MAINCRAFTS TECHNOLOGY

www.maincrafts.com
hr@maincrafts.com

Artificial Intelligence & Machine Learning- Task 1

Build & Evaluate a Linear Regression Model (House Price Predictor)

Objective:

Introduce the ML workflow: data loading, exploration, preprocessing, training, evaluation, and reporting. You will train a linear regression model on the **California Housing** dataset and create a short report (notebook + slides).

Why this task?

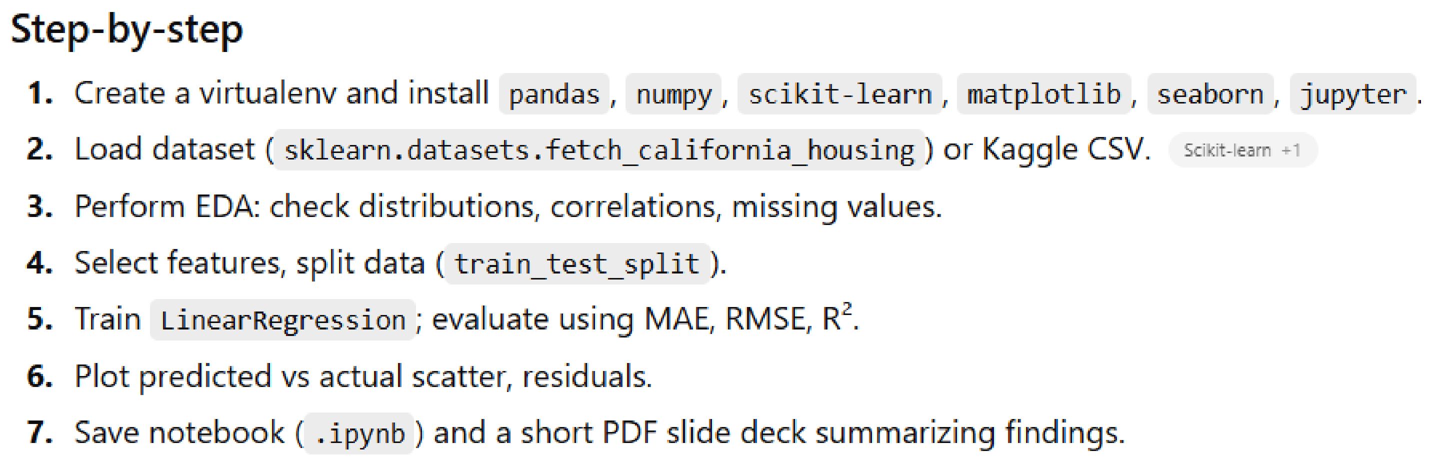
It introduces the entire ML lifecycle in a short, reproducible project ideal for portfolios.

Skills you'll gain

- Use Python, pandas, scikit-learn.
- Exploratory data analysis (EDA) and visualization.
- Train/test split, model training (LinearRegression), evaluation (MAE, RMSE, R^2).
- Save model and present results in a Jupyter Notebook.
-

