FORECASTING ENERSY CONSUMPTION

ABSTRACT:

This project aims to build a predictive energy consumption model to improve HVAC systems and reduce Carbon Dioxide emission. The project focuses on utilising machine learning algorithms such as CatBoost, XGBoost, LGBM, and linear regression to achieve better accuracy in predicting HVAC energy consumption. The performance of these algorithms will be compared, and their suitability for predicting energy consumption in 1000 different buildings from 13 different undisclosed sites will be evaluated. The dataset used for analysis and modelling contains over 20 million records with 16 features. The project includes exploratory data analysis, feature extraction tasks, and the implementation of various regression models to predict energy consumption accurately. The project's objective is to find a better solution for developing accurate and reliable predictive models that can contribute to reducing carbon emissions and promoting sustainable practices in the building sector.

INTRODUCTION:

In recent years, the excessive release of greenhouse gases has dramatically impacted climate change. Various human activities, such as transportation, deforestation, and industrial processes produce Carbon Dioxide gas which detriment's the earth's atmosphere. With the rate of Carbon Dioxide emissions increasing ([1] Köne & Büke, 2010), it is imperative that interventions are developed to regulate CO2 emission.

Energy consumption in buildings, such as heating, ventilation, and electricity usage, cause a great amount of Carbon Dioxide emission ([2] Santamouris et al., 2001). An algorithm to identify patterns of energy consumption can be useful for building owners and energy providers to improve energy efficiency. The purpose of this project is to build a predictive energy consumption model to improve HVAC systems and reduce Carbon Dioxide emission.

Energy consumption prediction in machine learning is a rapidly growing field of research that has gained significant attention in recent years. While various machine learning techniques, including artificial neural networks, decision trees, and regression algorithms, have been used to develop predictive models for energy consumption, there is still a knowledge gap in terms of developing accurate and reliable models that can be applied to Heating, Ventilation, and Air Conditioning (HVAC) systems in different types of buildings. To address this gap, our project focuses on utilising algorithms such as CatBoost, XGBoost, LGBM, and linear regression to achieve better accuracy in predicting HVAC energy consumption ([3] Zhao and Magoulès, 2012). We will compare the performance of these algorithms and evaluate their suitability for predicting energy consumption in 1000 different buildings from 13 different undisclosed sites. We will also collect weather data from the nearest weather stations to the buildings and

incorporate it into our predictive models to improve their accuracy. By exploring these algorithms, we aim to find a better solution for developing accurate and reliable predictive models that can contribute to reducing carbon emissions and promoting sustainable practices in the building sector.

MATERIALS AND METHODS:

Dataset

The dataset is a reference from Kaggle competition:

Source: https://www.kaggle.com/c/ashrae-energy-prediction/overview

The data has over 1000 buildings over a time frame of 3 years, which contains the attributes which describe the building, meter category and weather details. It can be used to predict Energy consumption in kWh. We have 3 Files of data(Weather Metadata File, Building Metadata File, Meter Reading File) which are used for Analysis and Modeling.

Data Description

Basic Statistics:

The dataset has 20 million records with 16 features
No of Training Samples: 20,216,100 – 70% *(*14,151,270)
No of Testing Samples: Subset of Training Samples (30%) – (6,064,830)

Attributes Description:

These are the attributes present in the data sets:

building_id – Building Identifier

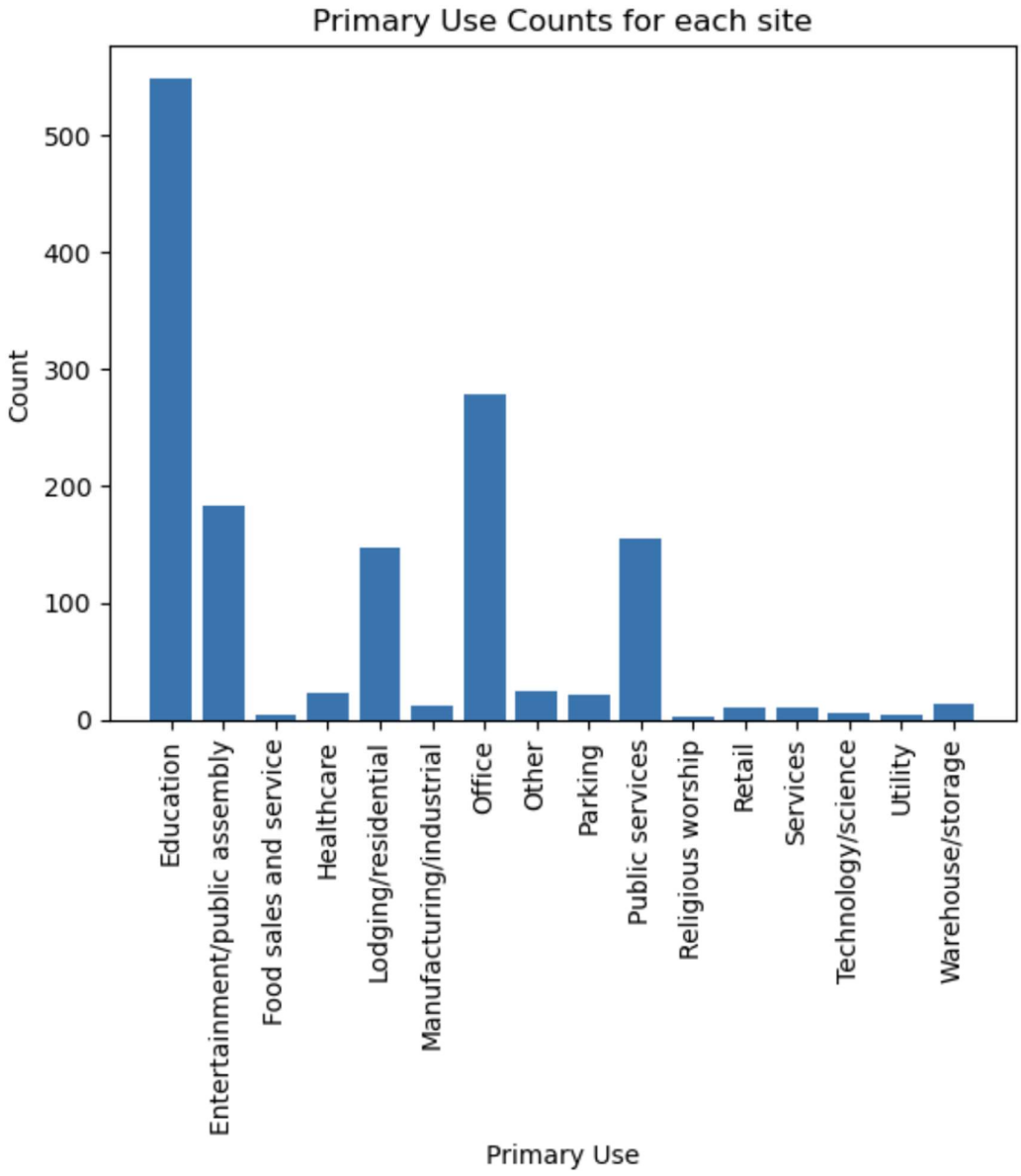
meter – It is a code read as

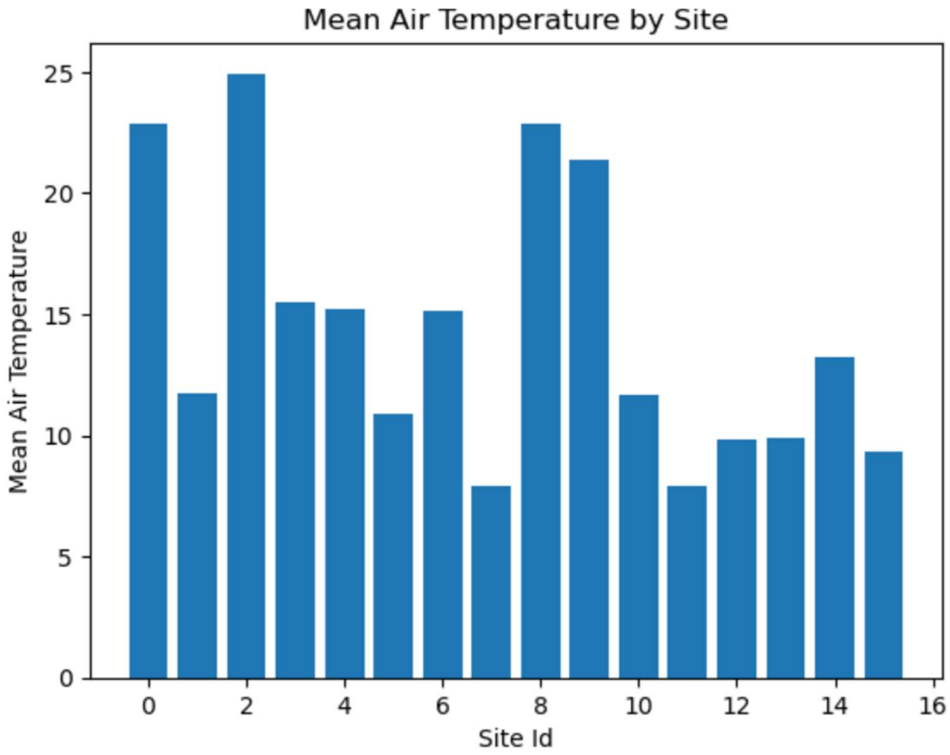
- 0: Electricity
- 1: Chilledwater
- 2: Steam
- 3: Hotwater

timestamp – Time and Date when measurements are taken.
meter_reading – Energy Consumption is kWh, Considered as Class Label/Target Label.
site_id – Identity for Weather files.
primary_use – Specifies the category for which the building is used.
Square_feet – Floor area of the building.
Year_built – Year in which the building was built.
