final project

December 11, 2024

[1]:
```
import pandas as pd

# Load the dataset
file_path = 'california_housing_data.csv'
housing_data = pd.read_csv(file_path)

# Display the first few rows and dataset information
housing_data_info = housing_data.info()
housing_data_head = housing_data.head()

housing_data_info, housing_data_head
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 20640 entries, 0 to 20639
Data columns (total 9 columns):
 #   Column       Non-Null Count  Dtype  
---  ------       --------------  -----  
 0   MedInc       20640 non-null  float64
 1   HouseAge     20640 non-null  float64
 2   AveRooms     20640 non-null  float64
 3   AveBedrms    20640 non-null  float64
 4   Population   20640 non-null  float64
 5   AveOccup     20640 non-null  float64
 6   Latitude     20640 non-null  float64
 7   Longitude    20640 non-null  float64
 8   MedHouseVal  20640 non-null  float64
dtypes: float64(9)
memory usage: 1.4 MB
```

[1]:
```
(None,
      MedInc  HouseAge  AveRooms  AveBedrms  Population  AveOccup  Latitude  \
 0  8.3252       41.0  6.984127   1.023810       322.0  2.555556     37.88   
 1  8.3014       21.0  6.238137   0.971880      2401.0  2.109842     37.86   
 2  7.2574       52.0  8.288136   1.073446       496.0  2.802260     37.85   
 3  5.6431       52.0  5.817352   1.073059       558.0  2.547945     37.85   
 4  3.8462       52.0  6.281853   1.081081       565.0  2.181467     37.85   

    Longitude  MedHouseVal
```

```
0    -122.23        4.526
1    -122.22        3.585
2    -122.24        3.521
3    -122.25        3.413
4    -122.25        3.422  )
```

[2]:
```
from sklearn.model_selection import train_test_split

# Define the features and target variable
X = housing_data.drop(columns=["MedHouseVal"])
y = housing_data["MedHouseVal"]

# Split the dataset into training and testing sets (80-20 split)
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,␣
 ↪random_state=42)

# Verify the shapes of the splits
X_train.shape, X_test.shape, y_train.shape, y_test.shape
```

[2]:
```
((16512, 8), (4128, 8), (16512,), (4128,))
```

[3]:
```
import mlflow
import mlflow.sklearn
from sklearn.ensemble import RandomForestRegressor
from sklearn.metrics import mean_squared_error
from math import sqrt
from flaml import AutoML
```

