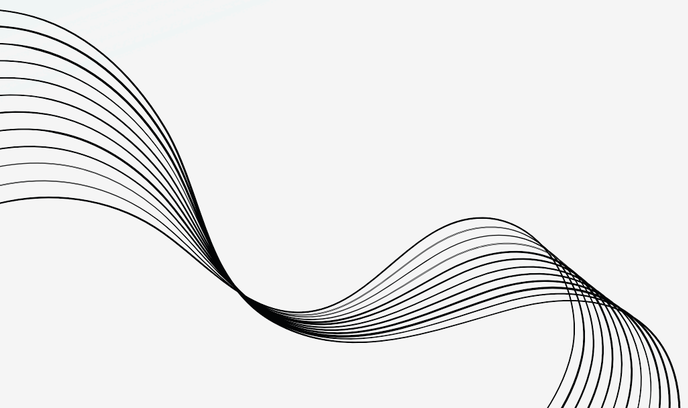
DATA MANAGAMENT PROJECT

Daoud Bourhan
Aliou Tely Diallo
Vincent Schmitz

CONTENT

INTRODUCTION

“ Predict wether a household’s electricity consumption cost will be high or low in the USA”



Objective n° 2

Objective n° 3

Analysing Data

DATA MERGING

```
[186] database1 = pd.read_csv(io.BytesIO(uploaded['database1 .csv']))
      database2 = pd.read_csv(io.BytesIO(uploaded['database2 .csv']))
      database3 = pd.read_csv(io.BytesIO(uploaded['database3 .csv']))
```

```
[191] print(merged_df.columns)
```

```
Index(['doeid', 'HHID', 'ba_climate', 'iecc_climate_code', 'uatyp10', 'hdd65',
       'cdd65', 'hdd30yr_pub', 'cdd30yr_pub', 'typehuq',
       ...
       'EMPLOYHH', 'EDUCATION', 'SDESCENT', 'HOUSEHOLDER_RACE', 'NHSLDMEM',
       'NUMCHILD', 'NUMADULT1', 'NUMADULT2', 'ATHOME', 'MONEYPY'],
      dtype='object', length=293)
```

Detection of the missing values

```
[195] merged_data.isnull().sum().sum()

      #This code checks if we had missing values in merged_data
      #isnull() returns a dataframe with True for NaN values and False for other values. %%
```