floor_count – Indicates the number of floors in a building.
air_temperature – Measure of Temperature in Degree Celsius.
cloud_coverage – Amount of cloud cover at given location in **okta.**
dew_temperature – Temperature in Degree Celsius.

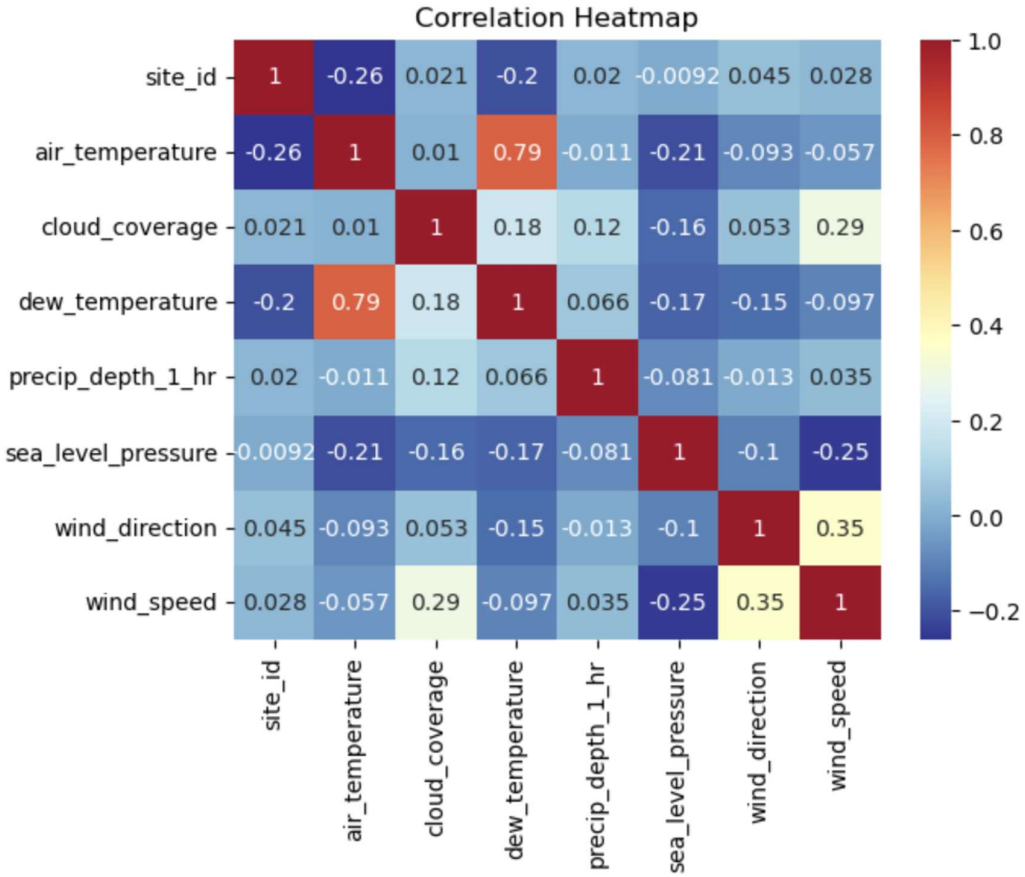
Precip_depth_1_hr – Precipitation depth for 1 hour in Millimeters.
sea_level_pressure – Sea Level Pressure in Millibar/Hectopascals.
wind_direction – Direction of wind in Compass (0-360o).
wind_speed – Speed of Wind in Meters per second.

