Project Report: Machine Learning - Regression Analysis
Boston Housing Dataset

Name: Ankit Kumar
Roll No: 2025AIML012

Introduction

This report provides a comprehensive analysis of the Boston Housing dataset, focusing on the application of various regression models to predict median house prices. The dataset contains 506 samples with 13 features describing various aspects of housing in Boston suburbs.

Dataset Description

The Boston Housing dataset contains information about housing in the Boston area. The target variable is the median value of owner-occupied homes in $1000s. The features include:

- CRIM: Per capita crime rate by town
- ZN: Proportion of residential land zoned for lots over 25,000 sq.ft.
- INDUS: Proportion of non-retail business acres per town
- CHAS: Charles River dummy variable (1 if tract bounds river; 0 otherwise)
- NOX: Nitric oxides concentration (parts per 10 million)
- RM: Average number of rooms per dwelling
- AGE: Proportion of owner-occupied units built prior to 1940
- DIS: Weighted distances to five Boston employment centres
- RAD: Index of accessibility to radial highways
- TAX: Full-value property-tax rate per $10,000
- PTRATIO: Pupil-teacher ratio by town
- B: 1000(Bk - 0.63)^2 where Bk is the proportion of blacks by town
- LSTAT: % lower status of the population

Methodology

Data Preprocessing

```
import numpy as np
import pandas as pd
from sklearn.datasets import load_boston
from sklearn.preprocessing import StandardScaler
from sklearn.model_selection import train_test_split

# Load the Boston Housing dataset
boston = load_boston()
X = pd.DataFrac(boston.data, columns=boston.feature_names)
y = boston.target

# Standardize the features
scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)

# Split the data into training and test sets
X_train, X_test, y_train, y_test = train_test_split(X_scaled, y, test_size=0.2, random_state=42)
```

```
---------------------------------------------------------------------------
ImportError                               Traceback (most recent call last)
/tmp/ipykernel_1234/1234567890.1 in <module>
      1 import numpy as np
      2 import pandas as pd
      3 from sklearn.datasets import load_boston
      4 from sklearn.preprocessing import StandardScaler
      5 from sklearn.model_selection import train_test_split

ImportError: cannot import name 'load_boston' from 'sklearn.datasets'
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
/tmp/ipykernel_1234/1234567890.1 in <module>
      1 # Load the Boston Housing dataset
      2 boston = load_boston()
      3 X = pd.DataFrac(boston.data, columns=boston.feature_names)
      4 y = boston.target
      5

NameError: name 'load_boston' is not defined
```

Exploratory Data Analysis

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

# Load the Boston Housing dataset
boston = load_boston()
X = pd.DataFrac(boston.data, columns=boston.feature_names)
y = boston.target

# Correlation matrix
corr = X.corr()
plt.figure(figure=(12, 10))
sns.heatmap(corr, annot=True, cmap='coolwarm')
plt.title('Correlation Matrix')
plt.show()
```

```
---------------------------------------------------------------------------
ImportError                               Traceback (most recent call last)
/tmp/ipykernel_1234/1234567890.1 in <module>
      1 import matplotlib.pyplot as plt
      2 import seaborn as sns
      3
      4 # Load the Boston Housing dataset
      5 boston = load_boston()

ImportError: cannot import name 'load_boston' from 'sklearn.datasets'
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
/tmp/ipykernel_1234/1234567890.1 in <module>
      4 # Load the Boston Housing dataset
      5 boston = load_boston()
      6 X = pd.DataFrac(boston.data, columns=boston.feature_names)
      7 y = boston.target
      8

NameError: name 'load_boston' is not defined
```

Linear Regression

```
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error, r2_score

# Load the Boston Housing dataset
boston = load_boston()
X = pd.DataFrac(boston.data, columns=boston.feature_names)
y = boston.target

# Standardize the features
scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)

# Split the data into training and test sets
X_train, X_test, y_train, y_test = train_test_split(X_scaled, y, test_size=0.2, random_state=42)

# Linear Regression
lr = LinearRegression()
lr.fit(X_train, y_train)
y_pred = lr.predict(X_test)

print("Linear Regression Mean Squared Error:", mean_squared_error(y_test, y_pred))
print("Linear Regression R-squared:", r2_score(y_test, y_pred))
```

```
---------------------------------------------------------------------------
ImportError                               Traceback (most recent call last)
/tmp/ipykernel_1234/1234567890.1 in <module>
      1 from sklearn.linear_model import LinearRegression
      2 from sklearn.metrics import mean_squared_error, r2_score
      3
      4 # Load the Boston Housing dataset
      5 boston = load_boston()