[4]:
```
# Step 1: Define RMSE as the evaluation metric
def calculate_rmse(y_true, y_pred):
    return sqrt(mean_squared_error(y_true, y_pred))

# Step 2: Set up MLflow experiment
mlflow.set_experiment("California Housing AutoML")

# Step 3: Train a model using FLAML's AutoML
automl = AutoML()
automl_settings = {
    "time_budget": 60,  # 1-minute budget for AutoML
    "metric": "rmse",
    "task": "regression",
    "log_file_name": "automl.log",
}

# Start an MLflow run
with mlflow.start_run():
    # Train the model
```

```
    automl.fit(X_train=X_train, y_train=y_train, **automl_settings)

    # Get the best model
    best_model = automl.model
    mlflow.sklearn.log_model(best_model, "best_model")

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

    # Calculate RMSE
    rmse = calculate_rmse(y_test, y_pred)
    mlflow.log_metric("RMSE", rmse)

# Output RMSE and best model details
best_model_details = {
    "Best Algorithm": automl.best_estimator,
    "Best Configuration": automl.best_config,
    "Best RMSE": rmse,
}
best_model_details
```

```
2024/12/11 01:01:55 WARNING mlflow.utils.git_utils: Failed to import Git (the
Git executable is probably not on your PATH), so Git SHA is not available.
Error: Failed to initialize: Bad git executable.
The git executable must be specified in one of the following ways:
    - be included in your $PATH
    - be set via $GIT_PYTHON_GIT_EXECUTABLE
    - explicitly set via git.refresh(<full-path-to-git-executable>)

All git commands will error until this is rectified.

This initial message can be silenced or aggravated in the future by setting the
$GIT_PYTHON_REFRESH environment variable. Use one of the following values:
    - quiet|q|silence|s|silent|none|n|0: for no message or exception
    - warn|w|warning|log|l|1: for a warning message (logging level CRITICAL,
displayed by default)
    - error|e|exception|raise|r|2: for a raised exception

Example:
    export GIT_PYTHON_REFRESH=quiet


[flaml.automl.logger: 12-11 01:01:55] {1728} INFO - task = regression
[flaml.automl.logger: 12-11 01:01:55] {1739} INFO - Evaluation method: cv
[flaml.automl.logger: 12-11 01:01:55] {1838} INFO - Minimizing error metric:
rmse
[flaml.automl.logger: 12-11 01:01:55] {1955} INFO - List of ML learners in
AutoML Run: ['lgbm', 'rf', 'xgboost', 'extra_tree', 'xgb_limitdepth', 'sgd']
```

```
[flaml.automl.logger: 12-11 01:01:55] {2258} INFO - iteration 0, current learner
lgbm
[flaml.automl.logger: 12-11 01:01:56] {2393} INFO - Estimated sufficient time
budget=7060s. Estimated necessary time budget=50s.
[flaml.automl.logger: 12-11 01:01:56] {2442} INFO -  at 1.0s,   estimator lgbm's
best error=0.9899,     best estimator lgbm's best error=0.9899
[flaml.automl.logger: 12-11 01:01:56] {2258} INFO - iteration 1, current learner
lgbm
[flaml.automl.logger: 12-11 01:01:57] {2442} INFO -  at 1.8s,   estimator lgbm's
best error=0.9899,     best estimator lgbm's best error=0.9899
[flaml.automl.logger: 12-11 01:01:57] {2258} INFO - iteration 2, current learner
lgbm
[flaml.automl.logger: 12-11 01:01:57] {2442} INFO -  at 2.6s,   estimator lgbm's
best error=0.8398,     best estimator lgbm's best error=0.8398
[flaml.automl.logger: 12-11 01:01:57] {2258} INFO - iteration 3, current learner
sgd
[flaml.automl.logger: 12-11 01:01:58] {2442} INFO -  at 2.9s,   estimator sgd's
best error=1.1889,      best estimator lgbm's best error=0.8398
[flaml.automl.logger: 12-11 01:01:58] {2258} INFO - iteration 4, current learner
lgbm
[flaml.automl.logger: 12-11 01:01:59] {2442} INFO -  at 4.7s,   estimator lgbm's
best error=0.6142,     best estimator lgbm's best error=0.6142
[flaml.automl.logger: 12-11 01:01:59] {2258} INFO - iteration 5, current learner
sgd
[flaml.automl.logger: 12-11 01:02:00] {2442} INFO -  at 4.9s,   estimator sgd's
best error=1.1866,      best estimator lgbm's best error=0.6142
[flaml.automl.logger: 12-11 01:02:00] {2258} INFO - iteration 6, current learner
xgboost
[flaml.automl.logger: 12-11 01:02:02] {2442} INFO -  at 7.1s,   estimator
xgboost's best error=0.9893,  best estimator lgbm's best error=0.6142
[flaml.automl.logger: 12-11 01:02:02] {2258} INFO - iteration 7, current learner
extra_tree
[flaml.automl.logger: 12-11 01:02:02] {2442} INFO -  at 7.7s,   estimator
extra_tree's best error=0.8916,       best estimator lgbm's best error=0.6142
[flaml.automl.logger: 12-11 01:02:02] {2258} INFO - iteration 8, current learner
lgbm
[flaml.automl.logger: 12-11 01:02:03] {2442} INFO -  at 8.5s,   estimator lgbm's
best error=0.6142,     best estimator lgbm's best error=0.6142
[flaml.automl.logger: 12-11 01:02:03] {2258} INFO - iteration 9, current learner
lgbm
