36106_25AU-AT1_25589351_experiment_2

March 29, 2025

1 Experiment Notebook

1.1 0. Setup Environment

1.1.1 0.a Install Mandatory Packages

Do not modify this code before running it

[ ]:
```
# Do not modify this code

import os
import sys
from pathlib import Path

COURSE = "36106"
ASSIGNMENT = "AT1"
DATA = "data"

asgmt_path = f"{COURSE}/assignment/{ASSIGNMENT}"
root_path = "./"

print("###### Install required Python packages ######")
! pip install -r https://raw.githubusercontent.com/aso-uts/labs_datasets/main/
  ↪36106-mlaa/requirements.txt

if os.getenv("COLAB_RELEASE_TAG"):

    from google.colab import drive
    from pathlib import Path

    print("\n###### Connect to personal Google Drive ######")
    gdrive_path = "/content/gdrive"
    drive.mount(gdrive_path)
    root_path = f"{gdrive_path}/MyDrive/"

print("\n###### Setting up folders ######")
folder_path = Path(f"{root_path}/{asgmt_path}/") / DATA
```

```
folder_path.mkdir(parents=True, exist_ok=True)
print(f"\nYou can now save your data files in: {folder_path}")

if os.getenv("COLAB_RELEASE_TAG"):
    %cd {folder_path}
```

```
###### Install required Python packages ######
Requirement already satisfied: pandas==2.2.2 in /usr/local/lib/python3.11/dist-
packages (from -r https://raw.githubusercontent.com/aso-
uts/labs_datasets/main/36106-mlaa/requirements.txt (line 1)) (2.2.2)
Requirement already satisfied: scikit-learn==1.6.1 in
/usr/local/lib/python3.11/dist-packages (from -r
https://raw.githubusercontent.com/aso-
uts/labs_datasets/main/36106-mlaa/requirements.txt (line 2)) (1.6.1)
Requirement already satisfied: altair==5.5.0 in /usr/local/lib/python3.11/dist-
packages (from -r https://raw.githubusercontent.com/aso-
uts/labs_datasets/main/36106-mlaa/requirements.txt (line 3)) (5.5.0)
Requirement already satisfied: numpy>=1.23.2 in /usr/local/lib/python3.11/dist-
packages (from pandas==2.2.2->-r https://raw.githubusercontent.com/aso-
uts/labs_datasets/main/36106-mlaa/requirements.txt (line 1)) (2.0.2)
Requirement already satisfied: python-dateutil>=2.8.2 in
/usr/local/lib/python3.11/dist-packages (from pandas==2.2.2->-r
https://raw.githubusercontent.com/aso-
uts/labs_datasets/main/36106-mlaa/requirements.txt (line 1)) (2.8.2)
Requirement already satisfied: pytz>=2020.1 in /usr/local/lib/python3.11/dist-
packages (from pandas==2.2.2->-r https://raw.githubusercontent.com/aso-
uts/labs_datasets/main/36106-mlaa/requirements.txt (line 1)) (2025.1)
Requirement already satisfied: tzdata>=2022.7 in /usr/local/lib/python3.11/dist-
packages (from pandas==2.2.2->-r https://raw.githubusercontent.com/aso-
uts/labs_datasets/main/36106-mlaa/requirements.txt (line 1)) (2025.1)
Requirement already satisfied: scipy>=1.6.0 in /usr/local/lib/python3.11/dist-
packages (from scikit-learn==1.6.1->-r https://raw.githubusercontent.com/aso-
uts/labs_datasets/main/36106-mlaa/requirements.txt (line 2)) (1.14.1)
Requirement already satisfied: joblib>=1.2.0 in /usr/local/lib/python3.11/dist-
packages (from scikit-learn==1.6.1->-r https://raw.githubusercontent.com/aso-
