Module 6 - Statistics

Statistics in Data Science

1. Write a Python script to analyze a dataset and demonstrate:
   a. The impact of outliers on the mean, median, and mode.
   b. How these measures can be used to summarize data effectively.
   Steps
      a. Create a dataset with outliers.
      b. Calculate and compare the central tendency measures before and after outlier removal.

Measures of Central Tendency

2. Use a dataset containing test scores to compute the mean, median, and mode using Python. Also, visualize these statistics on a histogram.
   Steps
      a. Use Python's `statistics` module for calculations.
      b. Use `matplotlib` or `seaborn` for the histogram.

Measures of Dispersion

3. Write a Python program to calculate:
   a. The range, interquartile range (IQR), variance, and standard deviation of a dataset.
   b. Visualize the spread using a boxplot.
   Steps
      a. Use `numpy` for dispersion measures.
      b. Use `matplotlib` for the boxplot.

Hypothesis Testing

4. Write a Python program to perform a paired t-test to evaluate the effectiveness of a study program. Use two datasets: scores before and after the program.
   Steps
      a. Use `scipy.stats.ttest_rel()` for the t-test.
      b. Interpret the p-value to determine significance.

Types of Errors

5. Write a Python function to simulate a hypothesis test with the following:
   a. A scenario that results in a Type I error (rejecting a true null hypothesis).
   b. A scenario that results in a Type II error (failing to reject a false null hypothesis).

Steps

a. Use synthetic data and statistical tests.

b. Highlight conditions leading to each error type.

Regression Analysis

6. Use a dataset of years of experience vs. salary to:

a. Fit a simple linear regression model.

b. Predict the salary for 5 years of experience.

Steps:

a. Use sklearn.linear_model.LinearRegression.

b. Visualize the regression line over the scatterplot.

Underfitting and Overfitting

7. Demonstrate underfitting and overfitting using polynomial regression:

a. Generate a dataset from a quadratic equation with noise.

b. Fit a linear model (underfitting) and a high-degree polynomial (overfitting).

c. Visualize both models and their performance on training and testing sets.

Steps

a. Use numpy to generate synthetic data.

b. Use sklearn for regression models.

Types of Regression Analysis

8. Compare the performance of:

a. Linear regression

b. Polynomial regression

c. Lasso regression

Use a real or synthetic dataset to fit these models and compare their mean squared errors.

Steps

a. Use sklearn for implementation.

b. Use matplotlib to visualize results.

Correlation and Regression

9. Create a scatterplot to visualize the relationship between two variables (e.g., hours studied vs. marks scored).

a. Calculate the Pearson correlation coefficient.

b. Fit a linear regression model and overlay the regression line on the scatterplot.

Steps

a. Use numpy for correlation.

b. Use sklearn for regression and matplotlib for visualization.

Note : After completing the code, submit it in .ipynb format along with a brief explanation for each part in your own words.