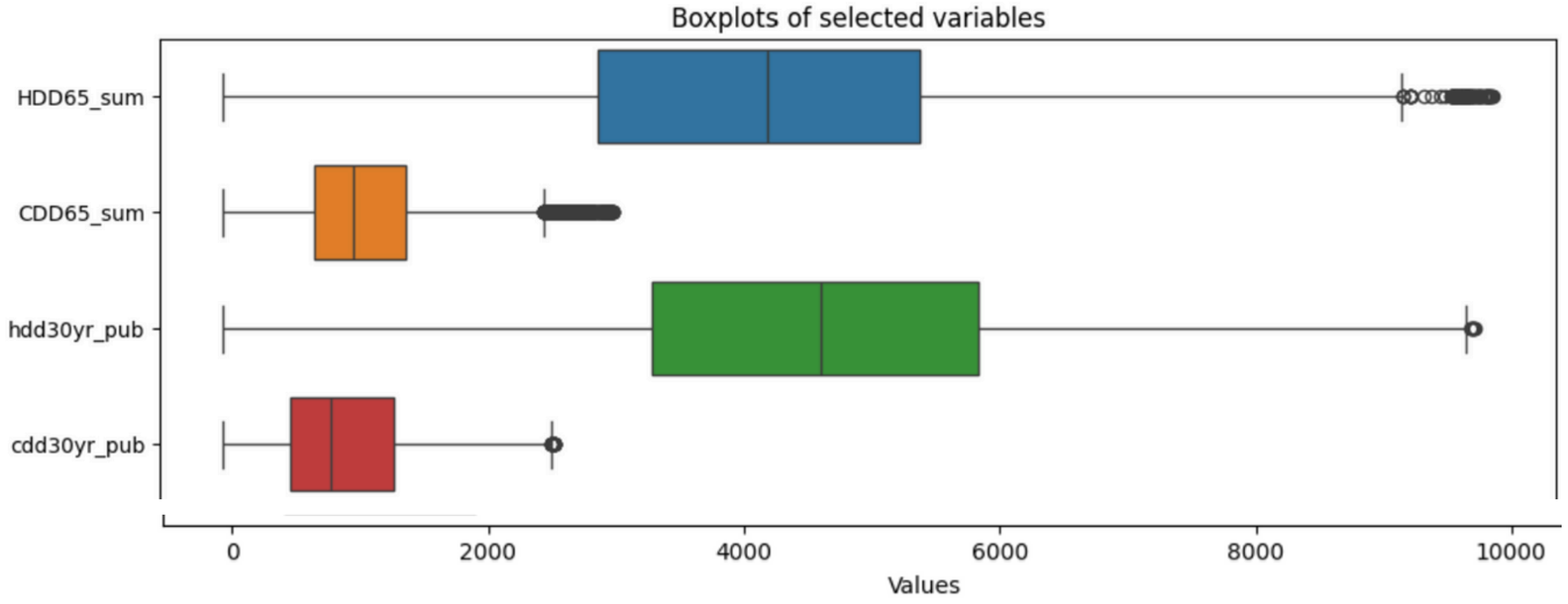
```
6768
```

```
[210] print("Columns with missing values in merged_data:")
      print(columns_with_missing_values)
```

```
Columns with missing values in merged_data:
Index(['medicaldev', 'evchrghome', 'evchrgwks', 'evchrgbus', 'evchrgmuni',
       'evchrgdlr', 'evchrghwy', 'evchrgoth', 'evhomeamt', 'evchrgtype'],
      dtype='object')
```

OUTLIERS

```
[325] possible_outliers = ['HDD65_sum', 'CDD65_sum', 'hdd30yr_pub', 'cdd30yr_pub', 'sqftest', 'totsqft_en', 'tothsqf
      plt.figure(figsize=(12, 8))
      sns.boxplot(data=merged_data[possible_outliers], orient='h')
      plt.title('Boxplots of selected variables')
      plt.xlabel('Values')
```



