model determined in multilinear regression?

Answer: The best-fitting model is found by minimizing the sum of squared differences between observed and predicted values.

Question: In multilinear regression, what is the significance of the p-value associated with each coefficient?

Answer: The p-value indicates the significance of each independent variable in predicting the dependent variable. Lower p-values suggest greater significance.

Question: How can you implement multilinear regression in Python?

Answer: You can use libraries like scikit-learn, statsmodels, or other regression libraries available in Python.

Question: What is multicollinearity in multilinear regression, and why is it a concern?
Answer: Multicollinearity occurs when independent variables are highly correlated, making it difficult to distinguish their individual impacts.

Question: How do you assess the goodness of fit in multilinear regression, and what metrics are commonly used?

Answer: Goodness of fit is assessed using metrics such as R-squared, adjusted R-squared, and mean squared error (MSE).

Question: What are some common assumptions made in multilinear regression, and why are they important?

Answer: Common assumptions include linearity, independence of errors, and normality of residuals. They are important for the validity of the model and its predictions.

PRACTICAL -06

Question: What is Logistic regression?

Answer: A classification algorithm for binary outcomes.

Question: How is the Logistic regression model represented?

Answer: Log-odds or probabilities are modeled using the sigmoid function.

Question: What's the primary use of Logistic regression?

Answer: Predicting binary outcomes like yes/no or 0/1.

Question: How do you implement Logistic regression in Python?

Answer: Use libraries like scikit-learn or statsmodels.

Question: What's the purpose of the logistic function in Logistic regression?

Answer: It transforms the linear combination of features into probabilities.

Question: What's the difference between Linear and Logistic regression?

Answer: Linear regression predicts continuous outcomes; Logistic regression predicts probabilities of discrete outcomes.

Question: What are the key assumptions in Logistic regression?

Answer: Linearity, independence of errors, and lack of multicollinearity.

Question: How is model performance in Logistic regression typically evaluated?

Answer: Using metrics like accuracy, precision, recall, and the area under the ROC curve.

Question: What is the "odds ratio" in Logistic regression?

Answer: It measures the odds of an event happening compared to the odds of it not happening.

Question: How is regularization applied in Logistic regression?

Answer: Regularization techniques like L1 or L2 regularization are used to prevent overfitting.

PRACTICAL-07

Question: What is a Decision Tree?

Answer: A tree-like model for decision-making.

Question: How does a Decision Tree make decisions?

Answer: It splits data into branches based on features.

Question: What's the root node in a Decision Tree?

Answer: The top node where the first decision is made.

Question: What's a leaf node in a Decision Tree?

Answer: End nodes representing final predictions.

Question: What's the primary goal of a Decision Tree?

Answer: To partition data into homogeneous subsets.

Question: How do Decision Trees handle both classification and regression tasks?

Answer: They use different criteria at each node (e.g., Gini impurity for classification, mean squared error for regression).

Question: How can Decision Trees be prone to overfitting, and how is it mitigated?

Answer: Overfitting can occur by creating very complex trees; it's mitigated by pruning or limiting tree depth.

Question: What is "entropy" in the context of Decision Trees?

Answer: A measure of impurity or disorder used to make splitting decisions.

Question: How do you implement Decision Trees in Python?

Answer: Use libraries like scikit-learn for easy implementation.

Question: What's a decision tree's strength and limitation?

Answer: Strength: Easy to interpret. Limitation: Prone to overfitting and might not capture complex relationships.

PRACTICAL-08

Question: What is the Naive Bayes Classifier?

Answer: A probabilistic classification algorithm.

Question: How does Naive Bayes work?

Answer: It uses Bayes' theorem to calculate conditional probabilities.

Question: What's the "naive" assumption in Naive Bayes?

Answer: It assumes independence between features.

Question: What's the primary application of Naive Bayes?

Answer: Text classification, spam filtering, and sentiment analysis.

Question: What's the key equation in Naive Bayes?

Answer: Bayes' theorem: P(A|B) = P(B|A) * P(A) / P(B).

Question: How does Naive Bayes handle text data?

Answer: It models the likelihood of each word or feature occurring in a class.

Question: What types of Naive Bayes classifiers exist?

Answer: Common types include Gaussian, Multinomial, and Bernoulli Naive Bayes.

Question: How is Laplace smoothing used in Naive Bayes?

Answer: It prevents zero probability estimates by adding a small constant to each count.

Question: What's the primary evaluation metric for Naive Bayes?

Answer: Accuracy, precision, recall, and F1-score are common evaluation metrics.

Question: How can you implement Naive Bayes in Python?

Answer: Use libraries like scikit-learn for easy implementation.