Exploratory Data Analysis:

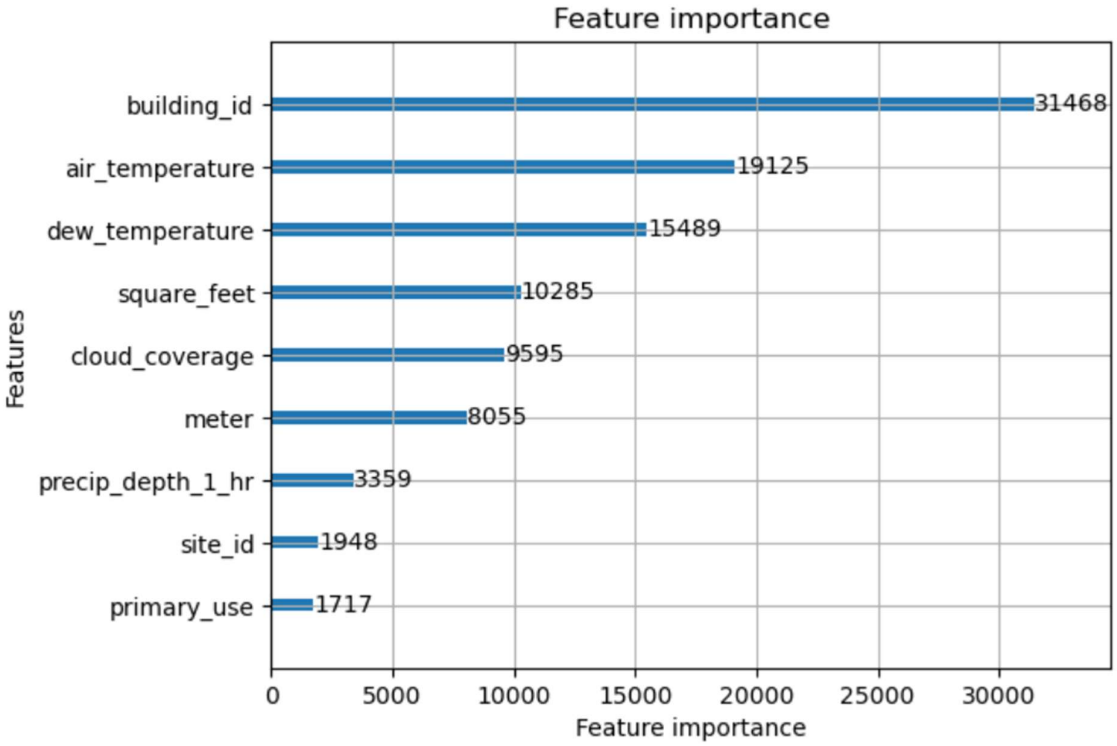



Mean Air Temperature by Site
Mean Air Temperature
Site Id
0
5
10
15
20
25
0
2
4
6
8
10
12
14
16