uts/labs_datasets/main/36106-mlaa/requirements.txt (line 2)) (1.4.2)
Requirement already satisfied: threadpoolctl>=3.1.0 in
/usr/local/lib/python3.11/dist-packages (from scikit-learn==1.6.1->-r
https://raw.githubusercontent.com/aso-
uts/labs_datasets/main/36106-mlaa/requirements.txt (line 2)) (3.6.0)
Requirement already satisfied: jinja2 in /usr/local/lib/python3.11/dist-packages
(from altair==5.5.0->-r https://raw.githubusercontent.com/aso-
uts/labs_datasets/main/36106-mlaa/requirements.txt (line 3)) (3.1.6)
Requirement already satisfied: jsonschema>=3.0 in
/usr/local/lib/python3.11/dist-packages (from altair==5.5.0->-r
https://raw.githubusercontent.com/aso-
uts/labs_datasets/main/36106-mlaa/requirements.txt (line 3)) (4.23.0)
```

```
Requirement already satisfied: narwhals>=1.14.2 in
/usr/local/lib/python3.11/dist-packages (from altair==5.5.0->-r
https://raw.githubusercontent.com/aso-
uts/labs_datasets/main/36106-mlaa/requirements.txt (line 3)) (1.31.0)
Requirement already satisfied: packaging in /usr/local/lib/python3.11/dist-
packages (from altair==5.5.0->-r https://raw.githubusercontent.com/aso-
uts/labs_datasets/main/36106-mlaa/requirements.txt (line 3)) (24.2)
Requirement already satisfied: typing-extensions>=4.10.0 in
/usr/local/lib/python3.11/dist-packages (from altair==5.5.0->-r
https://raw.githubusercontent.com/aso-
uts/labs_datasets/main/36106-mlaa/requirements.txt (line 3)) (4.12.2)
Requirement already satisfied: attrs>=22.2.0 in /usr/local/lib/python3.11/dist-
packages (from jsonschema>=3.0->altair==5.5.0->-r
https://raw.githubusercontent.com/aso-
uts/labs_datasets/main/36106-mlaa/requirements.txt (line 3)) (25.3.0)
Requirement already satisfied: jsonschema-specifications>=2023.03.6 in
/usr/local/lib/python3.11/dist-packages (from jsonschema>=3.0->altair==5.5.0->-r
https://raw.githubusercontent.com/aso-
uts/labs_datasets/main/36106-mlaa/requirements.txt (line 3)) (2024.10.1)
Requirement already satisfied: referencing>=0.28.4 in
/usr/local/lib/python3.11/dist-packages (from jsonschema>=3.0->altair==5.5.0->-r
https://raw.githubusercontent.com/aso-
uts/labs_datasets/main/36106-mlaa/requirements.txt (line 3)) (0.36.2)
Requirement already satisfied: rpds-py>=0.7.1 in /usr/local/lib/python3.11/dist-
packages (from jsonschema>=3.0->altair==5.5.0->-r
https://raw.githubusercontent.com/aso-
uts/labs_datasets/main/36106-mlaa/requirements.txt (line 3)) (0.23.1)
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.11/dist-
packages (from python-dateutil>=2.8.2->pandas==2.2.2->-r
https://raw.githubusercontent.com/aso-
uts/labs_datasets/main/36106-mlaa/requirements.txt (line 1)) (1.17.0)
Requirement already satisfied: MarkupSafe>=2.0 in
/usr/local/lib/python3.11/dist-packages (from jinja2->altair==5.5.0->-r
https://raw.githubusercontent.com/aso-
uts/labs_datasets/main/36106-mlaa/requirements.txt (line 3)) (3.0.2)

###### Connect to personal Google Drive ######
Mounted at /content/gdrive

###### Setting up folders ######

You can now save your data files in:
/content/gdrive/MyDrive/36106/assignment/AT1/data
/content/gdrive/MyDrive/36106/assignment/AT1/data
```