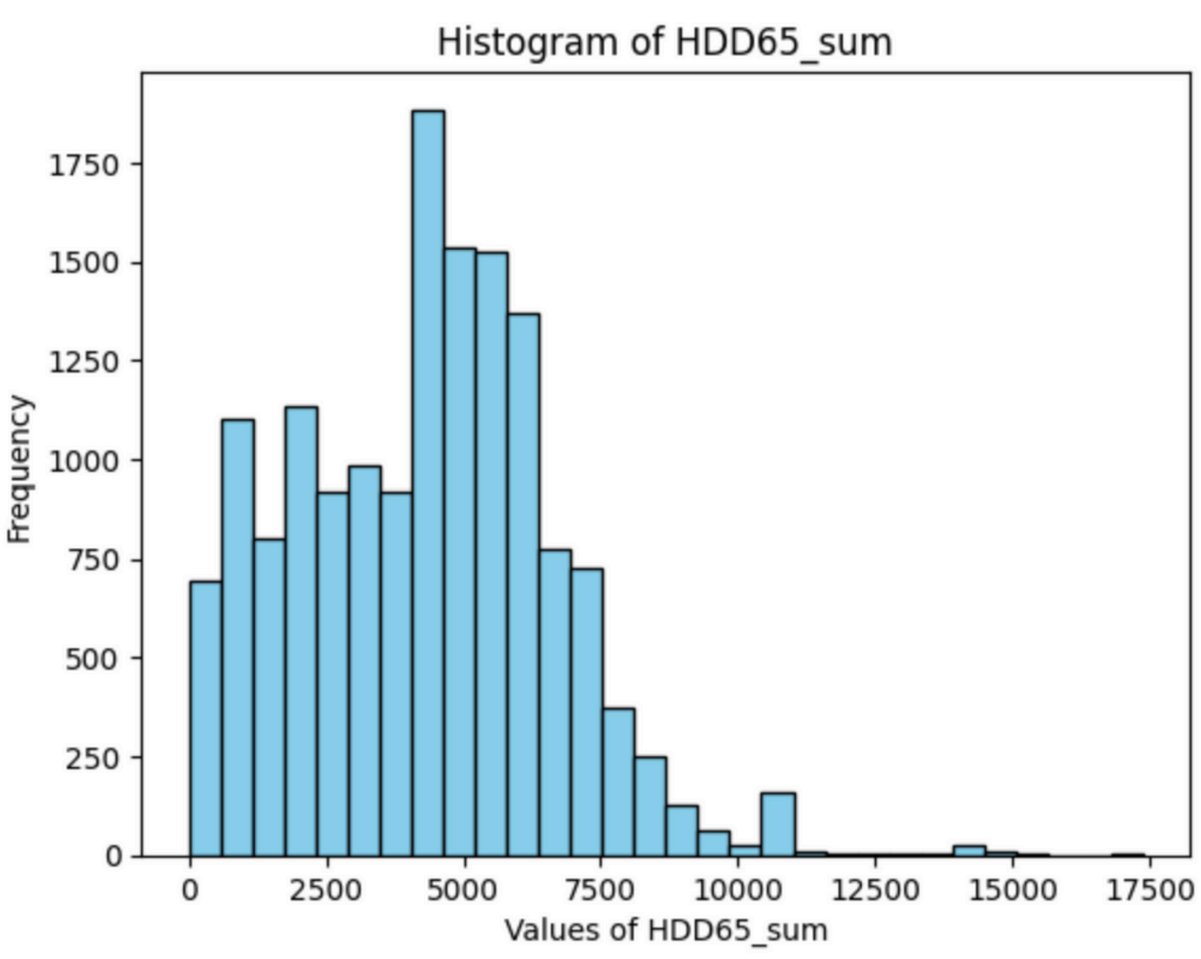
OUTLIERS

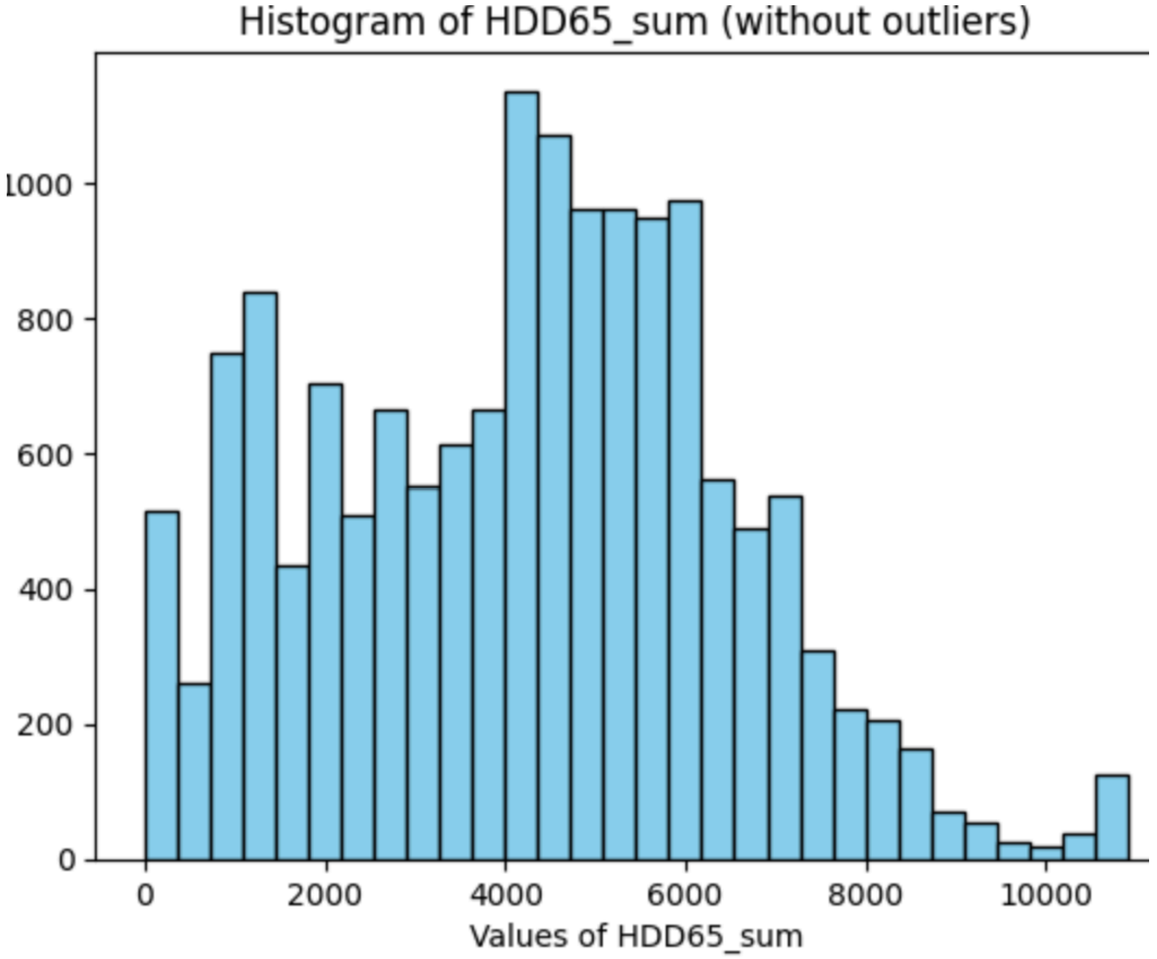
```
[221] def detect_outliers_tukey(data):
          Q1 = np.percentile(data, 25)
          Q3 = np.percentile(data, 75)
          IQR = Q3 - Q1
          lower_bound = Q1 - 1.5 * IQR
          upper_bound = Q3 + 1.5 * IQR
          outliers = (data < lower_bound) | (data > upper_bound)
          return outliers
```

```
[222] columns_to_check = ['HDD65_sum', 'CDD65_sum', 'hdd30yr_pub', 'cdd30yr_pub', 'sqftest', 'totsqft_en', 'tothsqft
```

```
[223] for column in columns_to_check:
          outliers = detect_outliers_tukey(merged_data[column])
          print(f"Number of outliers in'{column}': {outliers.sum()}")
```

```
Number of outliers in'HDD65_sum': 65
Number of outliers in'CDD65_sum': 1590
Number of outliers in'hdd30yr_pub': 64
Number of outliers in'cdd30yr_pub': 896
Number of outliers in'sqftest': 480
Number of outliers in'totsqft_en': 471
Number of outliers in'tothsqft': 444
Number of outliers in'totcsqft': 315
```


Histogram of HDD65_sum
Frequency
Values of HDD65_sum
0
250
500
750
1000
1250
1500
1750
0
2500
5000
7500
10000
12500
15000
17500


Histogram of HDD65_sum (without outliers)
Values of HDD65_sum
0
200
400
600
800
1000
0
2000
4000
6000
8000
10000

SINGLE VALUES

```
for column in merged_data.columns:

    print(f"Unique values in the column '{column}':")
    print(merged_data[column].unique())
    print("\n")
```

```
Unique values in the column 'DOEID':
[117107 112324 104821 ... 110887 115454 115493]


Unique values in the column 'HHID':
[217610 216582  63587 ... 123883  90396 137470]
```

