ASSIGNMENT: FEATURE ENGINEERING

1) What is a parameter?
   - A parameter in machine learning is a variable that the model learns from the training data. These are internal coefficients or weights that the model adjusts to make predictions. Examples include weights in linear regression and connection strengths in neural networks.

2) What is correlation?
   - Correlation is a statistical measure that describes the relationship between two variables. It indicates how changes in one variable are associated with changes in another.
   - *What does negative correlation mean?*
     Negative correlation means that as one variable increases, the other decreases. For example, in a dataset of students, if study time increases while entertainment time decreases, these two variables have a negative correlation.

3) Define Machine Learning. What are the main components in Machine Learning?
   - Machine Learning (ML) is a branch of artificial intelligence that enables computers to learn patterns from data and make predictions without explicit programming.
   - The main components of ML are:
     a) **Data**: The foundation for training models.
     b) **Features**: Input variables used to make predictions.
     c) **Model:** A mathematical representation that maps inputs to outputs.
     d) **Loss function**: Measures how well the model performs.
     e) **Optimizer**: Adjusts parameters to minimize errors.
     f) **Evaluation Metrics**: Metrics like accuracy and precision used to assess model performance.

4) How does loss value help in determining whether the model is good or not?
   - The loss value quantifies how far the model's predictions are from the actual values. A lower loss indicates a better model. Common loss functions include Mean Squared Error (MSE) for regression and Cross-Entropy Loss for classification.

5) What are continuous and categorical variables?
   - **Continuous variables**: Numeric values that can take an infinite range of values (e.g., height, temperature).
   - **Categorical variables**: Discrete values that represent different groups or categories (e.g., gender, colors).

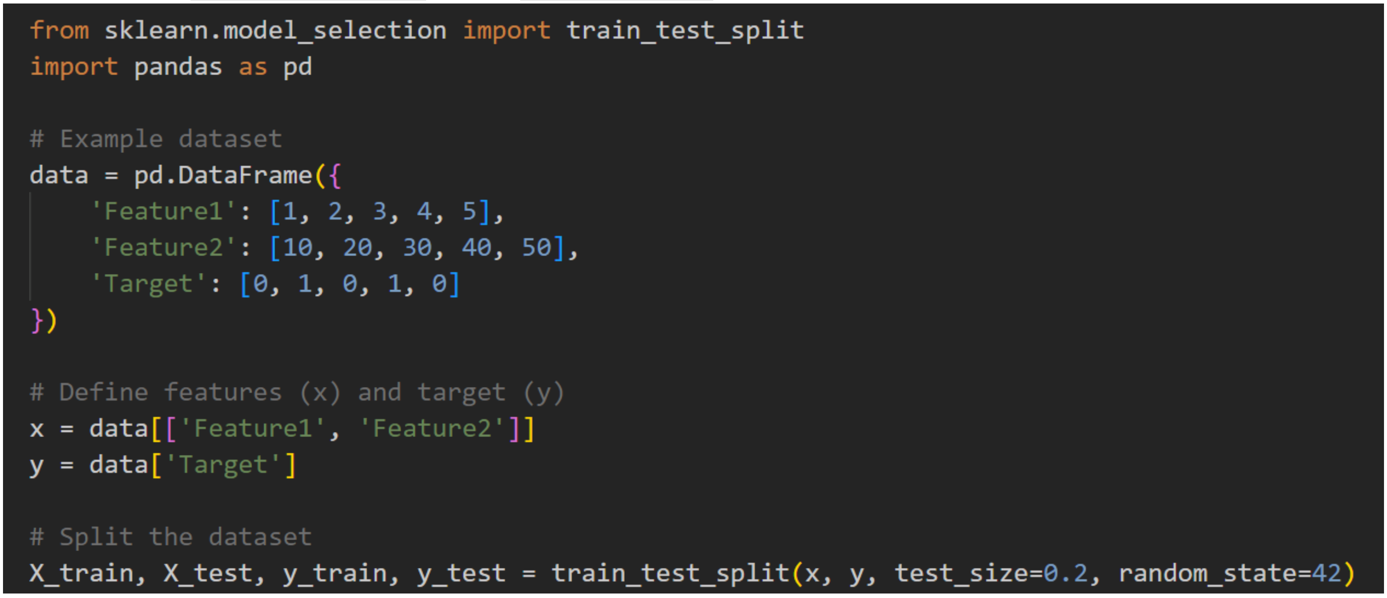
6) How do we handle categorical variables in Machine Learning? What are the common techniques?
   - Handling categorical variables involves transforming them into a numerical format. Common techniques include:
     a) **Label Encoding**: Assigning unique numbers to each category.
     b) **One-Hot Encoding**: Creating binary columns for each category.
     c) **Ordinal Encoding**: Assigning numbers based on order.
     d) **Target Encoding**: Replacing categories with mean target values.