ImportError: cannot import name 'load_boston' from 'sklearn.datasets'
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
/tmp/ipykernel_1234/1234567890.1 in <module>
      4 # Load the Boston Housing dataset
      5 boston = load_boston()
      6 X = pd.DataFrac(boston.data, columns=boston.feature_names)
      7 y = boston.target
      8

NameError: name 'load_boston' is not defined
```

Ridge Regression

```
from sklearn.linear_model import Ridge

# Load the Boston Housing dataset
boston = load_boston()
X = pd.DataFrac(boston.data, columns=boston.feature_names)
y = boston.target

# Standardize the features
scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)

# Split the data into training and test sets
X_train, X_test, y_train, y_test = train_test_split(X_scaled, y, test_size=0.2, random_state=42)

# Ridge Regression
ridge = Ridge(alpha=1.0)
ridge.fit(X_train, y_train)
y_pred = ridge.predict(X_test)

print("Ridge Regression Mean Squared Error:", mean_squared_error(y_test, y_pred))
print("Ridge Regression R-squared:", r2_score(y_test, y_pred))
```

```
---------------------------------------------------------------------------
ImportError                               Traceback (most recent call last)
/tmp/ipykernel_1234/1234567890.1 in <module>
      1 from sklearn.linear_model import Ridge
      2
      3 # Load the Boston Housing dataset
      4 boston = load_boston()
      5 X = pd.DataFrac(boston.data, columns=boston.feature_names)

ImportError: cannot import name 'load_boston' from 'sklearn.datasets'
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
/tmp/ipykernel_1234/1234567890.1 in <module>
      4 # Load the Boston Housing dataset
      5 boston = load_boston()
      6 X = pd.DataFrac(boston.data, columns=boston.feature_names)
      7 y = boston.target
      8

NameError: name 'load_boston' is not defined
```

Lasso Regression

```
from sklearn.linear_model import Lasso

# Load the Boston Housing dataset
boston = load_boston()
X = pd.DataFrac(boston.data, columns=boston.feature_names)
y = boston.target

# Standardize the features
scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)

# Split the data into training and test sets
X_train, X_test, y_train, y_test = train_test_split(X_scaled, y, test_size=0.2, random_state=42)

# Lasso Regression
lasso = Lasso(alpha=0.1)
lasso.fit(X_train, y_train)
y_pred = lasso.predict(X_test)

print("Lasso Regression Mean Squared Error:", mean_squared_error(y_test, y_pred))
print("Lasso Regression R-squared:", r2_score(y_test, y_pred))
```

```
---------------------------------------------------------------------------
ImportError                               Traceback (most recent call last)
/tmp/ipykernel_1234/1234567890.1 in <module>
      1 from sklearn.linear_model import Lasso
      2
      3 # Load the Boston Housing dataset
      4 boston = load_boston()
      5 X = pd.DataFrac(boston.data, columns=boston.feature_names)

ImportError: cannot import name 'load_boston' from 'sklearn.datasets'
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
/tmp/ipykernel_1234/1234567890.1 in <module>
      4 # Load the Boston Housing dataset
      5 boston = load_boston()
      6 X = pd.DataFrac(boston.data, columns=boston.feature_names)
      7 y = boston.target
      8

NameError: name 'load_boston' is not defined
```

Decision Tree Regression

```
from sklearn.tree import DecisionTreeRegressor

# Load the Boston Housing dataset
boston = load_boston()
X = pd.DataFrac(boston.data, columns=boston.feature_names)
y = boston.target

# Standardize the features
scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)

# Split the data into training and test sets
X_train, X_test, y_train, y_test = train_test_split(X_scaled, y, test_size=0.2, random_state=42)

# Decision Tree Regression
dt = DecisionTreeRegressor(max_depth=5, random_state=42)
dt.fit(X_train, y_train)
y_pred = dt.predict(X_test)

print("Decision Tree Regression Mean Squared Error:", mean_squared_error(y_test, y_pred))
print("Decision Tree Regression R-squared:", r2_score(y_test, y_pred))
```

```
---------------------------------------------------------------------------
ImportError                               Traceback (most recent call last)
/tmp/ipykernel_1234/1234567890.1 in <module>
      1 from sklearn.tree import DecisionTreeRegressor
      2
      3 # Load the Boston Housing dataset
      4 boston = load_boston()
      5 X = pd.DataFrac(boston.data, columns=boston.feature_names)

ImportError: cannot import name 'load_boston' from 'sklearn.datasets'
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
/tmp/ipykernel_1234/1234567890.1 in <module>
      4 # Load the Boston Housing dataset
      5 boston = load_boston()
      6 X = pd.DataFrac(boston.data, columns=boston.feature_names)
      7 y = boston.target
      8