1.1.2 0.b Disable Warnings Messages

Do not modify this code before running it

[ ]:
```
import warnings
warnings.simplefilter(action='ignore', category=FutureWarning)
```

1.1.3 0.c Install Additional Packages

If you are using additional packages, you need to install them here using the command:
`! pip install <package_name>`

[ ]:
```
# <Student to fill this section>
```

1.1.4 0.d Import Packages

[ ]:
```
import ipywidgets as widgets
import pandas as pd
import altair as alt
import numpy as np
from sklearn.metrics import mean_squared_error, r2_score, mean_absolute_error
from itertools import product
```

1.2 A. Project Description

[ ]:
```
# @title Student Information
wgt_student_name = widgets.Text(
    value="Fatemeh Elyasifar",
    placeholder='<student to fill this section>',
    description='Student Name:',
    style={'description_width': 'initial'},
    disabled=False
)

wgt_student_id = widgets.Text(
    value="25589351",
    placeholder='<student to fill this section>',
    description='Student Id:',
    style={'description_width': 'initial'},
    disabled=False
)

widgets.HBox([wgt_student_name, wgt_student_id])
```

```
HBox(children=(Text(value='Fatemeh Elyasifar', description='Student Name:',␣
 ↪placeholder='<student to fill this…
```

[ ]:
```
print("Student Name:", wgt_student_name.value)
print("Student Id:", wgt_student_id.value)
```

Student Name: Fatemeh Elyasifar
Student Id: 25589351

[ ]:
```
# @title Experiment ID

wgt_experiment_id = widgets.BoundedIntText(
    value="2",
    min=0,
    max=3,
    step=1,
    description='Experiment ID:',
    style={'description_width': 'initial'},
    disabled=False
)
wgt_experiment_id
```

```
BoundedIntText(value=2, description='Experiment ID:', max=3,␣
 ↪style=DescriptionStyle(description_width='initial…
```

[ ]:
```
print("Experiment ID:", wgt_experiment_id.value)
```

```
Experiment ID: 2
```

[ ]:
```
# @title Business Objective

wgt_business_objective = widgets.Textarea(
    value="The main objective is to develop a machine learning model that␣
 ↪accurately predicts rental prices specifically for affordable properties in␣
 ↪Australia, excluding luxury homes. This model aims to assist real estate␣
 ↪agencies, property investors, and tenants in making informed decisions based␣
 ↪on the features of affordable housing and market trends. The key success␣
 ↪metric is RMSE, with the goal of achieving an RMSE score of less than 16 on␣
 ↪the validation set, which quantifies the prediction error. Additionally,␣
 ↪feature tuning will be necessary to optimise the model's performance and␣
 ↪ensure accurate predictions based on the most relevant property␣
 ↪characteristics.",
    placeholder='<student to fill this section>',
    description='Business Objective:',
    disabled=False,
    style={'description_width': 'initial'},
    layout=widgets.Layout(height="100%", width="auto")
)
wgt_business_objective
```

```
Textarea(value='The main objective is to develop a machine learning model that␣
 ↪accurately predicts rental pric…
```

[ ]:
```
print("Business Objective:", wgt_business_objective.value)
```

Business Objective: The main objective is to develop a machine learning model that accurately predicts rental prices specifically for affordable properties in Australia, excluding luxury homes. This model aims to assist real estate agencies, property investors, and tenants in making informed decisions based on the features of affordable housing and market trends. The key success metric is RMSE, with the goal of achieving an RMSE score of less than 16 on the validation set, which quantifies the prediction error. Additionally, feature tuning will be necessary to optimise the model's performance and ensure accurate predictions based on the most relevant property characteristics.

1.3 B. Experiment Description

[ ]:
```
# @title Experiment Hypothesis

wgt_experiment_hypothesis = widgets.Textarea(
    value="Implementing ElasticNet regularisation will improve the accuracy of␣
 ↪rental price predictions by reducing overfitting, leading to better␣
 ↪generalisation and more effective pricing strategies for a real estate␣
 ↪business. Null Hypothesis (H ): Varying alpha and l1_ratio does not␣
 ↪significantly impact the RMSE of the ElasticNet model.",
    placeholder='<student to fill this section>',
    description='Experiment Hypothesis:',
    disabled=False,
    style={'description_width': 'initial'},
    layout=widgets.Layout(height="100%", width="auto")
)
wgt_experiment_hypothesis
```

```
Textarea(value='Implementing ElasticNet regularisation will improve the accuracy␣
 ↪of rental price predictions b…
```