7) What do you mean by training and testing a dataset?
   - **Training dataset**: The data used to train the model and adjust its parameters.
   - **Testing dataset**: The unseen data used to evaluate model performance and generalization.

8) What is sklearn.preprocessing?
   - sklearn.preprocessing is a module in the Scikit-Learn library that provides tools for feature scaling, normalization, encoding categorical variables, and transforming data.

9) What is a Test set?
   - A test set is a subset of the dataset that is used to evaluate the model's performance after training. It helps measure how well the model generalizes to unseen data.

10) How do we split data for model fitting (training and testing) in Python?
   - We can use train_test_split from Scikit-Learn:



```python
from sklearn.model_selection import train_test_split
import pandas as pd

# Example dataset
data = pd.DataFrame({
    'Feature1': [1, 2, 3, 4, 5],
    'Feature2': [10, 20, 30, 40, 50],
    'Target': [0, 1, 0, 1, 0]
})

# Define features (x) and target (y)
x = data[['Feature1', 'Feature2']]
y = data['Target']

# Split the dataset
X_train, X_test, y_train, y_test = train_test_split(x, y, test_size=0.2, random_state=42)
```

     Here, test_size=0.2 means 20% of the data is used for testing.
   - How do you approach a Machine Learning problem?
     The general approach includes:
     a) **Understanding the Problem** – Define the objective.
     b) **Collecting Data** – Gather relevant data.

c) **Preprocessing Data** – Clean and prepare data (handle missing values, encoding).
d) **Exploratory Data Analysis (EDA)** – Visualize and analyze trends.
e) **Feature Engineering** – Select and transform features.
f) **Choosing a Model** – Select an appropriate algorithm.
g) **Training the Model** – Train using the dataset.
h) **Evaluating the Model** – Use metrics like accuracy or RMSE.
i) **Hyperparameter Tuning** – Optimize model performance.
j) **Deployment** – Deploy the model for real-world use.

11) Why do we have to perform EDA before fitting a model to the data?

- Exploratory Data Analysis (EDA) helps in understanding the structure, distribution, and relationships in the data. It identifies missing values, outliers, and inconsistencies, ensuring data quality before model training. EDA also helps in selecting relevant features and determining preprocessing techniques.

12) What is correlation?

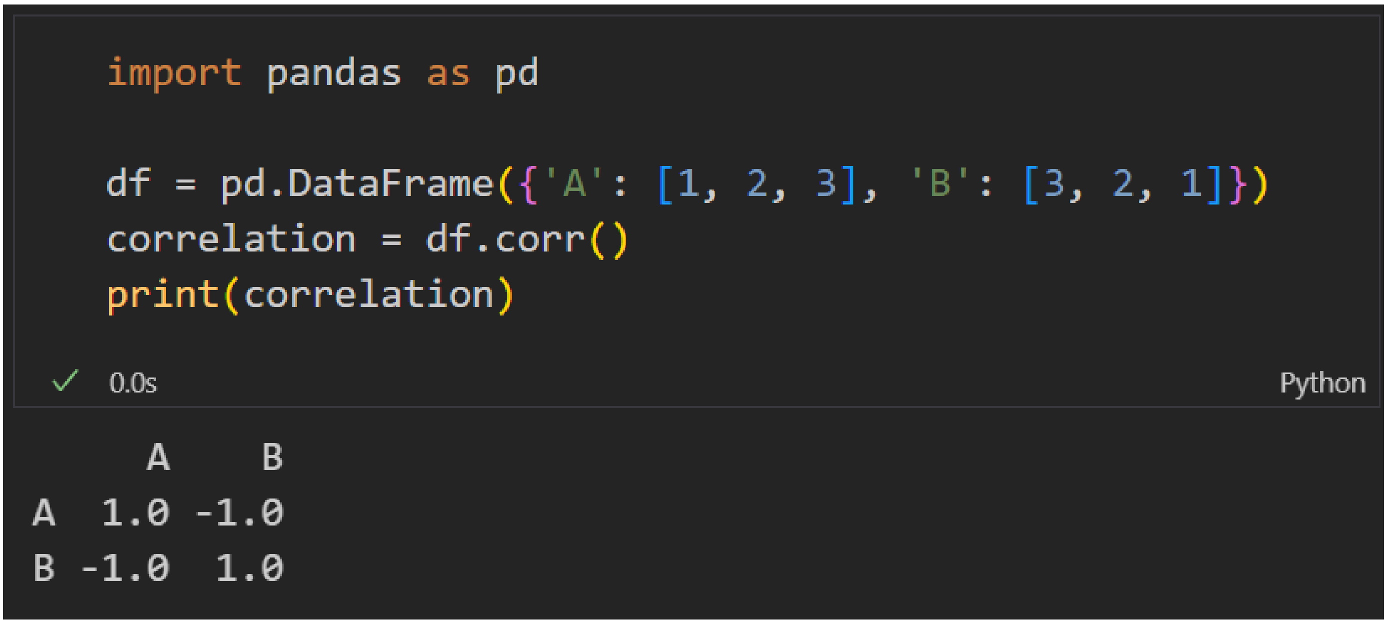
- Correlation measures the statistical relationship between two variables. It indicates how changes in one variable are associated with changes in another. Correlation values range from -1 to 1:
  a) **1**: Perfect positive correlation
  b) **0**: No correlation
  c) **-1**: Perfect negative correlation