[flaml.automl.logger: 12-11 01:02:05] {2442} INFO -  at 10.6s,  estimator lgbm's
best error=0.5907,     best estimator lgbm's best error=0.5907
[flaml.automl.logger: 12-11 01:02:05] {2258} INFO - iteration 10, current
learner xgboost
[flaml.automl.logger: 12-11 01:02:06] {2442} INFO -  at 11.5s,  estimator
xgboost's best error=0.9893,  best estimator lgbm's best error=0.5907
[flaml.automl.logger: 12-11 01:02:06] {2258} INFO - iteration 11, current
learner extra_tree
```

```
[flaml.automl.logger: 12-11 01:02:07] {2442} INFO -  at 12.3s,  estimator
extra_tree's best error=0.7583,       best estimator lgbm's best error=0.5907
[flaml.automl.logger: 12-11 01:02:07] {2258} INFO - iteration 12, current
learner rf
[flaml.automl.logger: 12-11 01:02:08] {2442} INFO -  at 13.0s,  estimator rf's
best error=0.8363,       best estimator lgbm's best error=0.5907
[flaml.automl.logger: 12-11 01:02:08] {2258} INFO - iteration 13, current
learner rf
[flaml.automl.logger: 12-11 01:02:09] {2442} INFO -  at 13.9s,  estimator rf's
best error=0.7055,       best estimator lgbm's best error=0.5907
[flaml.automl.logger: 12-11 01:02:09] {2258} INFO - iteration 14, current
learner xgboost
[flaml.automl.logger: 12-11 01:02:10] {2442} INFO -  at 14.9s,  estimator
xgboost's best error=0.8470,  best estimator lgbm's best error=0.5907
[flaml.automl.logger: 12-11 01:02:10] {2258} INFO - iteration 15, current
learner rf
[flaml.automl.logger: 12-11 01:02:11] {2442} INFO -  at 15.9s,  estimator rf's
best error=0.7055,       best estimator lgbm's best error=0.5907
[flaml.automl.logger: 12-11 01:02:11] {2258} INFO - iteration 16, current
learner extra_tree
[flaml.automl.logger: 12-11 01:02:11] {2442} INFO -  at 16.4s,  estimator
extra_tree's best error=0.7583,       best estimator lgbm's best error=0.5907
[flaml.automl.logger: 12-11 01:02:11] {2258} INFO - iteration 17, current
learner lgbm
[flaml.automl.logger: 12-11 01:02:13] {2442} INFO -  at 18.3s,  estimator lgbm's
best error=0.5907,     best estimator lgbm's best error=0.5907
[flaml.automl.logger: 12-11 01:02:13] {2258} INFO - iteration 18, current
learner extra_tree
[flaml.automl.logger: 12-11 01:02:14] {2442} INFO -  at 19.0s,  estimator
extra_tree's best error=0.7064,       best estimator lgbm's best error=0.5907
[flaml.automl.logger: 12-11 01:02:14] {2258} INFO - iteration 19, current
learner lgbm
[flaml.automl.logger: 12-11 01:02:15] {2442} INFO -  at 19.8s,  estimator lgbm's
best error=0.5907,     best estimator lgbm's best error=0.5907
[flaml.automl.logger: 12-11 01:02:15] {2258} INFO - iteration 20, current
learner xgboost
[flaml.automl.logger: 12-11 01:02:15] {2442} INFO -  at 20.6s,  estimator
xgboost's best error=0.7014,  best estimator lgbm's best error=0.5907
[flaml.automl.logger: 12-11 01:02:15] {2258} INFO - iteration 21, current
learner xgboost
[flaml.automl.logger: 12-11 01:02:16] {2442} INFO -  at 21.5s,  estimator
xgboost's best error=0.7014,  best estimator lgbm's best error=0.5907
[flaml.automl.logger: 12-11 01:02:16] {2258} INFO - iteration 22, current
learner xgboost
[flaml.automl.logger: 12-11 01:02:17] {2442} INFO -  at 22.3s,  estimator
xgboost's best error=0.7014,  best estimator lgbm's best error=0.5907
[flaml.automl.logger: 12-11 01:02:17] {2258} INFO - iteration 23, current
learner lgbm
```

```
[flaml.automl.logger: 12-11 01:02:24] {2442} INFO -  at 28.9s,  estimator lgbm's
best error=0.5455,     best estimator lgbm's best error=0.5455
[flaml.automl.logger: 12-11 01:02:24] {2258} INFO - iteration 24, current
learner rf
[flaml.automl.logger: 12-11 01:02:25] {2442} INFO -  at 29.9s,  estimator rf's
best error=0.6327,       best estimator lgbm's best error=0.5455
[flaml.automl.logger: 12-11 01:02:25] {2258} INFO - iteration 25, current
learner extra_tree
[flaml.automl.logger: 12-11 01:02:25] {2442} INFO -  at 30.7s,  estimator
extra_tree's best error=0.6303,       best estimator lgbm's best error=0.5455
[flaml.automl.logger: 12-11 01:02:25] {2258} INFO - iteration 26, current
learner extra_tree
[flaml.automl.logger: 12-11 01:02:26] {2442} INFO -  at 31.5s,  estimator
extra_tree's best error=0.6303,       best estimator lgbm's best error=0.5455
[flaml.automl.logger: 12-11 01:02:26] {2258} INFO - iteration 27, current
learner rf
[flaml.automl.logger: 12-11 01:02:27] {2442} INFO -  at 32.4s,  estimator rf's
best error=0.5824,       best estimator lgbm's best error=0.5455
[flaml.automl.logger: 12-11 01:02:27] {2258} INFO - iteration 28, current
learner xgboost
[flaml.automl.logger: 12-11 01:02:29] {2442} INFO -  at 34.3s,  estimator
xgboost's best error=0.6082,  best estimator lgbm's best error=0.5455
