prediction-system-via-cv-analysis

September 11, 2023

[2]:
```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error

# Sample mock dataset
data = {
    'cv_text': ["I am a highly motivated individual with a strong work ethic...
 ↪",
               "During my time at XYZ company, I managed a team of...",
               "I have a background in computer science and have worked on...",
               ],
    'personality_trait': [4.5, 3.2, 4.8]  # Mock personality scores (should␣
 ↪match the number of CV texts)
}

# Create a DataFrame
df = pd.DataFrame(data)

# Feature extraction (mock implementation)
# In a real-world scenario, you would use NLP techniques for this step
df['feature'] = df['cv_text'].apply(lambda x: len(x.split()))

# Split data into features (X) and target (y)
X = df[['feature']].values
y = df['personality_trait'].values

# Split the data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,␣
 ↪random_state=42)

# Create and train the linear regression model
model = LinearRegression()
model.fit(X_train, y_train)

# Make predictions on the test set
y_pred = model.predict(X_test)
```

```python

# Calculate Mean Squared Error (MSE) to evaluate the model
mse = mean_squared_error(y_test, y_pred)
print(f"Mean Squared Error: {mse:.2f}")

# Input new CV features to predict personality trait
new_cv_feature = [[150]]  # Replace with your input data
predicted_personality = model.predict(new_cv_feature)
print(f"Predicted personality trait score: {predicted_personality[0]:.2f}")
```

```
Mean Squared Error: 0.25
Predicted personality trait score: 4.00
```