13) What does negative correlation mean?

- A negative correlation means that as one variable increases, the other decreases. For example, in a dataset of car speeds and fuel efficiency, a higher speed may correlate negatively with fuel efficiency (higher speed → lower efficiency).

14) How can you find correlation between variables in Python?

- In Python, you can use the corr() function from pandas or pearsonr from scipy.



```python
import pandas as pd

df = pd.DataFrame({'A': [1, 2, 3], 'B': [3, 2, 1]})
correlation = df.corr()
print(correlation)
```

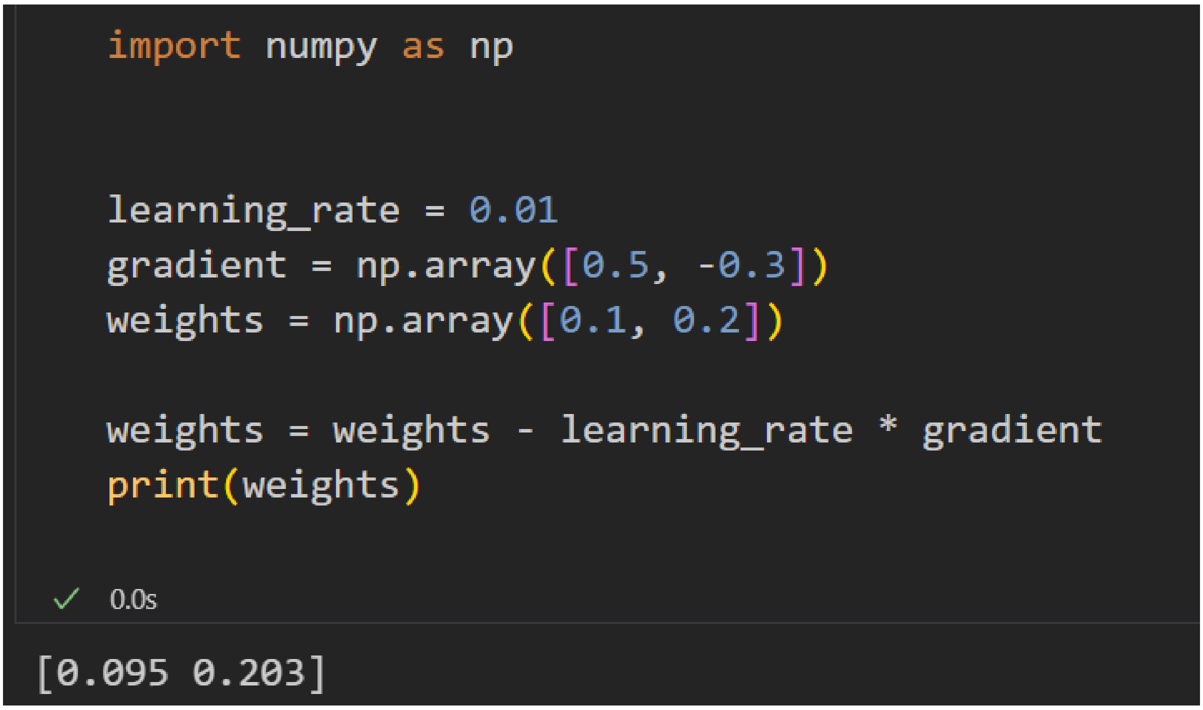
```
     A    B
A  1.0 -1.0
B -1.0  1.0
```

15) What is causation? Explain difference between correlation and causation with an example.

- Causation means that one event causes another.
- Correlation does not imply causation.
- Example:
  a) **Correlation:** Ice cream sales and drowning incidents are correlated.
  b) **Causation:** Summer increases both ice cream sales and swimming activities, but ice cream does not cause drowning.

16) What is an Optimizer? What are different types of optimizers? Explain each with an example.

- An optimizer is an algorithm that minimizes the loss function in machine learning models. Common optimizers:
  a) **Gradient Descent**: Updates weights based on gradients.



```python
import numpy as np


learning_rate = 0.01
gradient = np.array([0.5, -0.3])
weights = np.array([0.1, 0.2])

weights = weights - learning_rate * gradient
print(weights)
```

```
[0.095 0.203]
```

b) **Adam**: Combines momentum and adaptive learning.

c) **RMSprop**: Adjusts learning rate dynamically.

17) What is sklearn.linear_model?

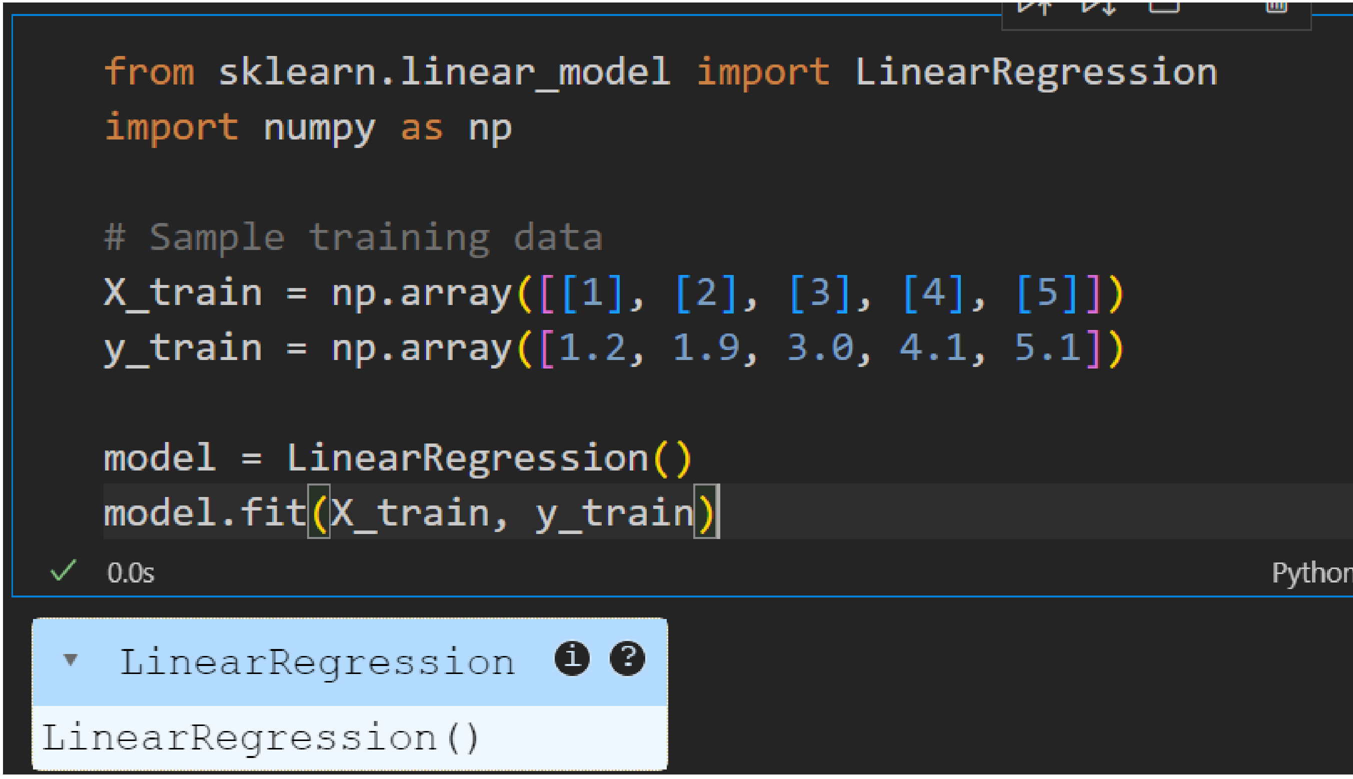
– sklearn.linear_model is a module in Scikit-learn that provides linear regression algorithms such as:

a) LinearRegression
b) LogisticRegression
c) Ridge
d) Lasso

18) What does model.fit() do? What arguments must be given?

– model.fit() trains the machine learning model on given data. It requires:

a) **X** (features)
b) **y** (target variable)

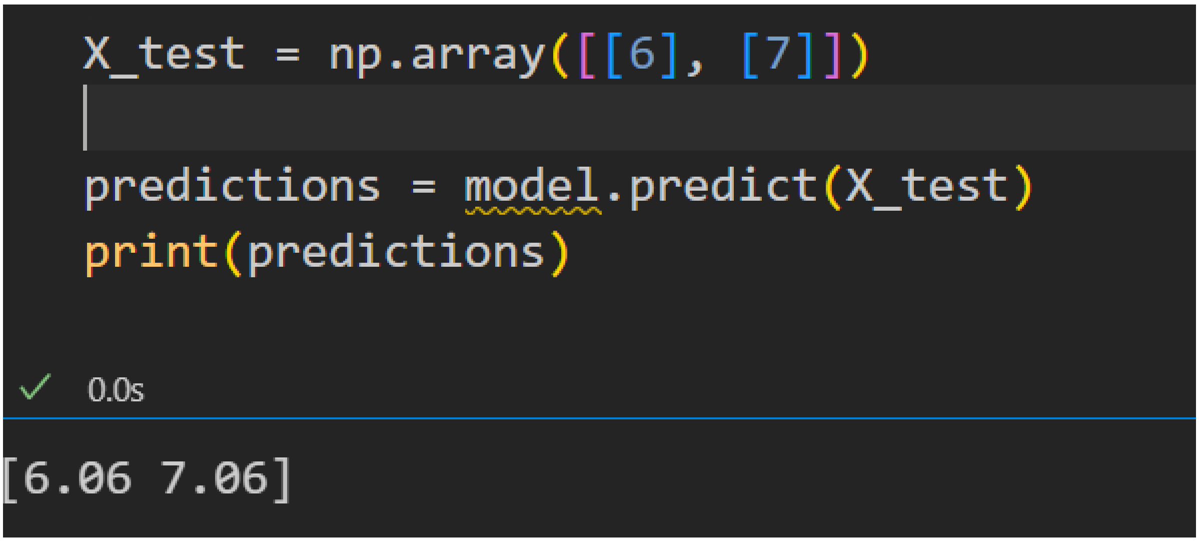


```
from sklearn.linear_model import LinearRegression
import numpy as np

# Sample training data
X_train = np.array([[1], [2], [3], [4], [5]])
y_train = np.array([1.2, 1.9, 3.0, 4.1, 5.1])

model = LinearRegression()
model.fit(X_train, y_train)
```

```
▾ LinearRegression ℹ ?
LinearRegression()
```

19) What does model.predict() do? What arguments must be given?

– model.predict() makes predictions using trained models. It requires feature data X.



```
X_test = np.array([[6], [7]])

predictions = model.predict(X_test)
print(predictions)
```

```
✓ 0.0s
```

```
[6.06 7.06]
```

20) What are continuous and categorical variables?

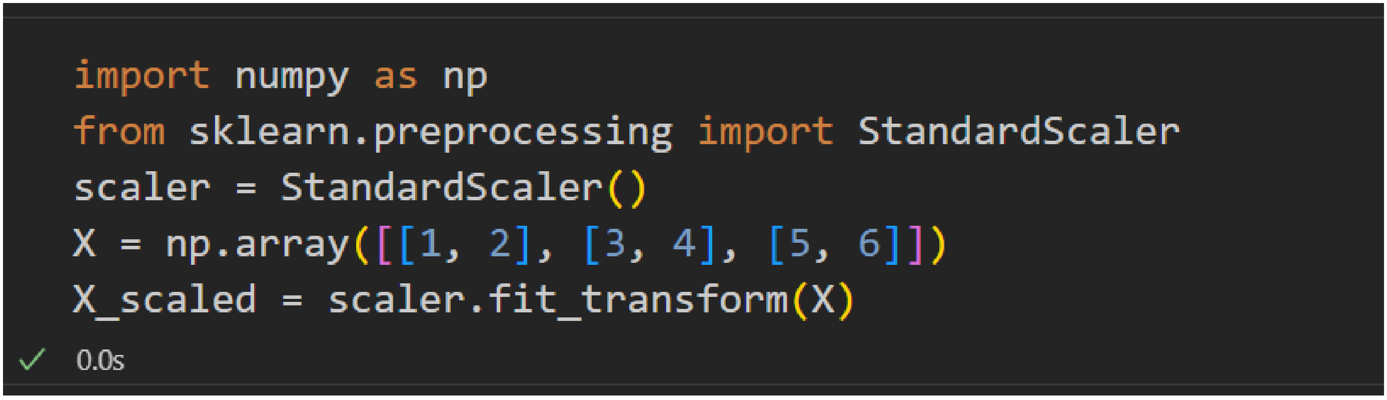
- **Continuous Variables**: Numeric values with infinite possibilities (e.g., height, weight).
- **Categorical Variables**: Discrete values (e.g., gender, color).