[flaml.automl.logger: 12-11 01:02:29] {2258} INFO - iteration 29, current
learner extra_tree
[flaml.automl.logger: 12-11 01:02:30] {2442} INFO -  at 35.0s,  estimator
extra_tree's best error=0.5768,       best estimator lgbm's best error=0.5455
[flaml.automl.logger: 12-11 01:02:30] {2258} INFO - iteration 30, current
learner extra_tree
[flaml.automl.logger: 12-11 01:02:30] {2442} INFO -  at 35.5s,  estimator
extra_tree's best error=0.5768,       best estimator lgbm's best error=0.5455
[flaml.automl.logger: 12-11 01:02:30] {2258} INFO - iteration 31, current
learner rf
[flaml.automl.logger: 12-11 01:02:31] {2442} INFO -  at 36.7s,  estimator rf's
best error=0.5824,       best estimator lgbm's best error=0.5455
[flaml.automl.logger: 12-11 01:02:31] {2258} INFO - iteration 32, current
learner lgbm
[flaml.automl.logger: 12-11 01:02:34] {2442} INFO -  at 38.9s,  estimator lgbm's
best error=0.5455,     best estimator lgbm's best error=0.5455
[flaml.automl.logger: 12-11 01:02:34] {2258} INFO - iteration 33, current
learner rf
[flaml.automl.logger: 12-11 01:02:35] {2442} INFO -  at 40.0s,  estimator rf's
best error=0.5608,       best estimator lgbm's best error=0.5455
[flaml.automl.logger: 12-11 01:02:35] {2258} INFO - iteration 34, current
learner extra_tree
[flaml.automl.logger: 12-11 01:02:36] {2442} INFO -  at 40.9s,  estimator
extra_tree's best error=0.5603,       best estimator lgbm's best error=0.5455
[flaml.automl.logger: 12-11 01:02:36] {2258} INFO - iteration 35, current
learner rf
```

```
[flaml.automl.logger: 12-11 01:02:37] {2442} INFO -  at 41.9s,  estimator rf's
best error=0.5608,       best estimator lgbm's best error=0.5455
[flaml.automl.logger: 12-11 01:02:37] {2258} INFO - iteration 36, current
learner xgboost
[flaml.automl.logger: 12-11 01:02:40] {2442} INFO -  at 45.0s,  estimator
xgboost's best error=0.5754,  best estimator lgbm's best error=0.5455
[flaml.automl.logger: 12-11 01:02:40] {2258} INFO - iteration 37, current
learner extra_tree
[flaml.automl.logger: 12-11 01:02:40] {2442} INFO -  at 45.7s,  estimator
extra_tree's best error=0.5603,       best estimator lgbm's best error=0.5455
[flaml.automl.logger: 12-11 01:02:40] {2258} INFO - iteration 38, current
learner xgboost
[flaml.automl.logger: 12-11 01:02:42] {2442} INFO -  at 47.6s,  estimator
xgboost's best error=0.5754,  best estimator lgbm's best error=0.5455
[flaml.automl.logger: 12-11 01:02:42] {2258} INFO - iteration 39, current
learner extra_tree
[flaml.automl.logger: 12-11 01:02:43] {2442} INFO -  at 48.3s,  estimator
extra_tree's best error=0.5503,       best estimator lgbm's best error=0.5455
[flaml.automl.logger: 12-11 01:02:43] {2258} INFO - iteration 40, current
learner xgboost
[flaml.automl.logger: 12-11 01:02:46] {2442} INFO -  at 51.4s,  estimator
xgboost's best error=0.5395,  best estimator xgboost's best error=0.5395
[flaml.automl.logger: 12-11 01:02:46] {2258} INFO - iteration 41, current
learner extra_tree
[flaml.automl.logger: 12-11 01:02:47] {2442} INFO -  at 52.1s,  estimator
extra_tree's best error=0.5503,       best estimator xgboost's best error=0.5395
[flaml.automl.logger: 12-11 01:02:47] {2258} INFO - iteration 42, current
learner extra_tree
[flaml.automl.logger: 12-11 01:02:48] {2442} INFO -  at 53.0s,  estimator
extra_tree's best error=0.5114,       best estimator extra_tree's best
error=0.5114
[flaml.automl.logger: 12-11 01:02:48] {2258} INFO - iteration 43, current
learner lgbm
[flaml.automl.logger: 12-11 01:02:55] {2442} INFO -  at 60.1s,  estimator lgbm's
best error=0.4995,     best estimator lgbm's best error=0.4995
[flaml.automl.logger: 12-11 01:02:55] {521} INFO - logging best model lgbm
[flaml.automl.logger: 12-11 01:02:59] {2685} INFO - retrain lgbm for 4.2s
[flaml.automl.logger: 12-11 01:02:59] {2688} INFO - retrained model:
LGBMRegressor(colsample_bytree=0.6649148062238498,
              learning_rate=0.17402065726724145, max_bin=255,
              min_child_samples=3, n_estimators=146, n_jobs=-1, num_leaves=18,
              reg_alpha=0.0009765625, reg_lambda=0.006761362450996487,
              verbose=-1)
[flaml.automl.logger: 12-11 01:02:59] {2690} INFO - Best MLflow run name:
bright-squid-72_child_43
[flaml.automl.logger: 12-11 01:02:59] {2691} INFO - Best MLflow run id:
f4d61ef193694ebc9c14b5831b332239
[flaml.automl.logger: 12-11 01:03:14] {1985} INFO - fit succeeded
```

```
[flaml.automl.logger: 12-11 01:03:14] {1986} INFO - Time taken to find the best
model: 60.06856393814087

2024/12/11 01:03:19 WARNING mlflow.models.model: Model logged without a
signature and input example. Please set `input_example` parameter when logging
the model to auto infer the model signature.
```