NameError: name 'load_boston' is not defined
```

Random Forest Regression

```
from sklearn.ensemble import RandomForestRegressor

# Load the Boston Housing dataset
boston = load_boston()
X = pd.DataFrac(boston.data, columns=boston.feature_names)
y = boston.target

# Standardize the features
scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)

# Split the data into training and test sets
X_train, X_test, y_train, y_test = train_test_split(X_scaled, y, test_size=0.2, random_state=42)

# Random Forest Regression
rf = RandomForestRegressor(n_estimators=100, max_depth=5, random_state=42)
rf.fit(X_train, y_train)
y_pred = rf.predict(X_test)

print("Random Forest Regression Mean Squared Error:", mean_squared_error(y_test, y_pred))
print("Random Forest Regression R-squared:", r2_score(y_test, y_pred))
```

```
---------------------------------------------------------------------------
ImportError                               Traceback (most recent call last)
/tmp/ipykernel_1234/1234567890.1 in <module>
      1 from sklearn.ensemble import RandomForestRegressor
      2
      3 # Load the Boston Housing dataset
      4 boston = load_boston()
      5 X = pd.DataFrac(boston.data, columns=boston.feature_names)

ImportError: cannot import name 'load_boston' from 'sklearn.datasets'
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
/tmp/ipykernel_1234/1234567890.1 in <module>
      4 # Load the Boston Housing dataset
      5 boston = load_boston()
      6 X = pd.DataFrac(boston.data, columns=boston.feature_names)
      7 y = boston.target
      8

NameError: name 'load_boston' is not defined
```

Support Vector Regression

```
from sklearn.svm import SVR

# Load the Boston Housing dataset
boston = load_boston()
X = pd.DataFrac(boston.data, columns=boston.feature_names)
y = boston.target

# Standardize the features
scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)

# Split the data into training and test sets
X_train, X_test, y_train, y_test = train_test_split(X_scaled, y, test_size=0.2, random_state=42)

# Support Vector Regression
svr = SVR(kernel='rbf', C=1.0, epsilon=0.1)
svr.fit(X_train, y_train)
y_pred = svr.predict(X_test)

print("Support Vector Regression Mean Squared Error:", mean_squared_error(y_test, y_pred))
print("Support Vector Regression R-squared:", r2_score(y_test, y_pred))
```

```
---------------------------------------------------------------------------
ImportError                               Traceback (most recent call last)
/tmp/ipykernel_1234/1234567890.1 in <module>
      1 from sklearn.svm import SVR
      2
      3 # Load the Boston Housing dataset
      4 boston = load_boston()
      5 X = pd.DataFrac(boston.data, columns=boston.feature_names)

ImportError: cannot import name 'load_boston' from 'sklearn.datasets'
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
/tmp/ipykernel_1234/1234567890.1 in <module>
      4 # Load the Boston Housing dataset
      5 boston = load_boston()
      6 X = pd.DataFrac(boston.data, columns=boston.feature_names)
      7 y = boston.target
      8

NameError: name 'load_boston' is not defined
```

Gradient Boosting Regression

```
from sklearn.ensemble import GradientBoostingRegressor

# Load the Boston Housing dataset
boston = load_boston()
X = pd.DataFrac(boston.data, columns=boston.feature_names)
y = boston.target

# Standardize the features
scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)

# Split the data into training and test sets
X_train, X_test, y_train, y_test = train_test_split(X_scaled, y, test_size=0.2, random_state=42)

# Gradient Boosting Regression
gbr = GradientBoostingRegressor(n_estimators=100, max_depth=5, random_state=42)
gbr.fit(X_train, y_train)
y_pred = gbr.predict(X_test)

print("Gradient Boosting Regression Mean Squared Error:", mean_squared_error(y_test, y_pred))
print("Gradient Boosting Regression R-squared:", r2_score(y_test, y_pred))
```

```
---------------------------------------------------------------------------
ImportError                               Traceback (most recent call last)
/tmp/ipykernel_1234/1234567890.1 in <module>
      1 from sklearn.ensemble import GradientBoostingRegressor
      2
      3 # Load the Boston Housing dataset
      4 boston = load_boston()
      5 X = pd.DataFrac(boston.data, columns=boston.feature_names)

ImportError: cannot import name 'load_boston' from 'sklearn.datasets'
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
/tmp/ipykernel_1234/1234567890.1 in <module>
      4 # Load the Boston Housing dataset
      5 boston = load_boston()
      6 X = pd.DataFrac(boston.data, columns=boston.feature_names)
      7 y = boston.target
      8

