Cal_Electerycity_bill

December 19, 2025

[86]:
```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import r2_score, mean_absolute_error, mean_squared_error
import numpy as np

# 1 Load your Excel data
data = pd.read_excel(r"C:
 ↪\Users\waeil\OneDrive\Desktop\ML_AI\My_electricity_bill.xlsx")

# Convert to DataFrame (optional, read_excel already returns a DataFrame)
df = pd.DataFrame(data)

# 2 Features and target
# Make sure these columns exist in your Excel file exactly as named
X = df[['BillUsage', 'PrevBalance', 'NewCharges', 'PayAdjAmount',␣
 ↪'BillingDays']]
y = df['CurrentDueAmount']

# 3 Train-test split
X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.33, random_state=42
)

# 4 Train Linear Regression
model = LinearRegression()
model.fit(X_train, y_train)

# 5 Predictions
y_pred = model.predict(X_test)

# 6 Evaluate the model
r2 = r2_score(y_test, y_pred)
mae = mean_absolute_error(y_test, y_pred)
rmse = np.sqrt(mean_squared_error(y_test, y_pred))

print("Predictions:", y_pred)
```

```
print("R² Score:", r2)
print("MAE:", mae)
print("RMSE:", rmse)

# 7 Model coefficients
coefficients = pd.DataFrame({
    "Feature": X.columns,
    "Coefficient": model.coef_
})
print(coefficients)
print("Intercept:", model.intercept_)
```

```
---------------------------------------------------------------------------
KeyError                                  Traceback (most recent call last)
Cell In[86], line 15
     11 df = pd.DataFrame(data)
     13 # 2 Features and target
     14 # Make sure these columns exist in your Excel file exactly as named
---> 15 X =␣
 ↪df[[                ,                 ,                 ,                 ,                 ]]
     16 y = df['CurrentDueAmount']
     18 # 3 Train-test split

File ~\anaconda3\Lib\site-packages\pandas\core\frame.py:4119, in DataFrame.
 ↪__getitem__(self, key)
   4117     if is_iterator(key):
   4118         key = list(key)
-> 4119     indexer = self.columns._get_indexer_strict(key,           )[1]
   4121 # take() does not accept boolean indexers
   4122 if getattr(indexer, "dtype", None) == bool:

File ~\anaconda3\Lib\site-packages\pandas\core\indexes\base.py:6212, in Index.
 ↪_get_indexer_strict(self, key, axis_name)
   6209 else:
   6210     keyarr, indexer, new_indexer = self._reindex_non_unique(keyarr)
-> 6212 self._raise_if_missing(keyarr, indexer, axis_name)
   6214 keyarr = self.take(indexer)
   6215 if isinstance(key, Index):
   6216     # GH 42790 - Preserve name from an Index

File ~\anaconda3\Lib\site-packages\pandas\core\indexes\base.py:6264, in Index.
 ↪_raise_if_missing(self, key, indexer, axis_name)
   6261     raise KeyError(f"None of [{key}] are in the [{axis_name}]")
   6263 not_found = list(ensure_index(key)[missing_mask.nonzero()[0]].unique())
-> 6264 raise KeyError(f"{not_found} not in index")
```

```
KeyError: "['BillingDays'] not in index"
```

[ ]: