Day 4 – Machine Learning and Pandas with California Housing Dataset

📅 **Date:** 27 June 2025

📘 Today’s Highlights

Today, we were introduced to the basics of **Machine Learning** and its types.
Alongside this, we implemented a simple machine learning model using the **California Housing dataset** to understand the practical applications of **Pandas** and **Scikit-learn** libraries in Python.

□ Machine Learning Overview

Machine Learning (ML) is a subset of Artificial Intelligence (AI) that enables systems to learn from data and improve over time without being explicitly programmed.

Types of Machine Learning:

- **Supervised Learning**: Uses labeled data to train models (e.g., regression, classification)
- **Unsupervised Learning**: Finds patterns in data without labels (e.g., clustering)
- **Reinforcement Learning**: Learns through feedback and reward-based systems

---

🛠□ Practical Implementation

We worked on a real-world dataset called **California Housing** using Pandas and Scikit-learn.
The dataset contains information such as population, income, and number of rooms.
We applied **multiple linear regression** to predict housing prices.

📌 Python Code Used

```
import pandas as pd
import numpy as np
from sklearn.datasets import fetch_california_housing
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error, r2_score
import joblib

# Load the California Housing dataset
```

```
california = fetch_california_housing()
X = california.data
y = california.target

# Convert to DataFrame
df = pd.DataFrame(X, columns=california.feature_names)
print("First few rows of the dataset:")
print(df.head())

# Train-test split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,
random_state=42)

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

# Predict and evaluate
y_pred = model.predict(X_test)
mse = mean_squared_error(y_test, y_pred)
r2 = r2_score(y_test, y_pred)

# Print results
for feature, coef in zip(california.feature_names, model.coef_):
    print(f"{feature}: {coef:.2f}")
print(f"Mean Squared Error: {mse:.2f}")
print(f"R-squared Score: {r2:.2f}")

# Save model
joblib.dump(model, 'housing_model.pkl')
print("Model saved as 'housing_model.pkl'")
```

📊 Output & Results

The model returned **coefficients** for each feature in the dataset and calculated two evaluation metrics:

- **Mean Squared Error (MSE)**: Measures average error
- **R-squared Score (R^2)**: Measures goodness of fit

Finally, the trained model was saved as a `.pkl` file using the `joblib` library.