Heat Map:


Correlation Heatmap
site_id
air_temperature
cloud_coverage
dew_temperature
precip_depth_1_hr
sea_level_pressure
wind_direction
wind_speed
1 -0.26 0.021 -0.2 0.02 -0.0092 0.045 0.028
-0.26 1 0.01 0.79 -0.011 -0.21 -0.093 -0.057
0.021 0.01 1 0.18 0.12 -0.16 0.053 0.29
-0.2 0.79 0.18 1 0.066 -0.17 -0.15 -0.097
0.02 -0.011 0.12 0.066 1 -0.081 -0.013 0.035
-0.0092 -0.21 -0.16 -0.17 -0.081 1 -0.1 -0.25
0.045 -0.093 0.053 -0.15 -0.013 -0.1 1 0.35
0.028 -0.057 0.29 -0.097 0.035 -0.25 0.35 1
1.0
0.8
0.6
0.4
0.2
0.0
−0.2

Feature Importance:



Feature Extraction Tasks:

There are 15 variables available in the weather and building data sets, we performed the majority of the tasks on the weather dataset because of the feature importance of the dataset.

Task 1:

After calculating null values, we have decided to drop few columns with high missing values

In [14]:
```
weather.isna().sum()/weather.shape[0] * 100.00
```

Out[14]:
```
site_id                0.000000
timestamp              0.000000
air_temperature        0.039350
cloud_coverage        49.489529
dew_temperature        0.080845
precip_depth_1_hr     35.979052
sea_level_pressure     7.596603
wind_direction         4.484414
wind_speed             0.217496
dtype: float64
```

In [15]:
```
building.isna().sum()/building.shape[0] * 100.00
```

Out[15]:
```
site_id          0.000000
building_id      0.000000
primary_use      0.000000
square_feet      0.000000
year_built      53.416149
floor_count     75.500345
dtype: float64
```

In [16]:
```
train.isna().sum()/train.shape[0] * 100.00
```

Out[16]:
```
building_id      0.0
meter            0.0
timestamp        0.0
meter_reading    0.0
dtype: float64
```

Task 2:

```
train.head()
```

	building_id	meter	timestamp	meter_reading
0	105	0	2016-01-01 00:00:00	23.3036
1	106	3	2016-01-01 00:00:00	0.0000
2	108	0	2016-01-01 00:00:00	91.2653
3	109	0	2016-01-01 00:00:00	80.9300
4	109	3	2016-01-01 00:00:00	0.0000

```
def add_time_stamp_features(df,timestamp_colname):

    """Adding time stamp features"""

    df["datetime"] = pd.to_datetime(df[''+timestamp_colname+''])
    df["day"] = df["datetime"].dt.day
    df["week"] = df["datetime"].dt.week
    df["month"] = df["datetime"].dt.month
```

We are splitting the timestamp feature in train dataset into multiple columns for better readability and optimized prediction.

Task 3:

Label Encoding

We have defined a function for the label encoding purpose. We have label encoded the variable "primary_use"

```
def label_encoding(df,column_name):

    """Label encoding column name from a data frame"""

    le = LabelEncoder()
    df[''+column_name+''] = le.fit_transform(df[''+column_name+''])
```