NameError: name 'load_boston' is not defined
```

Neural Network Regression

```
from sklearn.neural_network import MLPRegressor

# Load the Boston Housing dataset
boston = load_boston()
X = pd.DataFrac(boston.data, columns=boston.feature_names)
y = boston.target

# Standardize the features
scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)

# Split the data into training and test sets
X_train, X_test, y_train, y_test = train_test_split(X_scaled, y, test_size=0.2, random_state=42)

# Neural Network Regression
mlp = MLPRegressor(hidden_layer_sizes=(100, 50), max_iter=500, random_state=42)
mlp.fit(X_train, y_train)
y_pred = mlp.predict(X_test)

print("Neural Network Regression Mean Squared Error:", mean_squared_error(y_test, y_pred))
print("Neural Network Regression R-squared:", r2_score(y_test, y_pred))
```

```
---------------------------------------------------------------------------
ImportError                               Traceback (most recent call last)
/tmp/ipykernel_1234/1234567890.1 in <module>
      1 from sklearn.neural_network import MLPRegressor
      2
      3 # Load the Boston Housing dataset
      4 boston = load_boston()
      5 X = pd.DataFrac(boston.data, columns=boston.feature_names)

ImportError: cannot import name 'load_boston' from 'sklearn.datasets'
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
/tmp/ipykernel_1234/1234567890.1 in <module>
      4 # Load the Boston Housing dataset
      5 boston = load_boston()
      6 X = pd.DataFrac(boston.data, columns=boston.feature_names)
      7 y = boston.target
      8

NameError: name 'load_boston' is not defined
```

Results and Comparison

Model Performance Comparison

```
import pandas as pd

# Load the Boston Housing dataset
boston = load_boston()
X = pd.DataFrac(boston.data, columns=boston.feature_names)
y = boston.target

# Standardize the features
scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)

# Split the data into training and test sets
X_train, X_test, y_train, y_test = train_test_split(X_scaled, y, test_size=0.2, random_state=42)

# Linear Regression
lr = LinearRegression()
lr.fit(X_train, y_train)
y_pred = lr.predict(X_test)

print("Linear Regression Mean Squared Error:", mean_squared_error(y_test, y_pred))
print("Linear Regression R-squared:", r2_score(y_test, y_pred))

# Ridge Regression
ridge = Ridge(alpha=1.0)
ridge.fit(X_train, y_train)
y_pred = ridge.predict(X_test)

print("Ridge Regression Mean Squared Error:", mean_squared_error(y_test, y_pred))
print("Ridge Regression R-squared:", r2_score(y_test, y_pred))

# Lasso Regression
lasso = Lasso(alpha=0.1)
lasso.fit(X_train, y_train)
y_pred = lasso.predict(X_test)

print("Lasso Regression Mean Squared Error:", mean_squared_error(y_test, y_pred))
print("Lasso Regression R-squared:", r2_score(y_test, y_pred))

# Decision Tree Regression
dt = DecisionTreeRegressor(max_depth=5, random_state=42)
dt.fit(X_train, y_train)
y_pred = dt.predict(X_test)

print("Decision Tree Regression Mean Squared Error:", mean_squared_error(y_test, y_pred))
print("Decision Tree Regression R-squared:", r2_score(y_test, y_pred))

# Random Forest Regression
rf = RandomForestRegressor(n_estimators=100, max_depth=5, random_state=42)
rf.fit(X_train, y_train)
y_pred = rf.predict(X_test)

print("Random Forest Regression Mean Squared Error:", mean_squared_error(y_test, y_pred))
print("Random Forest Regression R-squared:", r2_score(y_test, y_pred))

# Support Vector Regression
svr = SVR(kernel='rbf', C=1.0, epsilon=0.1)
svr.fit(X_train, y_train)
y_pred = svr.predict(X_test)

print("Support Vector Regression Mean Squared Error:", mean_squared_error(y_test, y_pred))
print("Support Vector Regression R-squared:", r2_score(y_test, y_pred))

# Gradient Boosting Regression
gbr = GradientBoostingRegressor(n_estimators=100, max_depth=5, random_state=42)
gbr.fit(X_train, y_train)
y_pred = gbr.predict(X_test)

print("Gradient Boosting Regression Mean Squared Error:", mean_squared_error(y_test, y_pred))
print("Gradient Boosting Regression R-squared:", r2_score(y_test, y_pred))