[4]:
```
{'Best Algorithm': 'lgbm',
 'Best Configuration': {'n_estimators': 146,
  'num_leaves': 18,
  'min_child_samples': 3,
  'learning_rate': 0.17402065726724145,
  'log_max_bin': 8,
  'colsample_bytree': 0.6649148062238498,
  'reg_alpha': 0.0009765625,
  'reg_lambda': 0.006761362450996487},
 'Best RMSE': 0.4671886754678566}
```

[6]:
```
# Redefine and reload necessary components
from lightgbm import LGBMRegressor
import joblib
import os

# Reload the best model parameters (from earlier results)
best_model = LGBMRegressor(
    n_estimators=146,
    num_leaves=18,
    min_child_samples=3,
    learning_rate=0.17402065726724145,
    max_bin=2**8,
    colsample_bytree=0.6649148062238498,
    reg_alpha=0.0009765625,
    reg_lambda=0.006761362450996487,
)
# Fit the best model on the full training data
best_model.fit(X_train, y_train)

# Save the trained model
model_dir = "/mnt/data/deployed_model"
os.makedirs(model_dir, exist_ok=True)
model_path = os.path.join(model_dir, "lightgbm_best_model.pkl")
joblib.dump(best_model, model_path)

# Verify the model is saved
os.path.exists(model_path)
```