Dataset

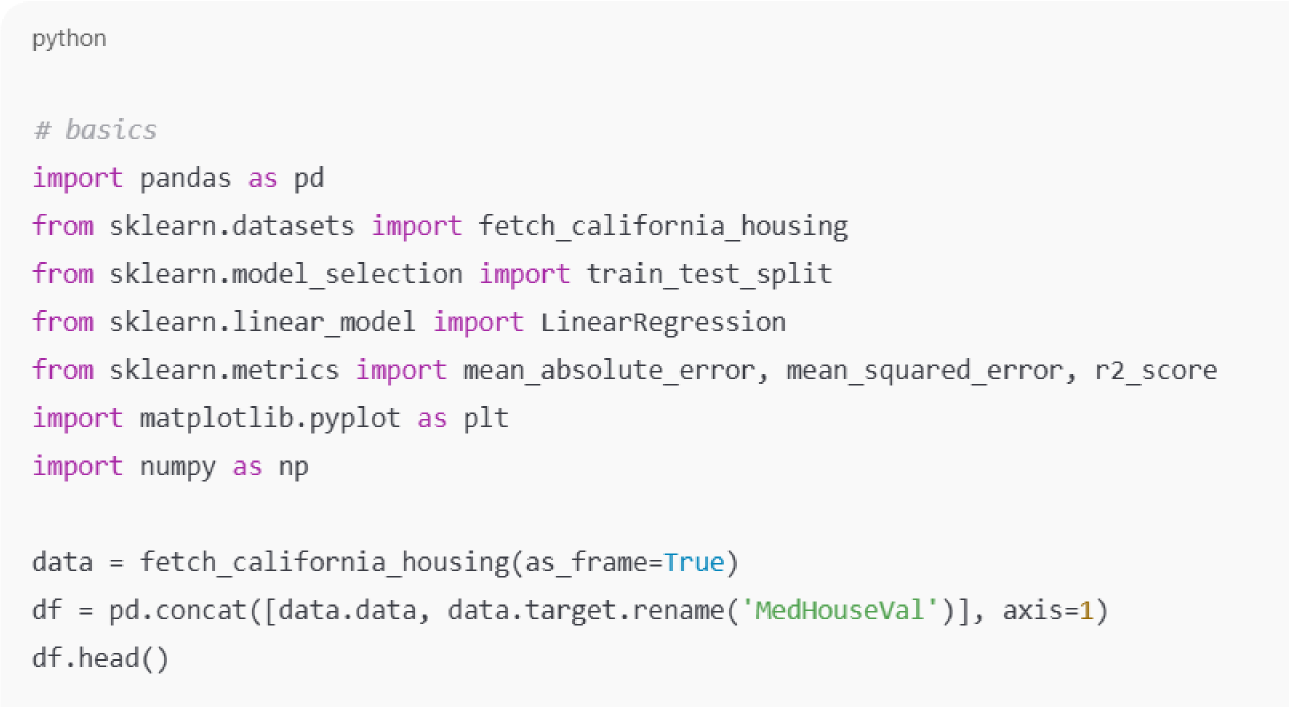
Use the California Housing dataset [(built into scikit-learn or available on Kaggle).]()



Step-by-step

1. Create a virtualenv and install `pandas`, `numpy`, `scikit-learn`, `matplotlib`, `seaborn`, `jupyter`.
2. Load dataset (`sklearn.datasets.fetch_california_housing`) or Kaggle CSV. Scikit-learn +1
3. Perform EDA: check distributions, correlations, missing values.
4. Select features, split data (`train_test_split`).
5. Train `LinearRegression`; evaluate using MAE, RMSE, R^2.
6. Plot predicted vs actual scatter, residuals.
7. Save notebook (`.ipynb`) and a short PDF slide deck summarizing findings.

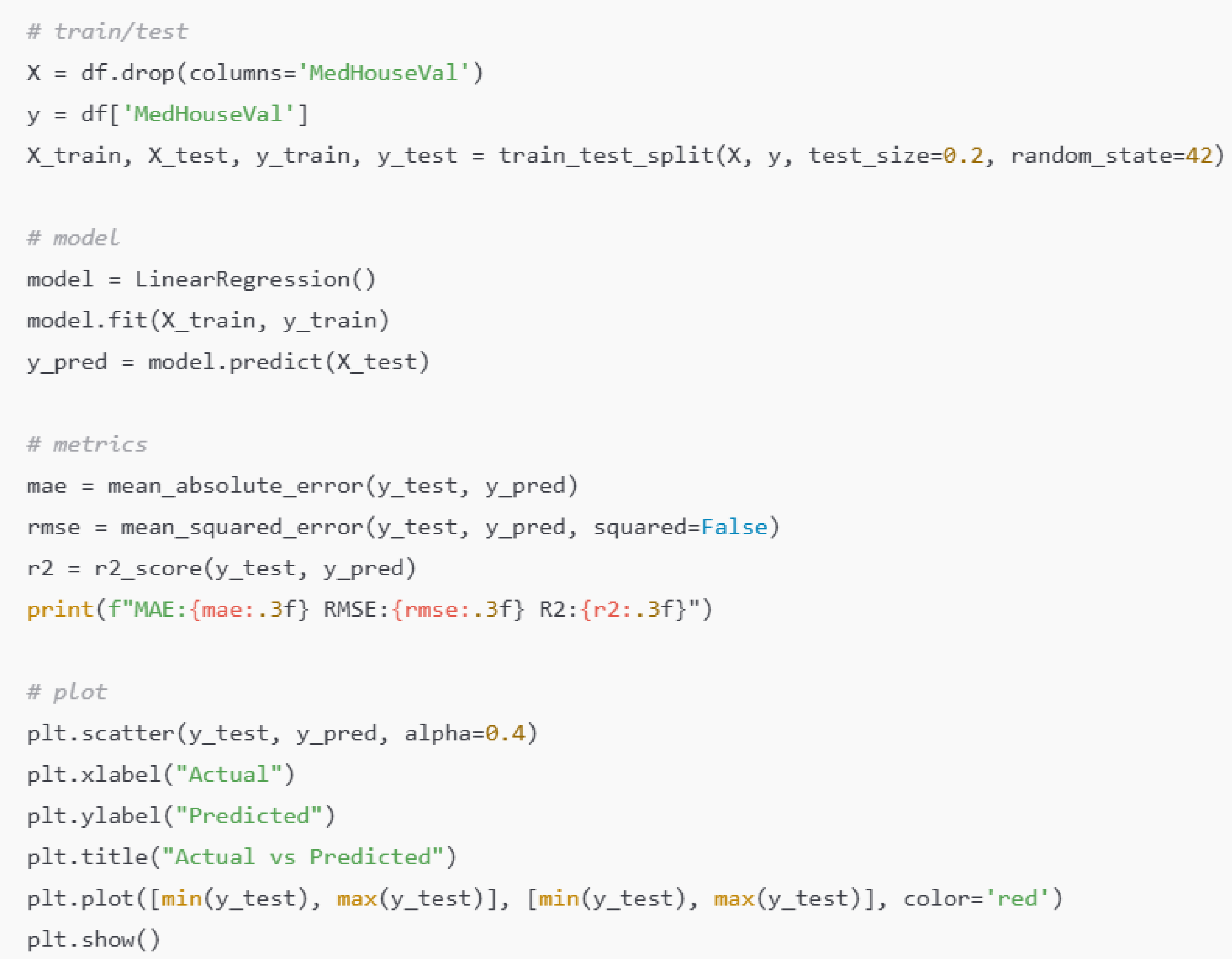
Starter notebook (key code blocks)