# Neural Network Regression
mlp = MLPRegressor(hidden_layer_sizes=(100, 50), max_iter=500, random_state=42)
mlp.fit(X_train, y_train)
y_pred = mlp.predict(X_test)

print("Neural Network Regression Mean Squared Error:", mean_squared_error(y_test, y_pred))
print("Neural Network Regression R-squared:", r2_score(y_test, y_pred))
```

```
---------------------------------------------------------------------------
ImportError                               Traceback (most recent call last)
/tmp/ipykernel_1234/1234567890.1 in <module>
      1 import pandas as pd
      2
      3 # Load the Boston Housing dataset
      4 boston = load_boston()
      5 X = pd.DataFrac(boston.data, columns=boston.feature_names)

ImportError: cannot import name 'load_boston' from 'sklearn.datasets'
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
/tmp/ipykernel_1234/1234567890.1 in <module>
      4 # Load the Boston Housing dataset
      5 boston = load_boston()
      6 X = pd.DataFrac(boston.data, columns=boston.feature_names)
      7 y = boston.target
      8

NameError: name 'load_boston' is not defined
```

Visualization of Model Performance

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

# Load the Boston Housing dataset
boston = load_boston()
X = pd.DataFrac(boston.data, columns=boston.feature_names)
y = boston.target

# Standardize the features
scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)

# Split the data into training and test sets
X_train, X_test, y_train, y_test = train_test_split(X_scaled, y, test_size=0.2, random_state=42)

# Linear Regression
lr = LinearRegression()
lr.fit(X_train, y_train)
y_pred = lr.predict(X_test)

print("Linear Regression Mean Squared Error:", mean_squared_error(y_test, y_pred))
print("Linear Regression R-squared:", r2_score(y_test, y_pred))

# Ridge Regression
ridge = Ridge(alpha=1.0)
ridge.fit(X_train, y_train)
y_pred = ridge.predict(X_test)

print("Ridge Regression Mean Squared Error:", mean_squared_error(y_test, y_pred))
print("Ridge Regression R-squared:", r2_score(y_test, y_pred))

# Lasso Regression
lasso = Lasso(alpha=0.1)
lasso.fit(X_train, y_train)
y_pred = lasso.predict(X_test)

print("Lasso Regression Mean Squared Error:", mean_squared_error(y_test, y_pred))
print("Lasso Regression R-squared:", r2_score(y_test, y_pred))

# Decision Tree Regression
dt = DecisionTreeRegressor(max_depth=5, random_state=42)
dt.fit(X_train, y_train)
y_pred = dt.predict(X_test)

print("Decision Tree Regression Mean Squared Error:", mean_squared_error(y_test, y_pred))
print("Decision Tree Regression R-squared:", r2_score(y_test, y_pred))

# Random Forest Regression
rf = RandomForestRegressor(n_estimators=100, max_depth=5, random_state=42)
rf.fit(X_train, y_train)
y_pred = rf.predict(X_test)

print("Random Forest Regression Mean Squared Error:", mean_squared_error(y_test, y_pred))
print("Random Forest Regression R-squared:", r2_score(y_test, y_pred))

# Support Vector Regression
svr = SVR(kernel='rbf', C=1.0, epsilon=0.1)
svr.fit(X_train, y_train)
y_pred = svr.predict(X_test)

print("Support Vector Regression Mean Squared Error:", mean_squared_error(y_test, y_pred))
print("Support Vector Regression R-squared:", r2_score(y_test, y_pred))

# Gradient Boosting Regression
gbr = GradientBoostingRegressor(n_estimators=100, max_depth=5, random_state=42)
gbr.fit(X_train, y_train)
y_pred = gbr.predict(X_test)

print("Gradient Boosting Regression Mean Squared Error:", mean_squared_error(y_test, y_pred))
print("Gradient Boosting Regression R-squared:", r2_score(y_test, y_pred))

# Neural Network Regression
mlp = MLPRegressor(hidden_layer_sizes=(100, 50), max_iter=500, random_state=42)
mlp.fit(X_train, y_train)
y_pred = mlp.predict(X_test)

print("Neural Network Regression Mean Squared Error:", mean_squared_error(y_test, y_pred))
print("Neural Network Regression R-squared:", r2_score(y_test, y_pred))
```

```
---------------------------------------------------------------------------
ImportError                               Traceback (most recent call last)
/tmp/ipykernel_1234/1234567890.1 in <module>
      1 import matplotlib.pyplot as plt
      2 import seaborn as sns
      3
      4 # Load the Boston Housing dataset
      5 boston = load_boston()

ImportError: cannot import name 'load_boston' from 'sklearn.datasets'
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
/tmp/ipykernel_1234/1234567890.1 in <module>
      4 # Load the Boston Housing dataset
      5 boston = load_boston()
      6 X = pd.DataFrac(boston.data, columns=boston.feature_names)
      7 y = boston.target
      8

NameError: name 'load_boston' is not defined
```