21) What is feature scaling? How does it help in Machine Learning?

- Feature scaling standardizes or normalizes data to ensure all features contribute equally. It prevents features with large ranges from dominating.

22) How do we perform scaling in Python?

- Using StandardScaler or MinMaxScaler from sklearn.preprocessing.



```
import numpy as np
from sklearn.preprocessing import StandardScaler
scaler = StandardScaler()
X = np.array([[1, 2], [3, 4], [5, 6]])
X_scaled = scaler.fit_transform(X)
```

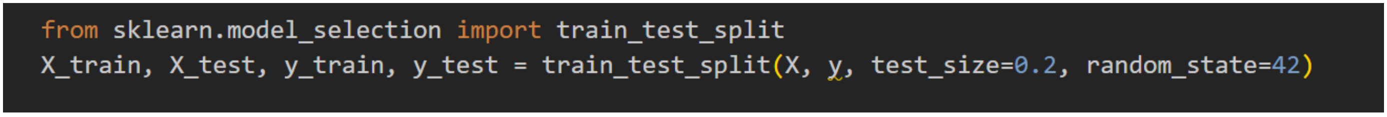
```
✓ 0.0s
```

23) What is sklearn.preprocessing?

- sklearn.preprocessing provides utilities for feature scaling, encoding, and transformation.

24) How do we split data for model fitting (training and testing) in Python?

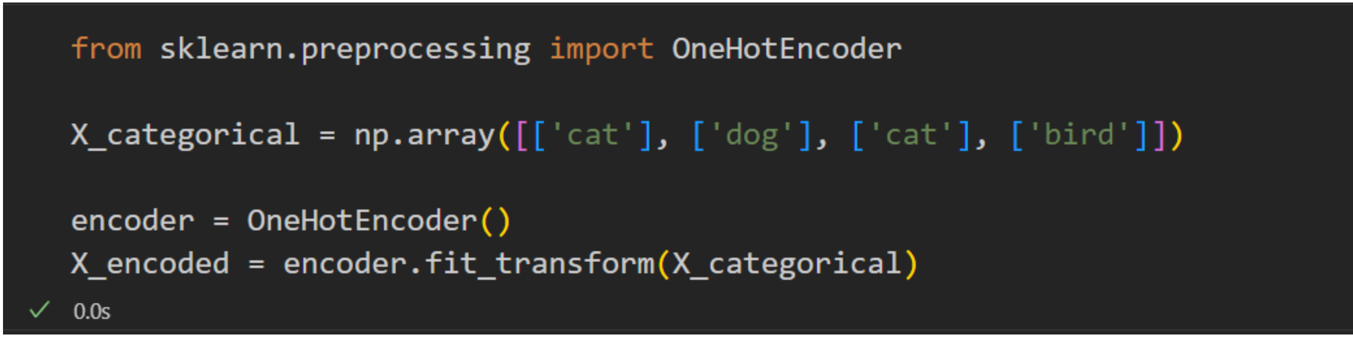
- Using train_test_split from sklearn.model_selection.



```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

25) Explain data encoding?

- Data encoding is the process of converting categorical data into numerical format so that machine learning models can process it effectively. There are two main types:
  a) **Label Encoding** – Assigns a unique number to each category (e.g., "Red" → 0, "Blue" → 1).
  b) **One-Hot Encoding** – Creates binary columns for each category (e.g., "Red" → [1,0,0], "Blue" → [0,1,0]).



```python
from sklearn.preprocessing import OneHotEncoder

X_categorical = np.array([['cat'], ['dog'], ['cat'], ['bird']])

encoder = OneHotEncoder()
X_encoded = encoder.fit_transform(X_categorical)
```