[ ]:
```
print("Experiment Hypothesis:", wgt_experiment_hypothesis.value)
```

```
Experiment Hypothesis: Implementing ElasticNet regularisation will improve the
accuracy of rental price predictions by reducing overfitting, leading to better
generalisation and more effective pricing strategies for a real estate business.
Null Hypothesis (H ): Varying alpha and l1_ratio does not significantly impact
the RMSE of the ElasticNet model.
```

[ ]:
```
# @title Experiment Expectations

wgt_experiment_expectations = widgets.Textarea(
```

```
    value="Train multiple ElasticNet models with varying alpha and l1_ratio␣
 ↪values. Analyse how different levels of regularisation affect RMSE. Identify␣
 ↪an optimal combination of alpha and l1_ratio that yields the best trade-off␣
 ↪between bias and variance. Provide recommendations on regularisation for␣
 ↪further experiments.",
    placeholder='<student to fill this section>',
    description='Experiment Expectations:',
    disabled=False,
    style={'description_width': 'initial'},
    layout=widgets.Layout(height="100%", width="auto")
)
wgt_experiment_expectations
```

```
Textarea(value='Train multiple ElasticNet models with varying alpha and l1_ratio␣
 ↪values. Analyse how different…
```

[ ]:
```
print("Experiment Expectations:", wgt_experiment_expectations.value)
```

```
Experiment Expectations: Train multiple ElasticNet models with varying alpha and
l1_ratio values. Analyse how different levels of regularisation affect RMSE.
Identify an optimal combination of alpha and l1_ratio that yields the best
trade-off between bias and variance. Provide recommendations on regularisation
for further experiments.
```

1.4 C. Data Understanding

1.4.1 C.1 Load Datasets

Do not change this code

[ ]:
```
# Load training data
X_train = pd.read_csv(folder_path / 'X_train.csv')
y_train = pd.read_csv(folder_path / 'y_train.csv')
```

[ ]:
```
# Load validation data
X_val = pd.read_csv(folder_path / 'X_val.csv')
y_val = pd.read_csv(folder_path / 'y_val.csv')
```

[ ]:
```
# Load testing data
X_test = pd.read_csv(folder_path / 'X_test.csv')
y_test = pd.read_csv(folder_path / 'y_test.csv')
```

1.5 D. Feature Selection

```
[ ]: # <Student to fill this section>

features_list =␣
 ↪['number_of_bedrooms','floor_area','number_of_bathrooms','month','level_numerator','level_r
 ↪Family','tenancy_preference_Family','suburb_Brisbane','suburb_Canberra','suburb_Melbourne',
```

```
[ ]: # @title Feature Selection Explanation

wgt_feat_selection_explanation = widgets.Textarea(
    value="These attributes offer a detailed perspective on the property's␣
 ↪features and market conditions, selected to highlight the primary factors␣
 ↪affecting rental rates.",
    placeholder='<student to fill this section>',
    description='Feature Selection Explanation:',
    disabled=False,
    style={'description_width': 'initial'},
    layout=widgets.Layout(height="100%", width="auto")
)
wgt_feat_selection_explanation
```

```
Textarea(value="These attributes offer a detailed perspective on the property's␣
 ↪features and market conditions…
```

```
[ ]: print("Feature Selection Explanation:", wgt_feat_selection_explanation.value)
```

```
Feature Selection Explanation: These attributes offer a detailed perspective on
the property's features and market conditions, selected to highlight the primary
factors affecting rental rates.
```

1.6 E. Train Machine Learning Model

1.6.1 E.1 Import Algorithm

Provide some explanations on why you believe this algorithm is a good fit

```
[ ]: # <Student to fill this section>
from sklearn.linear_model import ElasticNet
```

```
[ ]: # @title Algorithm Selection Explanation

wgt_algo_selection_explanation = widgets.Textarea(
    value="The ElasticNet algorithm was selected because it combines both Lasso␣
 ↪(L1) and Ridge (L2) regularisation, which helps improve model performance by␣
 ↪balancing feature selection and preventing overfitting.",
    placeholder='<student to fill this section>',
```

```
    description='Algorithm Selection Explanation:',
    disabled=False,
    style={'description_width': 'initial'},
    layout=widgets.Layout(height="100%", width="auto")
)
wgt_algo_selection_explanation
```

```
Textarea(value='The ElasticNet algorithm was selected because it combines both␣
 ↪Lasso (L1) and Ridge (L2) regul…
```

