AI Machine Learning

Machine Learning Project

Presented By Joel Austen

next slide →

Workflow

1. 
2. 
3. 
4. 

INTRODUCTION → DATA PREPARATION → EXPLORATORY DATA ANALYSIS → FEATURE ENGINEERING

DATA SPLITTING ← MODEL IMPLIMENTATION ← MODEL EVALUTAION ← CONCLUSION

5. 
6. 
7. 
8. 

next slide →

Introduction

Assignment : *Compare the accuracy of Linear Regression and Decision Tree models in predicting student scores based on study hours.*

next slide →

Import Dataset

```
[ ]  # read the dataset using pandas
     data = pd.read_csv('student_scores.csv')
     data.head()
```

	Hours	Scores
0	2.5	21
1	5.1	47
2	3.2	27
3	8.5	75
4	3.5	30

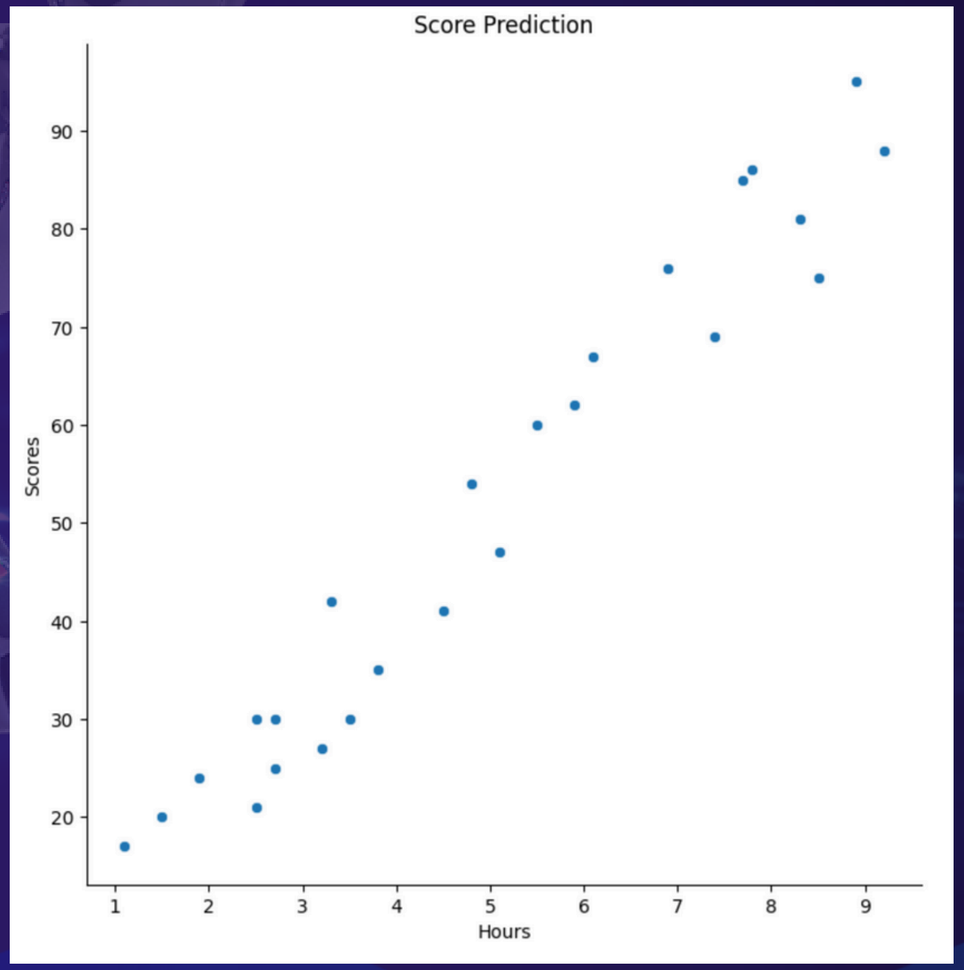
Data Preparation

Download and import the dataset containing two columns: Hours and Scores.

next slide →

Scatter Plot

```
plt.figure(figsize=(9,5))
sns.pairplot(data,x_vars=['Hours'],y_vars=['Scores'],size=7,kind='scatter')
plt.xlabel('Hours')
plt.ylabel('Scores')
plt.title('Score Prediction')
plt.show()
```