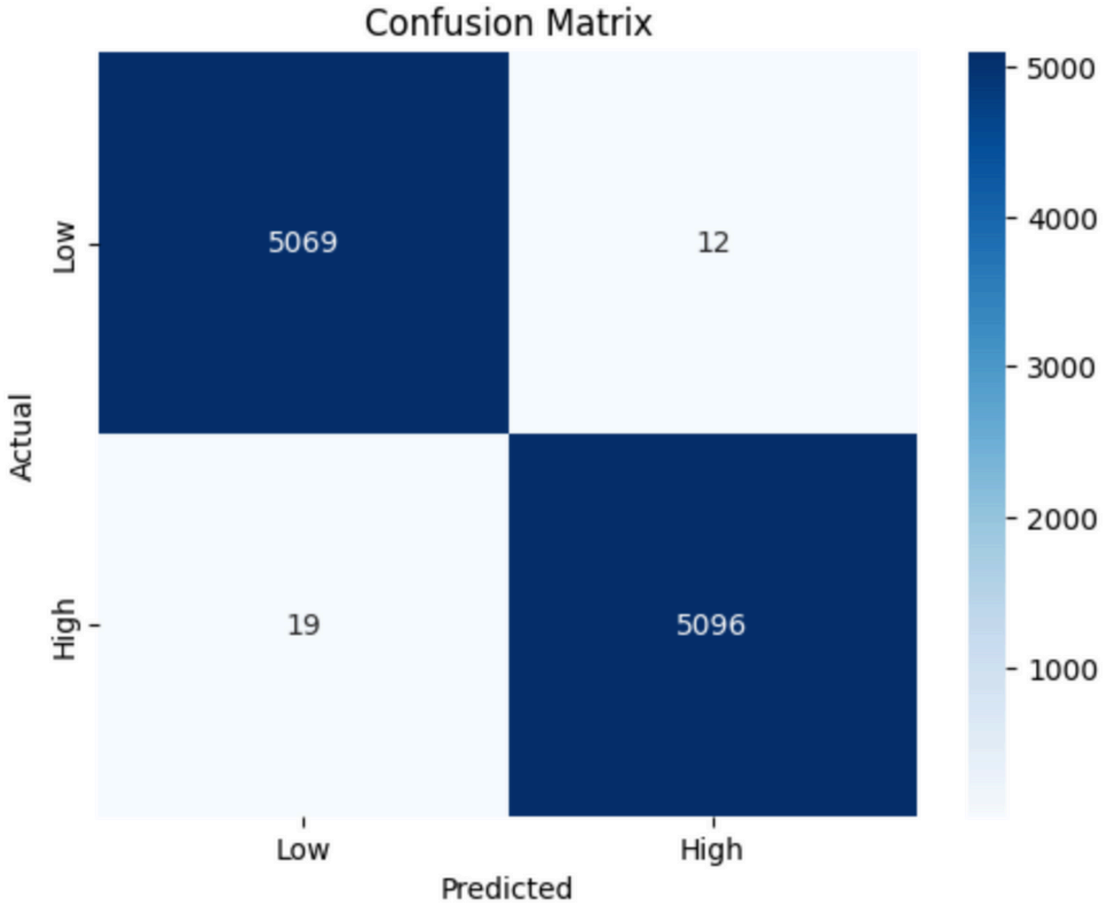
Models

LOGISTIC REGRESSION

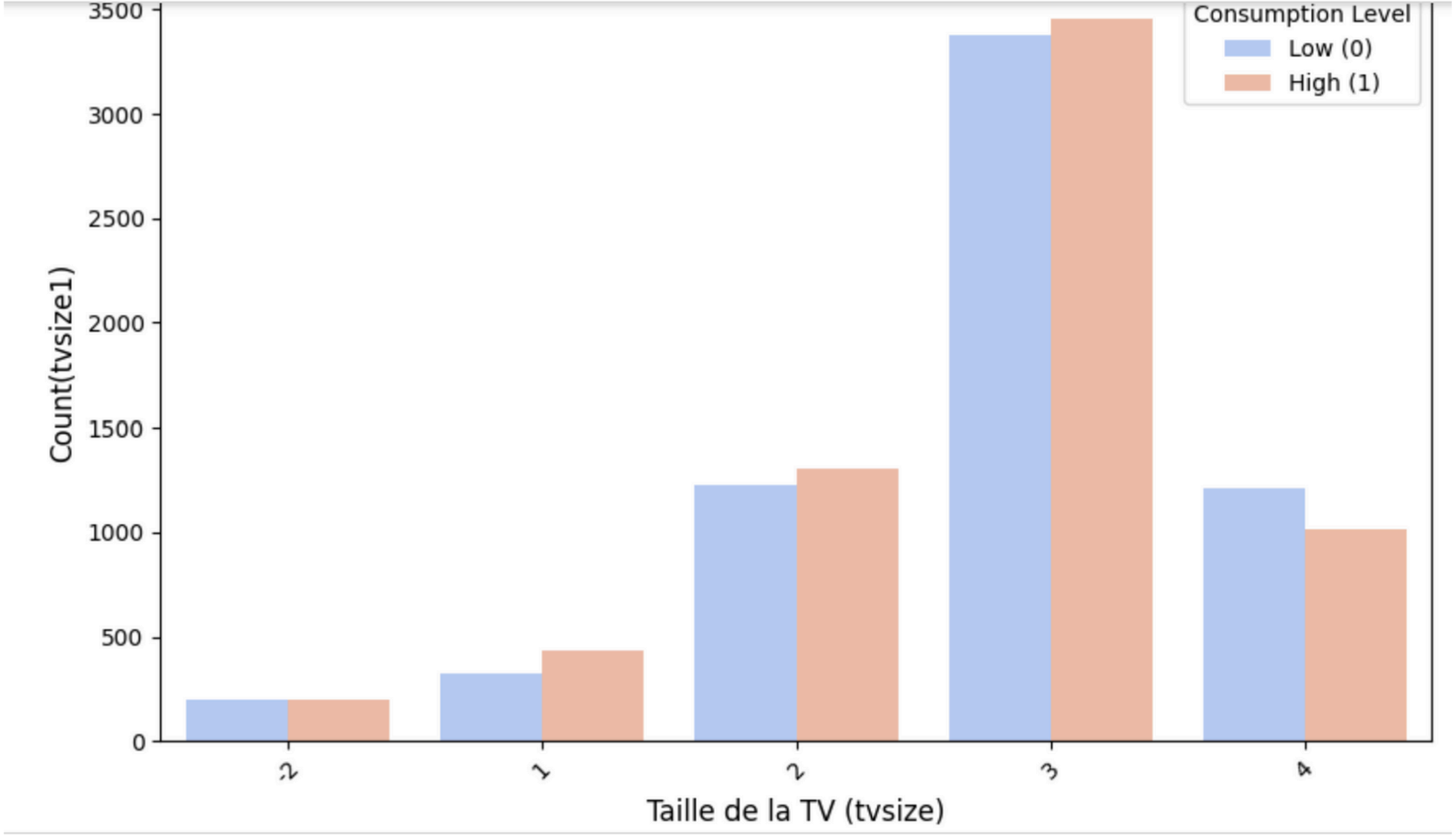
```
Accuracy: 1.00
Confusion Matrix:
[[5069   12]
 [  19 5096]]
Classification Report:
              precision    recall  f1-score   support

           0       1.00      1.00      1.00      5081
           1       1.00      1.00      1.00      5115

    accuracy                           1.00     10196
   macro avg       1.00      1.00      1.00     10196
weighted avg       1.00      1.00      1.00     10196
```