[6]:
```
True
```

[7]:
```
# Load the saved model
loaded_model = joblib.load(model_path)

# Step 1: Predict on the original test dataset
y_pred_original = loaded_model.predict(X_test)

# Calculate RMSE for the original test dataset
rmse_original = calculate_rmse(y_test, y_pred_original)

# Step 2: Alter the test dataset (swap two features and randomize another)
X_test_altered = X_test.copy()
X_test_altered["AveRooms"], X_test_altered["AveBedrms"] =␣
↪X_test_altered["AveBedrms"], X_test_altered["AveRooms"]
X_test_altered["Population"] = X_test_altered["Population"].sample(frac=1).
↪values  # Random shuffle

# Predict on the altered test dataset
y_pred_altered = loaded_model.predict(X_test_altered)

# Calculate RMSE for the altered test dataset
rmse_altered = calculate_rmse(y_test, y_pred_altered)

# Return RMSE for both cases
rmse_results = {
    "RMSE (Original Test Data)": rmse_original,
    "RMSE (Altered Test Data)": rmse_altered,
}
rmse_results
```

[7]:
```
{'RMSE (Original Test Data)': 0.4674563512832409,
 'RMSE (Altered Test Data)': 0.663975312564932}
```

[8]:
```
def make_prediction(input_data):
    """
    Simulate a deployed model's prediction.
    :param input_data: DataFrame or dictionary of features.
    :return: Model predictions.
    """
    prediction = loaded_model.predict(input_data)
    return prediction

# Example: Predict for a sample from the test dataset
sample_data = X_test.iloc[:1]
predicted_value = make_prediction(sample_data)
print(f"Predicted Value: {predicted_value}")
```