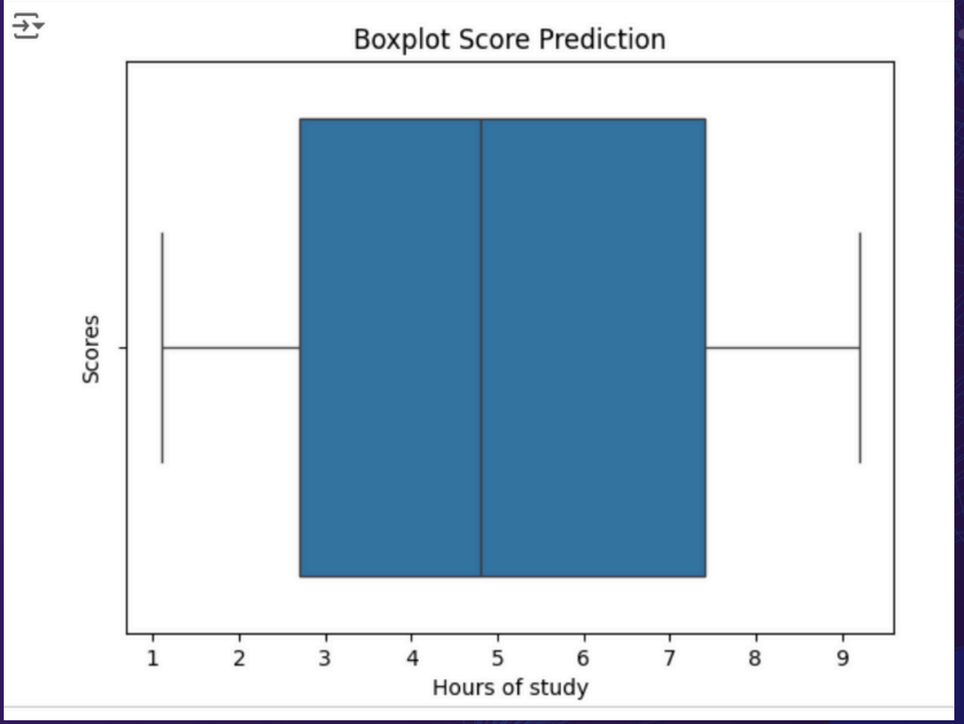
Exploratory Data Analysis

Visualizing the relationship between study hours and scores.

next slide →

Check for Outliers

```python
import seaborn as sns
import matplotlib.pyplot as plt

# Outlier Analysis
sns.boxplot(x="Hours", data=df)

# Menambahkan label sumbu dan judul
plt.xlabel("Hours of study")
plt.ylabel("Scores")
plt.title("Boxplot Score Prediction")

# Menampilkan plot
plt.show()
```



Feature Engineering

Using bloxplot to identify and handle outliers appropriately.

next slide →

Data Splitting

Split the data into training sets :

- *X_train*
- *X_test*
- *y_train*
- *y_test*

```
# Import machine learning datafrom scikit learn
from sklearn.model_selection import train_test_split
```

```
# Split the data for train and test
# 75:25 or 70:30 or 80:20 or 85:15, train size > test size
X_train,X_test,y_train,y_test = train_test_split(X,y,train_size=0.75,random_state=42)
```

next slide →

Model Implementation

Linear Regression

```
[ ]  # Fitting the model using Linear Regression
     lr_model = LinearRegression()
     lr_model.fit(X_train,y_train)
```

```
▾ LinearRegression
LinearRegression()
```

Linear Regression: *A Linear Regression model is trained on the training data to predict the scores based on the number of hours studied.*

Decision Tree Regressor

```
[ ]  from sklearn.tree import DecisionTreeRegressor
```

```
[ ]  dt_model = DecisionTreeRegressor()
     dt_model.fit(X_train,y_train)
```

```
▾ DecisionTreeRegressor
DecisionTreeRegressor()
```

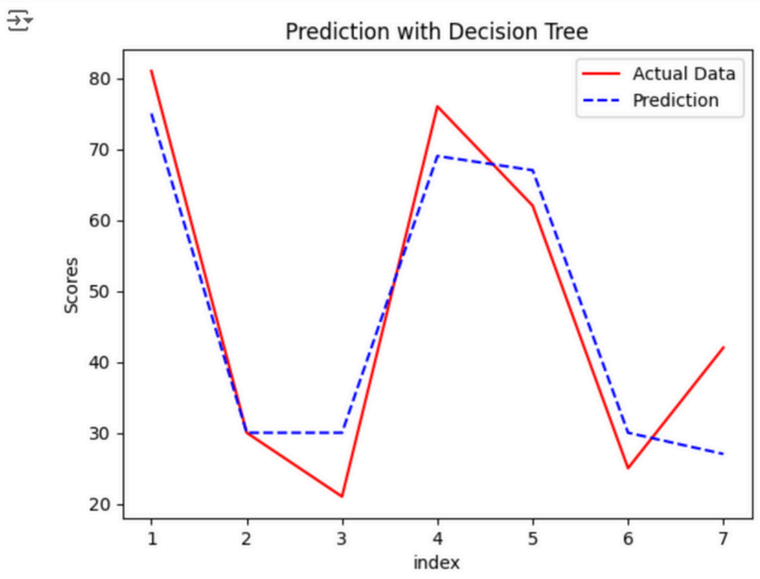
Decision Tree Regressor: *A Decision Tree Regressor model is also trained on the same training data for comparison.*

