

## Task 1 Answer: Description of the Linear Regression Model

### Overview of Linear Regression

Linear regression is a statistical method that is used to model the relationship between a dependent variable and one or more independent variables by fitting a linear equation to observed data. The simplest form of the linear regression equation with one dependent and one independent variable is defined as:

$$y = \beta_0 + \beta_1 x + \epsilon$$

where:

- $y$  is the dependent variable,
- $x$  is the independent variable,
- $\beta_0$  is the intercept,
- $\beta_1$  is the slope of the line,
- $\epsilon$  is the error term, which accounts for variability in  $y$  that cannot be explained by  $x$ .

The coefficients  $\beta_0$  and  $\beta_1$  are derived based on minimizing the sum of the squared difference between the observed values and the values predicted by the model.

### Practical Example with Data

Let's consider a simple dataset where we want to predict a student's test score based on the number of hours they studied. Here is some sample data:

Hours Studied	Test Score
1	50
2	55
3	60
4	65
5	70

## **Python Code for Linear Regression**

```
import numpy as np

from sklearn.linear_model import LinearRegression

import matplotlib.pyplot as plt

# some data

X = np.array([[1], [2], [3], [4], [5]]) # Hours Studied
y = np.array([50, 55, 60, 65, 70]) # Test Score

# Create a linear regression model

model = LinearRegression()

# Fit the model

model.fit(X, y)

# Coefficients

print("Intercept:", model.intercept_)
print("Slope:", model.coef_[0])

# Making predictions

X_pred = np.array([[6]])
y_pred = model.predict(X_pred)
print(f"Predicted test score for 6 hours of study: {y_pred[0]}")

# Plotting the results

plt.scatter(X, y, color='blue')
plt.plot(X, model.predict(X), color='red')
plt.title('Hours vs Score')
```

**`plt.xlabel('Hours Studied')`**

**`plt.ylabel('Test Score')`**

**`plt.show()`**

#### Explanation

- **Intercept and Slope:** After fitting the model to the data, `model.intercept_` gives the y-intercept ( $\beta_0$ ), and `model.coef_` provides the slope ( $\beta_1$ ) of the line. These values represent the estimated coefficients for the linear regression problem.
- **Predicting New Data:** The `model.predict()` function is used to predict the response for new observations. Here, it predicts the test score for a student who studies for 6 hours.
- **Plotting:** The code plots the original data points and the linear regression line. This visualization helps to understand how well the linear model fits the data.

#### Output Details

- *Intercept:* This value indicates the expected test score for someone who has studied zero hours.
- *Slope:* This represents the increase in the test score for each additional hour of study.
- *Predicted Score:* The expected test score for studying 6 hours, based on the model.

*This example covers the basic implementation and interpretation of a linear regression model in Python using scikit-learn. For the task requirements, you can expand on this by explaining the code and model assumptions in more detail, and then compiling it into a PDF for submission. Remember to include any plots and outputs in the PDF as well.*