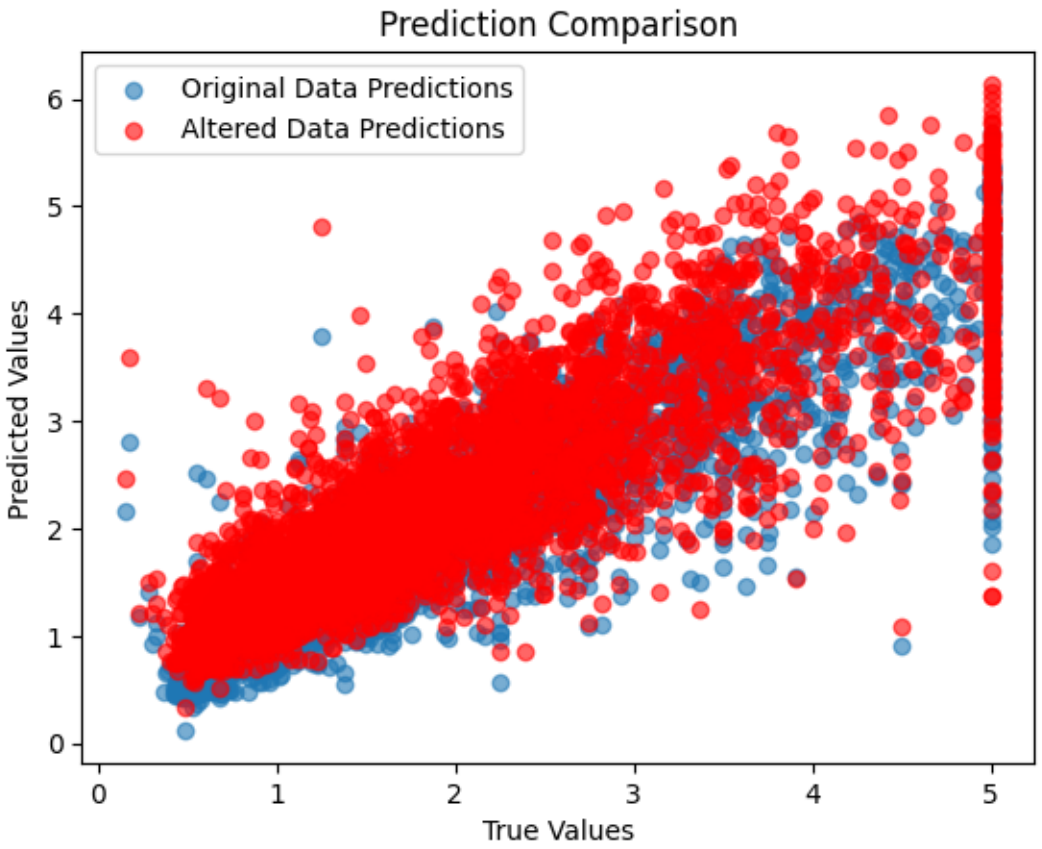
```
Predicted Value: [0.59670358]
```

```
C:\Users\16251\AppData\Local\Programs\Python\Python39\lib\site-
packages\lightgbm\basic.py:722: UserWarning: Usage of np.ndarray subset (sliced
data) is not recommended due to it will double the peak memory cost in LightGBM.
  _log_warning(
```

[9]:
```
import matplotlib.pyplot as plt

# Log predictions for original and altered data
original_predictions = loaded_model.predict(X_test)
altered_predictions = loaded_model.predict(X_test_altered)

# Compare predictions
plt.figure()
plt.scatter(y_test, original_predictions, label="Original Data Predictions",␣
 ↪alpha=0.6)
plt.scatter(y_test, altered_predictions, label="Altered Data Predictions",␣
 ↪alpha=0.6, color='red')
plt.xlabel("True Values")
plt.ylabel("Predicted Values")
plt.legend()
plt.title("Prediction Comparison")
plt.show()
```



[10]:
```
# Simulate data drift by altering feature distributions
X_test_drifted = X_test.copy()
X_test_drifted["MedInc"] *= 1.2  # Increase median income by 20%

# Predict on drifted data
drifted_predictions = loaded_model.predict(X_test_drifted)

# Compare drifted predictions with original
plt.figure()
plt.hist(original_predictions, alpha=0.5, label="Original")
plt.hist(drifted_predictions, alpha=0.5, label="Drifted", color="orange")
plt.legend()
plt.title("Prediction Distribution: Original vs Drifted")
plt.show()
```