[ ]:
```
print("Algorithm Selection Explanation:", wgt_algo_selection_explanation.value)
```

```
Algorithm Selection Explanation: The ElasticNet algorithm was selected because
it combines both Lasso (L1) and Ridge (L2) regularisation, which helps improve
model performance by balancing feature selection and preventing overfitting.
```

1.6.2 E.2 Set Hyperparameters

Provide some explanations on why you believe this algorithm is a good fit

[ ]:
```
# <Student to fill this section>

alpha_values = [0.0002, 0.001, 0.005, 0.1, 1]
l1_ratio_values = [0.1, 0.5, 0.9, 1]
```

[ ]:
```
# @title Hyperparameters Selection Explanation

wgt_hyperparams_selection_explanation = widgets.Textarea(
    value="The hyperparameters (alpha_values and l1_ratio_values) were chosen␣
 ↪to test different regularisation strengths and Lasso/Ridge mixes. Alpha␣
 ↪controls the regularisation strength, and l1_ratio adjusts the balance␣
 ↪between Lasso (L1) and Ridge (L2) regularisation. This helps find the best␣
 ↪model that fits the data without overfitting.",
    placeholder='<student to fill this section>',
    description='Hyperparameters Selection Explanation:',
    disabled=False,
    style={'description_width': 'initial'},
    layout=widgets.Layout(height="100%", width="auto")
)
wgt_hyperparams_selection_explanation
```

```
Textarea(value='The hyperparameters (alpha_values and l1_ratio_values) were␣
 ↪chosen to test different regularis…
```

[ ]:
```
print("Hyperparameters Selection Explanation:",␣
 ↪wgt_hyperparams_selection_explanation.value)
```

```
Hyperparameters Selection Explanation: The hyperparameters (alpha_values and
```

l1_ratio_values) were chosen to test different regularisation strengths and Lasso/Ridge mixes. Alpha controls the regularisation strength, and l1_ratio adjusts the balance between Lasso (L1) and Ridge (L2) regularisation. This helps find the best model that fits the data without overfitting.

1.6.3 E.3 Fit Model

[ ]:
```
# <Student to fill this section>
predictions = {}

for alpha, l1_ratio in product(alpha_values, l1_ratio_values):
    model = ElasticNet(alpha=alpha, l1_ratio=l1_ratio, random_state=42)
    model.fit(X_train, y_train)

    y_pred = model.predict(X_val)

    # Store predictions in a dictionary
    key = f"y_pred_{alpha}_{l1_ratio}"
    predictions[key] = y_pred
```

```
/usr/local/lib/python3.11/dist-
packages/sklearn/linear_model/_coordinate_descent.py:695: ConvergenceWarning:
Objective did not converge. You might want to increase the number of iterations,
check the scale of the features or consider increasing regularisation. Duality
gap: 1.616e+05, tolerance: 1.370e+02
  model = cd_fast.enet_coordinate_descent(
/usr/local/lib/python3.11/dist-
packages/sklearn/linear_model/_coordinate_descent.py:695: ConvergenceWarning:
Objective did not converge. You might want to increase the number of iterations,
check the scale of the features or consider increasing regularisation. Duality
gap: 4.627e+04, tolerance: 1.370e+02
  model = cd_fast.enet_coordinate_descent(
/usr/local/lib/python3.11/dist-
packages/sklearn/linear_model/_coordinate_descent.py:695: ConvergenceWarning:
Objective did not converge. You might want to increase the number of iterations,
check the scale of the features or consider increasing regularisation. Duality
gap: 1.140e+05, tolerance: 1.370e+02
  model = cd_fast.enet_coordinate_descent(
/usr/local/lib/python3.11/dist-
packages/sklearn/linear_model/_coordinate_descent.py:695: ConvergenceWarning:
Objective did not converge. You might want to increase the number of iterations,
check the scale of the features or consider increasing regularisation. Duality
gap: 5.006e+04, tolerance: 1.370e+02
  model = cd_fast.enet_coordinate_descent(
/usr/local/lib/python3.11/dist-
packages/sklearn/linear_model/_coordinate_descent.py:695: ConvergenceWarning:
Objective did not converge. You might want to increase the number of iterations,
check the scale of the features or consider increasing regularisation. Duality
```

```
gap: 3.233e+04, tolerance: 1.370e+02
  model = cd_fast.enet_coordinate_descent(
/usr/local/lib/python3.11/dist-
packages/sklearn/linear_model/_coordinate_descent.py:695: ConvergenceWarning:
Objective did not converge. You might want to increase the number of iterations,
check the scale of the features or consider increasing regularisation. Duality
gap: 2.800e+04, tolerance: 1.370e+02
  model = cd_fast.enet_coordinate_descent(
/usr/local/lib/python3.11/dist-
packages/sklearn/linear_model/_coordinate_descent.py:695: ConvergenceWarning:
Objective did not converge. You might want to increase the number of iterations,
check the scale of the features or consider increasing regularisation. Duality
gap: 1.118e+04, tolerance: 1.370e+02
  model = cd_fast.enet_coordinate_descent(
```