LOGISTIC REGRESSION



LOGISTIC REGRESSION WITH PIPELINE

Pipeline

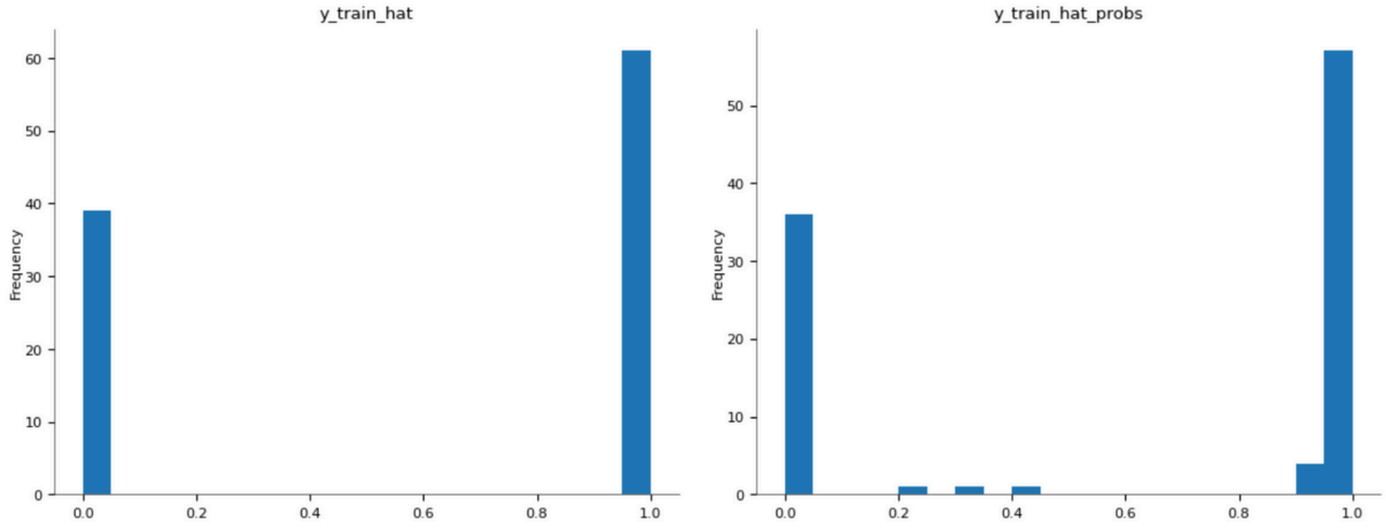
StandardScaler

StandardScaler()

LogisticRegression

LogisticRegression(max_iter=10000)