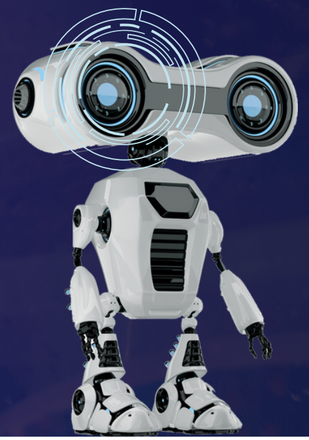
next slide →

Model Evaluation

Prediction and Plotting

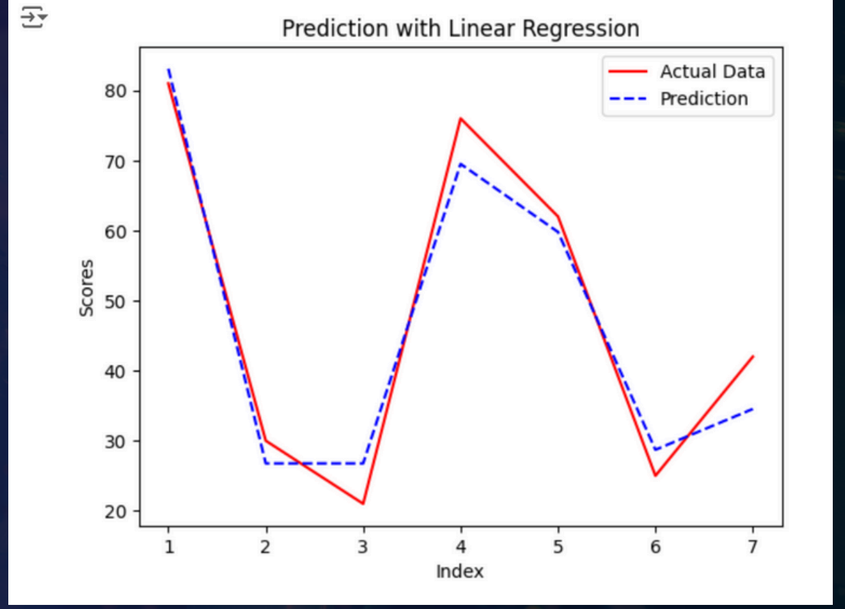
```
[ ]  # Plotting the actual and predicted values

     c = [i for i in range (1,len(y_test)+1,1)]
     plt.plot(c,y_test,color='r',linestyle='-',label='Actual Data')
     plt.plot(c,y_pred_dt,color='b',linestyle='dashed',label='Prediction')
     plt.xlabel('index')
     plt.ylabel('Scores')
     plt.title('Prediction with Decision Tree')
     plt.legend()
     plt.show()
```





next slide →

```
[ ]  # Plotting the actual and predicted values
     c = [i for i in range (1,len(y_test)+1,1)]
     plt.plot(c,y_test,color='r',linestyle='-',label='Actual Data')
     plt.plot(c,y_pred,color='b',linestyle='dashed',label='Prediction')
     plt.xlabel('Index')
     plt.ylabel('Scores')
     plt.title('Prediction with Linear Regression')
     plt.legend()
     plt.show()
```



Linear Regression Predictions: *Predictions are made using the Linear Regression model and plotted against the actual values.*

Decision Tree Predictions: *Predictions are also made using the Decision Tree Regressor model and plotted for comparison.*

Conclusion

```
[ ]  # Calculate R square value
     rsq = r2_score(y_test,y_pred)
```

```
[ ]  print('r square linear regression:',rsq)
```

```
r square linear regression: 0.9553509219739938
```

R-Squared Values: The R-Squared values are printed for comparison to identify the better-performing model.

```
[ ]  # Calculate R square value
     rsq_dt = r2_score(y_test,y_pred_dt)
```

```
[ ]  print('r square Decision Tree Results:',rsq_dt)
```

```
r square Decision Tree Results: 0.8803859268443893
```

Model Accuracy Percentage:

- ***Linear Regression: 95%***
- ***Decision Tree: 88%***

As a conclusion, Linear Regression is preferred because it has a higher accuracy rate of 95% compared to Decision Tree's 88%.

next slide →

Thanks!

I hope you find this informative and engaging