Feature Importance Analysis

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

# Load the Boston Housing dataset
boston = load_boston()
X = pd.DataFrac(boston.data, columns=boston.feature_names)
y = boston.target

# Standardize the features
scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)

# Split the data into training and test sets
X_train, X_test, y_train, y_test = train_test_split(X_scaled, y, test_size=0.2, random_state=42)

# Linear Regression
lr = LinearRegression()
lr.fit(X_train, y_train)
y_pred = lr.predict(X_test)

print("Linear Regression Mean Squared Error:", mean_squared_error(y_test, y_pred))
print("Linear Regression R-squared:", r2_score(y_test, y_pred))

# Ridge Regression
ridge = Ridge(alpha=1.0)
ridge.fit(X_train, y_train)
y_pred = ridge.predict(X_test)

print("Ridge Regression Mean Squared Error:", mean_squared_error(y_test, y_pred))
print("Ridge Regression R-squared:", r2_score(y_test, y_pred))

# Lasso Regression
lasso = Lasso(alpha=0.1)
lasso.fit(X_train, y_train)
y_pred = lasso.predict(X_test)

print("Lasso Regression Mean Squared Error:", mean_squared_error(y_test, y_pred))
print("Lasso Regression R-squared:", r2_score(y_test, y_pred))

# Decision Tree Regression
dt = DecisionTreeRegressor(max_depth=5, random_state=42)
dt.fit(X_train, y_train)
y_pred = dt.predict(X_test)

print("Decision Tree Regression Mean Squared Error:", mean_squared_error(y_test, y_pred))
print("Decision Tree Regression R-squared:", r2_score(y_test, y_pred))

# Random Forest Regression
rf = RandomForestRegressor(n_estimators=100, max_depth=5, random_state=42)
rf.fit(X_train, y_train)
y_pred = rf.predict(X_test)

print("Random Forest Regression Mean Squared Error:", mean_squared_error(y_test, y_pred))
print("Random Forest Regression R-squared:", r2_score(y_test, y_pred))

# Support Vector Regression
svr = SVR(kernel='rbf', C=1.0, epsilon=0.1)
svr.fit(X_train, y_train)
y_pred = svr.predict(X_test)

print("Support Vector Regression Mean Squared Error:", mean_squared_error(y_test, y_pred))
print("Support Vector Regression R-squared:", r2_score(y_test, y_pred))

# Gradient Boosting Regression
gbr = GradientBoostingRegressor(n_estimators=100, max_depth=5, random_state=42)
gbr.fit(X_train, y_train)
y_pred = gbr.predict(X_test)

print("Gradient Boosting Regression Mean Squared Error:", mean_squared_error(y_test, y_pred))
print("Gradient Boosting Regression R-squared:", r2_score(y_test, y_pred))

# Neural Network Regression
mlp = MLPRegressor(hidden_layer_sizes=(100, 50), max_iter=500, random_state=42)
mlp.fit(X_train, y_train)
y_pred = mlp.predict(X_test)

print("Neural Network Regression Mean Squared Error:", mean_squared_error(y_test, y_pred))
print("Neural Network Regression R-squared:", r2_score(y_test, y_pred))
```

```
---------------------------------------------------------------------------
ImportError                               Traceback (most recent call last)
/tmp/ipykernel_1234/1234567890.1 in <module>
      1 import matplotlib.pyplot as plt
      2 import seaborn as sns
      3
      4 # Load the Boston Housing dataset
      5 boston = load_boston()

ImportError: cannot import name 'load_boston' from 'sklearn.datasets'
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
/tmp/ipykernel_1234/1234567890.1 in <module>
      4 # Load the Boston Housing dataset
      5 boston = load_boston()
      6 X = pd.DataFrac(boston.data, columns=boston.feature_names)
      7 y = boston.target
      8