1.6.4 E.4 Model Technical Performance

Provide some explanations on model performance

[ ]:
```python
# <Student to fill this section>
for alpha, l1_ratio in product(alpha_values, l1_ratio_values):
    mse = mean_squared_error(y_val, predictions[f"y_pred_{alpha}_{l1_ratio}"])
    rmse = np.sqrt(mse)
    mae = mean_absolute_error(y_val, predictions[f"y_pred_{alpha}_{l1_ratio}"])
    r2 = r2_score(y_val, predictions[f"y_pred_{alpha}_{l1_ratio}"])

    print(f"alpha = {alpha}, l1_ratio = {l1_ratio}")
    print(f"RMSE: {rmse}")
    print(f"MAE: {mae}")
    print(f"R2: {r2}")
    print(".............................")
```

```
alpha = 0.0002, l1_ratio = 0.1
RMSE: 12.00623640511735
MAE: 8.775610206228876
R2: 0.7212619276649992
…
alpha = 0.0002, l1_ratio = 0.5
RMSE: 12.006153181288784
MAE: 8.77556513009934
R2: 0.7212657919182696
…
alpha = 0.0002, l1_ratio = 0.9
RMSE: 12.006070769026262
MAE: 8.775520263713384
R2: 0.7212696184624665
…
alpha = 0.0002, l1_ratio = 1
```

```
RMSE: 12.006051079652368
MAE: 8.775510113617853
R2: 0.7212705326703374
…
alpha = 0.001, l1_ratio = 0.1
RMSE: 12.007274413443403
MAE: 8.776046335125061
R2: 0.7212137285562465
…
alpha = 0.001, l1_ratio = 0.5
RMSE: 12.00684379549469
MAE: 8.775753601969546
R2: 0.7212337244713087
…
alpha = 0.001, l1_ratio = 0.9
RMSE: 12.006421091679147
MAE: 8.775476903380452
R2: 0.721253352192982
…
alpha = 0.001, l1_ratio = 1
RMSE: 12.00631751309932
MAE: 8.775411327962143
R2: 0.7212581616290582
…
alpha = 0.005, l1_ratio = 0.1
RMSE: 12.01261900174742
MAE: 8.778430612566817
R2: 0.7209654907959158
…
alpha = 0.005, l1_ratio = 0.5
RMSE: 12.010433616767614
MAE: 8.776978309006418
R2: 0.721067007768478
…
alpha = 0.005, l1_ratio = 0.9
RMSE: 12.008239147777434
MAE: 8.775472416111823
R2: 0.7211689281322119
…
alpha = 0.005, l1_ratio = 1
RMSE: 12.00774344630357
MAE: 8.774932833067295
R2: 0.721191948013608
…
alpha = 0.1, l1_ratio = 0.1
RMSE: 12.179592243515604
MAE: 8.862084308875033
R2: 0.7131545210352108
```

```
…
alpha = 0.1, l1_ratio = 0.5
RMSE: 12.115076880418767
MAE: 8.818244989095778
R2: 0.7161853166199124
…
alpha = 0.1, l1_ratio = 0.9
RMSE: 12.058735728402713
MAE: 8.77940592932157
R2: 0.71881893828021
…
alpha = 0.1, l1_ratio = 1
RMSE: 12.046949939020541
MAE: 8.768458951789883
R2: 0.7193683028870802
…
alpha = 1, l1_ratio = 0.1
RMSE: 14.176226309098928
MAE: 10.412545132466049
R2: 0.6113991137550343
…
alpha = 1, l1_ratio = 0.5
RMSE: 13.609002175937178
MAE: 9.840034404409803
R2: 0.64187463791331
…
alpha = 1, l1_ratio = 0.9
RMSE: 12.876012528397593
MAE: 9.153433833566641
R2: 0.6794134539211959
…
alpha = 1, l1_ratio = 1
RMSE: 12.648616862433004
MAE: 8.982487123708001
R2: 0.6906368461171115
…
```