python

```python
# basics
import pandas as pd
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_absolute_error, mean_squared_error, r2_score
import matplotlib.pyplot as plt
import numpy as np

data = fetch_california_housing(as_frame=True)
df = pd.concat([data.data, data.target.rename('MedHouseVal')], axis=1)
df.head()
```



```python
# train/test
X = df.drop(columns='MedHouseVal')
y = df['MedHouseVal']
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# model
model = LinearRegression()
model.fit(X_train, y_train)
y_pred = model.predict(X_test)

# metrics
mae = mean_absolute_error(y_test, y_pred)
rmse = mean_squared_error(y_test, y_pred, squared=False)
r2 = r2_score(y_test, y_pred)
print(f"MAE:{mae:.3f} RMSE:{rmse:.3f} R2:{r2:.3f}")

# plot
plt.scatter(y_test, y_pred, alpha=0.4)
plt.xlabel("Actual")
plt.ylabel("Predicted")
plt.title("Actual vs Predicted")
plt.plot([min(y_test), max(y_test)], [min(y_test), max(y_test)], color='red')
plt.show()
```

Deliverables:

1. Jupyter Notebook (task1_ml_linear_regression.ipynb) containing code, plots, and comments.
2. Short PDF report (2–4 pages) summarizing EDA, model, metrics, and improvement ideas.
3. (Optional) Saved model pickle and a small UI script to predict on new inputs.

Task-1 Support Resources

Topic: Linear Regression Model (California Housing Dataset)

▶ YouTube Tutorials (Best for Complete Beginners)

Linear Regression

1. **Linear Regression Explained Clearly**
   https://www.youtube.com/watch?v=E5RjzSK0fvY
2. **Linear Regression in Python (Hands-on Tutorial)**
   https://www.youtube.com/watch?v=J_LnPL3Qg70
3. **Scikit-Learn Crash Course (Regression Model)**
   https://www.youtube.com/watch?v=0Lt9w-BxKFQ
4. **Machine Learning Full Playlist (Krish Naik — India's #1 ML Teacher)**
   https://www.youtube.com/playlist?list=PLZoTAELRMXVPGU70ZGsckrMdr0FteeRUi
5. **ML for Beginners (FreeCodeCamp — 8-Hour Course)**
   https://www.youtube.com/watch?v=ukzFI9rgwfU

📘 Free Study Material & Docs

- scikit-learn Linear Regression Docs
  https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LinearRegression.html
- California Housing Dataset Info
  https://scikit-learn.org/stable/datasets/real_world.html#california-housing-dataset
- Kaggle Intro to Machine Learning Course
  https://www.kaggle.com/learn/intro-to-machine-learning
- Python for Data Science (freeCodeCamp)
  https://www.youtube.com/watch?v=LHBE6Q9XlzI