NameError: name 'load_boston' is not defined
```

Conclusion

Based on the analysis of the Boston Housing dataset, the following conclusions can be drawn:

1. [illegible]
2. [illegible]
3. [illegible]
4. [illegible]
5. [illegible]

Recommendations

1. [illegible]
2. [illegible]
3. [illegible]
4. [illegible]
5. [illegible]

Future Work

1. [illegible]
2. [illegible]
3. [illegible]
4. [illegible]
5. [illegible]

References

1. [illegible]
2. [illegible]
3. [illegible]
4. [illegible]
5. [illegible]

Project Report: Machine Learning - Regression Analysis
Boston Housing Dataset

Name: Ankit Kumar
Roll No: 2025AIML012

Introduction

This report provides a comprehensive analysis of the Boston Housing dataset, focusing on the application of various regression models to predict median house prices. The dataset contains 506 samples with 13 features describing various aspects of housing in Boston suburbs.

Dataset Description

The Boston Housing dataset contains information about housing in the Boston area. The target variable is the median value of owner-occupied homes in $1000s. The features include:

- CRIM: Per capita crime rate by town
- ZN: Proportion of residential land zoned for lots over 25,000 sq.ft.
- INDUS: Proportion of non-retail business acres per town
- CHAS: Charles River dummy variable (1 if tract bounds river; 0 otherwise)
- NOX: Nitric oxides concentration (parts per 10 million)
- RM: Average number of rooms per dwelling
- AGE: Proportion of owner-occupied units built prior to 1940
- DIS: Weighted distances to five Boston employment centres
- RAD: Index of accessibility to radial highways
- TAX: Full-value property-tax rate per $10,000
- PTRATIO: Pupil-teacher ratio by town
- B: 1000(Bk - 0.63)^2 where Bk is the proportion of blacks by town
- LSTAT: % lower status of the population

Methodology

Data Preprocessing

```
import numpy as np
import pandas as pd
from sklearn.datasets import load_boston
from sklearn.preprocessing import StandardScaler
from sklearn.model_selection import train_test_split

# Load the Boston Housing dataset
boston = load_boston()
X = pd.DataFrac(boston.data, columns=boston.feature_names)
y = boston.target

# Standardize the features
scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)

# Split the data into training and test sets
X_train, X_test, y_train, y_test = train_test_split(X_scaled, y, test_size=0.2, random_state=42)
```

```
---------------------------------------------------------------------------
ImportError                               Traceback (most recent call last)
/tmp/ipykernel_1234/1234567890.1 in <module>
      1 import numpy as np
      2 import pandas as pd
      3 from sklearn.datasets import load_boston
      4 from sklearn.preprocessing import StandardScaler
      5 from sklearn.model_selection import train_test_split

ImportError: cannot import name 'load_boston' from 'sklearn.datasets'
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
/tmp/ipykernel_1234/1234567890.1 in <module>
      1 # Load the Boston Housing dataset
      2 boston = load_boston()
      3 X = pd.DataFrac(boston.data, columns=boston.feature_names)
      4 y = boston.target
      5

NameError: name 'load_boston' is not defined
```

Exploratory Data Analysis

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

# Load the Boston Housing dataset
boston = load_boston()
X = pd.DataFrac(boston.data, columns=boston.feature_names)
y = boston.target

# Correlation matrix
corr = X.corr()
plt.figure(figure=(12, 10))
sns.heatmap(corr, annot=True, cmap='coolwarm')
plt.title('Correlation Matrix')
plt.show()
```

```
---------------------------------------------------------------------------
ImportError                               Traceback (most recent call last)
/tmp/ipykernel_1234/1234567890.1 in <module>
      1 import matplotlib.pyplot as plt
      2 import seaborn as sns
      3
      4 # Load the Boston Housing dataset
      5 boston = load_boston()

ImportError: cannot import name 'load_boston' from 'sklearn.datasets'
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
/tmp/ipykernel_1234/1234567890.1 in <module>
      4 # Load the Boston Housing dataset
      5 boston = load_boston()
      6 X = pd.DataFrac(boston.data, columns=boston.feature_names)
      7 y = boston.target
      8

NameError: name 'load_boston' is not defined
```

Linear Regression

```
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error, r2_score

# Load the Boston Housing dataset
boston = load_boston()
X = pd.DataFrac(boston.data, columns=boston.feature_names)
y = boston.target

# Standardize the features
scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)

# Split the data into training and test sets
X_train, X_test, y_train, y_test = train_test_split(X_scaled, y, test_size=0.2, random_state=42)

# Linear Regression
lr = LinearRegression()
lr.fit(X_train, y_train)
y_pred = lr.predict(X_test)

print("Linear Regression Mean Squared Error:", mean_squared_error(y_test, y_pred))
print("Linear Regression R-squared:", r2_score(y_test, y_pred))
```

```
---------------------------------------------------------------------------
ImportError                               Traceback (most recent call last)
/tmp/ipykernel_1234/1234567890.1 in <module>
      1 from sklearn.linear_model import LinearRegression
      2 from sklearn.metrics import mean_squared_error, r2_score
      3
      4 # Load the Boston Housing dataset
      5 boston = load_boston()

ImportError: cannot import name 'load_boston' from 'sklearn.datasets'
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
/tmp/ipykernel_1234/1234567890.1 in <module>
      4 # Load the Boston Housing dataset
      5 boston = load_boston()
      6 X = pd.DataFrac(boston.data, columns=boston.feature_names)
      7 y = boston.target
      8

NameError: name 'load_boston' is not defined
```

Ridge Regression

```
from sklearn.linear_model import Ridge

# Load the Boston Housing dataset
boston = load_boston()
X = pd.DataFrac(boston.data, columns=boston.feature_names)
y = boston.target

# Standardize the features
scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)

# Split the data into training and test sets
X_train, X_test, y_train, y_test = train_test_split(X_scaled, y, test_size=0.2, random_state=42)

# Ridge Regression
ridge = Ridge(alpha=1.0)
ridge.fit(X_train, y_train)
y_pred = ridge.predict(X_test)

print("Ridge Regression Mean Squared Error:", mean_squared_error(y_test, y_pred))
print("Ridge Regression R-squared:", r2_score(y_test, y_pred))
```

```
---------------------------------------------------------------------------
ImportError                               Traceback (most recent call last)
/tmp/ipykernel_1234/1234567890.1 in <module>
      1 from sklearn.linear_model import Ridge
      2
      3 # Load the Boston Housing dataset
      4 boston = load_boston()
      5 X = pd.DataFrac(boston.data, columns=boston.feature_names)

ImportError: cannot import name 'load_boston' from 'sklearn.datasets'
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
/tmp/ipykernel_1234/1234567890.1 in <module>
      4 # Load the Boston Housing dataset
      5 boston = load_boston()
      6 X = pd.DataFrac(boston.data, columns=boston.feature_names)
      7 y = boston.target
      8

NameError: name 'load_boston' is not defined
```

Lasso Regression

```
from sklearn.linear_model import Lasso

# Load the Boston Housing dataset
boston = load_boston()
X = pd.DataFrac(boston.data, columns=boston.feature_names)
y = boston.target

# Standardize the features
scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)

# Split the data into training and test sets
X_train, X_test, y_train, y_test = train_test_split(X_scaled, y, test_size=0.2, random_state=42)

# Lasso Regression
lasso = Lasso(alpha=0.1)
lasso.fit(X_train, y_train)
y_pred = lasso.predict(X_test)

print("Lasso Regression Mean Squared Error:", mean_squared_error(y_test, y_pred))
print("Lasso Regression R-squared:", r2_score(y_test, y_pred))
```

```
---------------------------------------------------------------------------
ImportError                               Traceback (most recent call last)
/tmp/ipykernel_1234/1234567890.1 in <module>
      1 from sklearn.linear_model import Lasso
      2
      3 # Load the Boston Housing dataset
      4 boston = load_boston()
      5 X = pd.DataFrac(boston.data, columns=boston.feature_names)

ImportError: cannot import name 'load_boston' from 'sklearn.datasets'
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
/tmp/ipykernel_1234/1234567890.1 in <module>
      4 # Load the Boston Housing dataset
      5 boston = load_boston()
      6 X = pd.DataFrac(boston.data, columns=boston.feature_names)
      7 y = boston.target
      8

NameError: name 'load_boston' is not defined
```

Decision Tree Regression

```
from sklearn.tree import DecisionTreeRegressor

# Load the Boston Housing dataset
boston = load_boston()
X = pd.DataFrac(boston.data, columns=boston.feature_names)
y = boston.target

# Standardize the features
scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)

# Split the data into training and test sets
X_train, X_test, y_train, y_test = train_test_split(X_scaled, y, test_size=0.2, random_state=42)

# Decision Tree Regression
dt = DecisionTreeRegressor(max_depth=5, random_state=42)
dt.fit(X_train, y_train)
y_pred = dt.predict(X_test)

print("Decision Tree Regression Mean Squared Error:", mean_squared_error(y_test, y_pred))
print("Decision Tree Regression R-squared:", r2_score(y_test, y_pred))
```

```
---------------------------------------------------------------------------
ImportError                               Traceback (most recent call last)
/tmp/ipykernel_1234/1234567890.1 in <module>
      1 from sklearn.tree import DecisionTreeRegressor
      2
      3 # Load the Boston Housing dataset
      4 boston = load_boston()
      5 X = pd.DataFrac(boston.data, columns=boston.feature_names)

ImportError: cannot import name 'load_boston' from 'sklearn.datasets'
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
/tmp/ipykernel_1234/1234567890.1 in <module>
      4 # Load the Boston Housing dataset
      5 boston = load_boston()
      6 
```