[ ]:
```
# @title Model Performance Explanation

wgt_model_performance_explanation = widgets.Textarea(
    value="The model's performance improved with alpha = 0.0002 and l1_ratio =␣
 ↪1, achieving the lowest RMSE of 12. This suggests that using a smaller alpha␣
 ↪value allows the model to better fit the data while avoiding overfitting.␣
 ↪The l1_ratio of 1 indicates a pure Lasso regularisation, which helps with␣
 ↪feature selection by driving less important feature coefficients to zero.␣
 ↪This combination leads to a more efficient model, minimising prediction␣
 ↪error.",
```

```
    placeholder='<student to fill this section>',
    description='Model Performance Explanation:',
    disabled=False,
    style={'description_width': 'initial'},
    layout=widgets.Layout(height="100%", width="auto")
)
wgt_model_performance_explanation
```

```
Textarea(value="The model's performance improved with alpha = 0.0002 and␣
 ↪l1_ratio = 1, achieving the lowest RM…
```

[ ]:
```
print("Model Performance Explanation:", wgt_model_performance_explanation.value)
```

```
Model Performance Explanation: The model's performance improved with alpha =
0.0002 and l1_ratio = 1, achieving the lowest RMSE of 12. This suggests that
using a smaller alpha value allows the model to better fit the data while
avoiding overfitting. The l1_ratio of 1 indicates a pure Lasso regularisation,
which helps with feature selection by driving less important feature
coefficients to zero. This combination leads to a more efficient model,
minimising prediction error.
```

1.6.5 E.5 Business Impact from Current Model Performance

Provide some analysis on the model impacts from the business point of view

[ ]:
```
# <Student to fill this section>

y_1 = y_train['rent']
y_pred = predictions[f"y_pred_{0.0002}_{1}"]
y_pred = pd.DataFrame(y_pred, columns=['rent_pred'])
y_2 = y_pred['rent_pred']

train_set = alt.Chart(pd.DataFrame({'target': y_1, 'preds': y_1})).
 ↪mark_line(color='green').encode(
    x='target',
    y='preds'
)
test_set = alt.Chart(pd.DataFrame({'target': y_1, 'preds': y_2})).mark_line().
 ↪encode(
    x='target',
    y='preds'
  )

test_set + train_set
```

[ ]:
```
alt.LayerChart(…)
```

[ ]:
```
model_final = ElasticNet(alpha=0.0002, l1_ratio=1, random_state=42)
model_final.fit(X_train, y_train)

y_pred_final = model_final.predict(X_test)

mse_final = mean_squared_error(y_test, y_pred_final)
rmse_final = np.sqrt(mse_final)
mae_final = mean_absolute_error(y_test, y_pred_final)
r2_final = r2_score(y_test, y_pred_final)

print(f"RMSE: {rmse_final}")
print(f"MAE: {mae_final}")
print(f"R2: {r2_final}")
```

```
RMSE: 22.64418234518295
MAE: 14.689283726992722
R2: 0.6113670147990962
```