Distributions



BAGGING REGRESSION

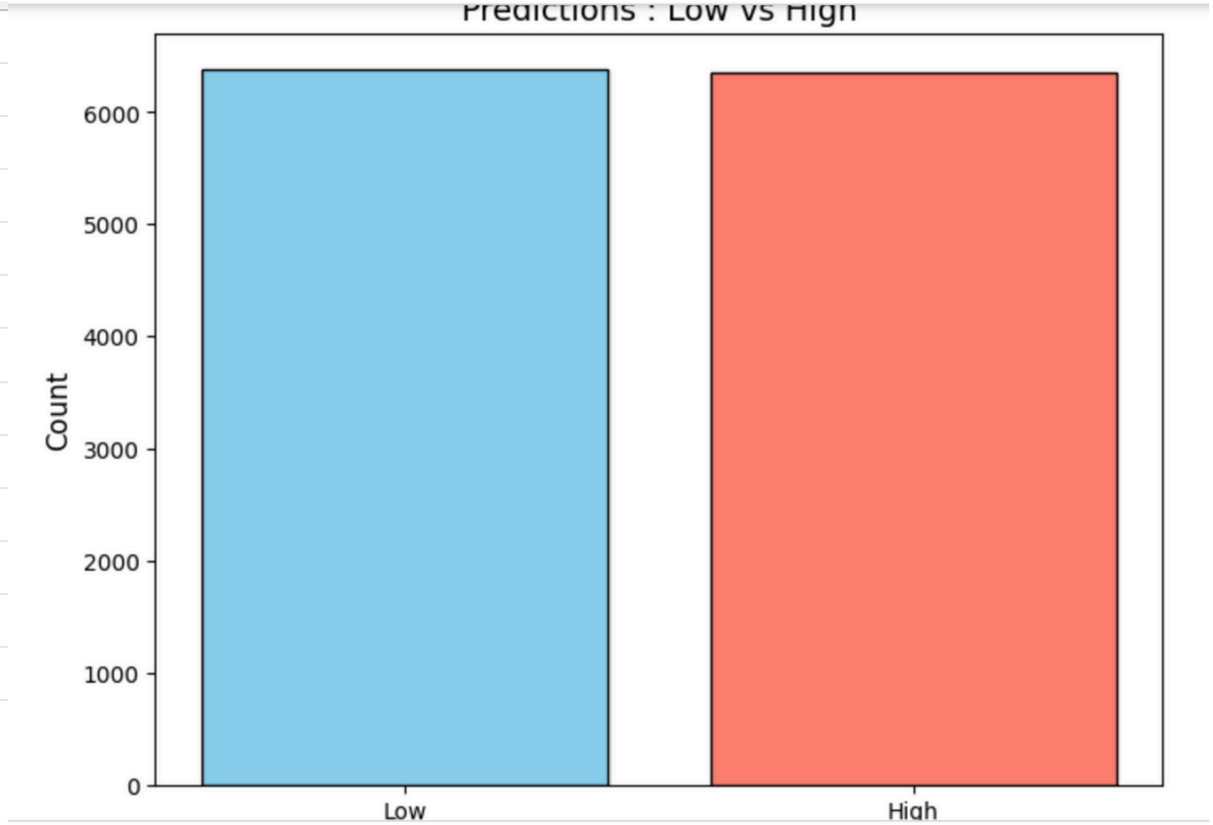
Accuracy: 1.00

Classification Report:

	precision	recall	f1-score	support
0	1.00	1.00	1.00	1874
1	1.00	1.00	1.00	1950
accuracy			1.00	3824
macro avg	1.00	1.00	1.00	3824
weighted avg	1.00	1.00	1.00	3824

XGBOOST

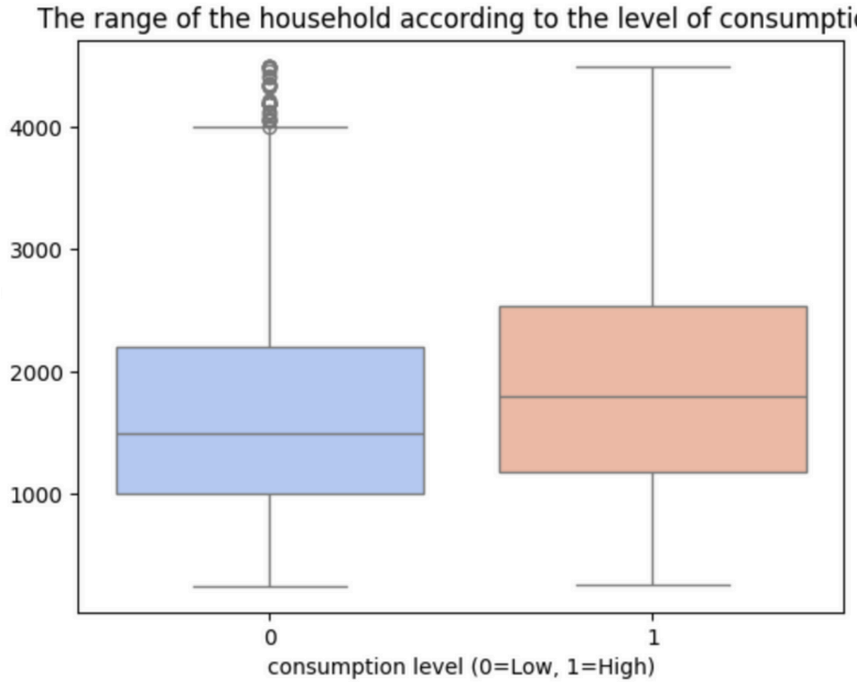
DOEID	predicted_consumption_level	xgb_predicted_consumption_level
117107	low	low
112324	low	low
104821	low	low
106481	low	low
113761	high	high
102598	low	low
101114	high	high
113656	high	high
104678	low	low
101895	high	high
117202	low	low
107528	low	low



XGBOOST

```
[308] scores = cross_val_score(logistic_model, X_scaled, y, cv=5, scoring='accuracy')
      print("Cross-validation scores :", scores)
      print("average score :", scores.mean())
```

```
Cross-validation scores : [0.97920753 0.98469988 0.98352295 0.98234602 0.98155416]
average score : 0.982266108041684
```



CONCLUSION

Key Influences:

Climatic conditions and housing size strongly impact energy consumption.

- **Use energy-efficient appliances to lower consumption**
- **Adopt renewable energy sources for sustainability**
- **Encourage solar panels to decrease dependence on the electricity grid.**