```python
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression, Lasso, Ridge
from sklearn.metrics import mean_squared_error
from sklearn.metrics import mean_absolute_error
```

#load data

```python
file_path = "California_Houses.csv"
data = pd.read_csv(file_path)
```

#Split the balanced dataset into training and the rest

```python
training_set, remaining_data = train_test_split(data, test_size=0.3,
random_state=42)
```

#split the remaining into validation and testing sets

#validation set should be reserved strictly for evaluating

#the model's performance after you've trained it using the training set and adjusted the hyperparameters separately.

```python
validation_set, test_set = train_test_split(
    remaining_data, test_size=0.5, random_state=42
)
```

#Define features (X) and target variable (y)

```python
x_train = training_set.drop(columns=["Median_House_Value"])
y_train = training_set["Median_House_Value"]

x_validation = validation_set.drop(columns=["Median_House_Value"])
y_validation = validation_set["Median_House_Value"]

x_test = test_set.drop(columns=["Median_House_Value"])
y_test = test_set["Median_House_Value"]
```

#Apply Linear Regression

```python
linear_model = LinearRegression()
linear_model.fit(x_train, y_train)
```

```
LinearRegression()
```

#Evaluate Linear Regression on validation set(Calculate Mean Square Error for Linear Regression)

```
linear_val_predictions = linear_model.predict(x_validation)
linear_val_mse = mean_squared_error(y_validation,
linear_val_predictions)
print("*****************")
print("Linear Regression Validation MSE:", linear_val_mse)
```

```
*****************
Linear Regression Validation MSE: 4907211997.374699
```

#Calculate Mean Absolute Error for Linear Regression

```
linear_val_mae = mean_absolute_error(y_validation,
linear_val_predictions)
print("Linear Regression Validation MAE:", linear_val_mae)
print("*****************")
```

```
Linear Regression Validation MAE: 50790.06027104908
*****************
```

#Apply Lasso Regression

```
lasso_model = Lasso(alpha=0.1, max_iter=10000)  # You can adjust the
alpha parameter
lasso_model.fit(x_train, y_train)
```

```
Lasso(alpha=0.1, max_iter=10000)
```

#Evaluate Lasso Regression on validation set

```
lasso_val_predictions = lasso_model.predict(x_validation)
lasso_val_mse = mean_squared_error(y_validation,
lasso_val_predictions)
print("*****************")
print("Lasso Regression Validation MSE:", lasso_val_mse)
```

```
*****************
Lasso Regression Validation MSE: 4907212766.117535
```

#Calculate Mean Absolute Error for Lasso Regression

```
lasso_val_mae = mean_absolute_error(y_validation,
lasso_val_predictions)
print("Lasso Regression Validation MAE:", lasso_val_mae)
print("*****************")
```

```
Lasso Regression Validation MAE: 50790.08159127155
*****************
```

#Apply Ridge Regression

```
ridge_model = Ridge(alpha=1.0)
ridge_model.fit(x_train, y_train)

Ridge()
```

#Evaluate Ridge Regression on validation set

```
ridge_val_predictions = ridge_model.predict(x_validation)
ridge_val_mse = mean_squared_error(y_validation,
ridge_val_predictions)
print("Ridge regression validation MSE:", ridge_val_mse)

Ridge regression validation MSE: 4907226928.247799
```

#Calculate Mean Absolute Error for Ridge Regression

```
ridge_val_predictions = ridge_model.predict(x_validation)
ridge_val_mae = mean_absolute_error(y_validation,
ridge_val_predictions)
print("Ridge regression validation MAE:", ridge_val_mae)
print("-----------------------------------------------------")

Ridge regression validation MAE: 50790.60731450408
-----------------------------------------------------
```

#Calculate Mean Squared Error for Linear Regression on test set

```
linear_test_predictions = linear_model.predict(x_test)
linear_test_mse = mean_squared_error(y_test, linear_test_predictions)
print("Linear regression test MSE:", linear_test_mse)

Linear regression test MSE: 4400953150.613167
```

#Calculate Mean Absolute Error for Linear Regression on test set

```
linear_test_mae = mean_absolute_error(y_test, linear_test_predictions)
print("Linear regression test MAE:", linear_test_mae)
print("-----------------------------------------------------")

Linear regression test MAE: 48782.03108085429
-----------------------------------------------------
```

#Calculate Mean Square Error for Lasso Regression on test set

```
lasso_test_predictions = lasso_model.predict(x_test)
lasso_test_mse = mean_squared_error(y_test, lasso_test_predictions)

print("Lasso regression test MSE:", lasso_test_mse)
```

```
Lasso regression test MSE: 4400953898.419449
```

#Calculate Mean Absolute Error for Lasso Regression on test set

```
lasso_test_mae = mean_absolute_error(y_test, lasso_test_predictions)
print("Lasso regression test MAE:", lasso_test_mae)

Lasso regression test MAE: 48782.04862279251
```

#Calculate Mean Squared Error for Ridge Regression on test set

```
ridge_test_predictions = ridge_model.predict(x_test)
ridge_test_mse = mean_squared_error(y_test, ridge_test_predictions)
print("-----------------------------------------------------")

print("Ridge regression test MSE:", ridge_test_mse)

-----------------------------------------------------
Ridge regression test MSE: 4400963939.986322
```

#Calculate Mean Absolute Error for Ridge Regression on test set

```
ridge_test_mae = mean_absolute_error(y_test, ridge_test_predictions)
print("Ridge regression test MAE:", ridge_test_mae)
print("-----------------------------------------------------")

Ridge regression test MAE: 48782.50871135065
-----------------------------------------------------
```

#Validation Set Evaluation

#Find the best-performing model based on validation set MSE

```
best_model_mse = min(linear_val_mse, min(lasso_val_mse,
ridge_val_mse))
best_model_mae = min(linear_val_mae, min(lasso_val_mae,
ridge_val_mae))

if best_model_mse == linear_val_mse:
    print("Best model based on validation set: Linear regression")
    print("MSE:", best_model_mse)
    print("MAE:", best_model_mae)
elif best_model_mse == lasso_val_mse:
    print("Best model based on validation set: Lasso regression")
    print("MSE:", best_model_mse)
    print("MAE:", best_model_mae)
else:
    print("Best model based on validation set: Ridge regression")
    print("MSE:", best_model_mse)
    print("MAE:", best_model_mae)

----------------------------------------------------------------------
-----
NameError                                 Traceback (most recent call
last)
Cell In[1], line 1
----> 1 if best_model_mse == linear_val_mse:
      2     print("Best model based on validation set: Linear
regression")
      3     print("MSE:", best_model_mse)

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

#Test Set Evaluation

#Evaluate the best-performing model on the test set

```
if best_model_mse == linear_val_mse:
    print("Best model based on test set: Linear regression")
    print("MSE:", linear_test_mse)
    print("MAE:", linear_test_mae)
elif best_model_mse == lasso_val_mse:
    print("Best model based on test set: Lasso regression")
    print("MSE:", lasso_test_mse)
    print("MAE:", lasso_test_mae)
else:  # Ridge Regression
    print("Best model based on test set: Ridge regression")
    print("MSE:", ridge_test_mse)
    print("MAE:", ridge_test_mae)

Best model based on test set: Linear regression
MSE: 4400953150.613167
MAE: 48782.03108085429
```