[ ]:
```
# @title Model Business Impacts Explanation

wgt_model_business_explanation = widgets.Textarea(
    value="The best combination of hyperparameters is alpha = 0.0002 and␣
 ↪l1_ratio = 1. However, on unseen data, the RMSE increases to 22.64, and the␣
 ↪R² drops to 0.61, suggesting that while the model performs well on the␣
 ↪training set, there is still room for improvement in generalisation to␣
 ↪unseen data. This combination of alpha and l1_ratio provides a balanced␣
 ↪trade-off between minimising prediction error and capturing the underlying␣
 ↪data patterns. Despite this, the result is quite similar to the performance␣
 ↪of the Multivariate Regression model, indicating that further optimisation␣
 ↪may be needed.",
    placeholder='<student to fill this section>',
    description='Model Business Impacts Explanation:',
    disabled=False,
    style={'description_width': 'initial'},
    layout=widgets.Layout(height="100%", width="auto")
)
wgt_model_business_explanation
```

```
Textarea(value='The best combination of hyperparameters is alpha = 0.0002 and␣
 ↪l1_ratio = 1. However, on unseen…
```

[ ]:
```
print("Model Business Impacts Explanation:", wgt_model_business_explanation.
 ↪value)
```

```
Model Business Impacts Explanation: The best combination of hyperparameters is
alpha = 0.0002 and l1_ratio = 1. However, on unseen data, the RMSE increases to
22.64, and the R² drops to 0.61, suggesting that while the model performs well
on the training set, there is still room for improvement in generalisation to
```

unseen data. This combination of alpha and l1_ratio provides a balanced trade-off between minimising prediction error and capturing the underlying data patterns. Despite this, the result is quite similar to the performance of the Multivariate Regression model, indicating that further optimisation may be needed.

1.7 F. Experiment Outcomes

[ ]:
```
# @title Experiment Outcomes Explanation

wgt_experiment_outcomes_explanation = widgets.Select(
    options=['Hypothesis Confirmed', 'Hypothesis Partially Confirmed',␣
 ↪'Hypothesis Rejected'],
    value='Hypothesis Rejected',
    description='Experiment Outcomes:',
    disabled=False,
)

wgt_experiment_outcomes_explanation
```

```
Select(description='Experiment Outcomes:', index=2, options=('Hypothesis␣
 ↪Confirmed', 'Hypothesis Partially Con…
```

[30]:
```
# @title Experiments Results Explanation

wgt_experiment_results_explanation = widgets.Textarea(
    value="Hypothesis Rejected. In this experiment, various combinations of␣
 ↪alpha and l1_ratio were tested to optimise the ElasticNet model's␣
 ↪performance. The best results were achieved with alpha = 0.0002 and l1_ratio␣
 ↪= 1, yielding the lowest RMSE of 12 and the highest R² of 0.72 on the␣
 ↪training set. While the model demonstrates reasonable performance on the␣
 ↪training set, there is still room for further improvement, possibly through␣
 ↪additional optimisation in data cleaning and feature engineering.",
    placeholder='<student to fill this section>',
    description='Experiments Results Explanation:',
    disabled=False,
    style={'description_width': 'initial'},
    layout=widgets.Layout(height="100%", width="auto")
)
wgt_experiment_results_explanation
```

```
Textarea(value="Hypothesis Rejected. In this experiment, various combinations of␣
 ↪alpha and l1_ratio were teste…
```

[31]:
```
print("Experiments Results Explanation:", wgt_experiment_results_explanation.
 ↪value)
```

```
Experiments Results Explanation: Hypothesis Rejected. In this experiment,
```

various combinations of alpha and l1_ratio were tested to optimise the ElasticNet model's performance. The best results were achieved with alpha = 0.0002 and l1_ratio = 1, yielding the lowest RMSE of 12 and the highest R^2 of 0.72 on the training set. While the model demonstrates reasonable performance on the training set, there is still room for further improvement, possibly through additional optimisation in data cleaning and feature engineering.