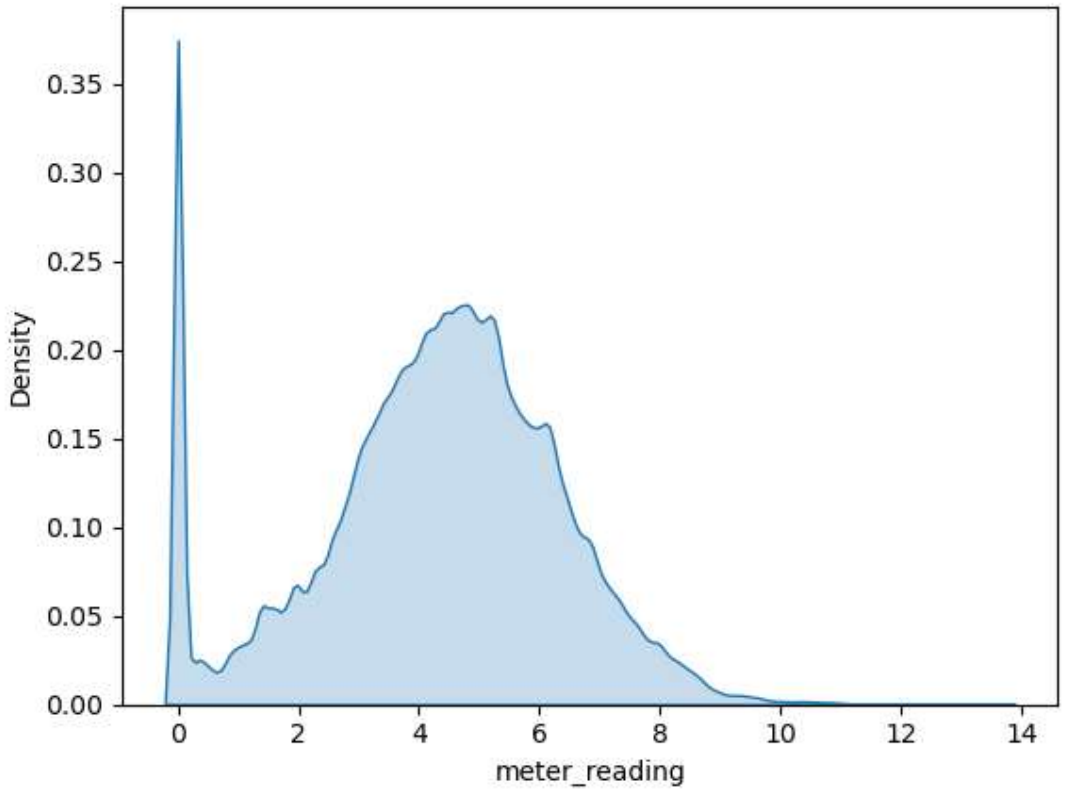
```
label_encoding(building,"primary_use")
```

```
building.head()
```

	site_id	building_id	primary_use	square_feet	year_built	floor_count
0	0	0	0	7432	2008.0	NaN
1	0	1	0	2720	2004.0	NaN
2	0	2	0	5376	1991.0	NaN
3	0	3	0	23685	2002.0	NaN
4	0	4	0	116607	1975.0	NaN

Task 4:

Applying log transform for the target variable and using density plot to see the distribution of values



Task 5:

Imputing the missing values and Nan values

We have defined a function to fill the missing values with forward fill and then filling values with mean for that day, in that month.

```
def weather_data_filler_1(df,column,method='ffill',by='mean'):

    """Filling the NA values by forward fill and then by mean"""

    if by=='mean':
        filler = df.groupby(['site_id','day','month'])[''+column+''].mean()
    elif by=='count':
        filler = df.groupby(['site_id','day','month'])[''+column+''].count()
    elif by=='min':
        filler = df.groupby(['site_id','day','month'])[''+column+''].min()
    elif by=='max':
        filler = df.groupby(['site_id','day','month'])[''+column+''].max()
    elif by=='mode':
        filler = df.groupby(['site_id','day','month'])[''+column+''].mode()

    filler = pd.DataFrame(filler.fillna(method=method),columns=[''+column+''])

    df.update(filler,overwrite=False)
```

We filled that following columns with the above function:

cloud_coverage, dew_temperature, sea_level_pressure, wind_direction, wind_speed, and precip_depth_1_hr

Task 6:

Removing the following features with low importance

"Timestamp","sea_level_pressure","wind_direction","wind_speed","year_built",

"floor_count"

Machine Learning Methods:

This is the most vital part while practising machine learning, it is important to understand how a model works and selecting the best suitable model which does not either underfit or overfit the model. The objective of the project is to use regression models to predict the energy consumed by the building. So we tried the following models to find the best model which can predict the usage with least errors. Models Implemented:

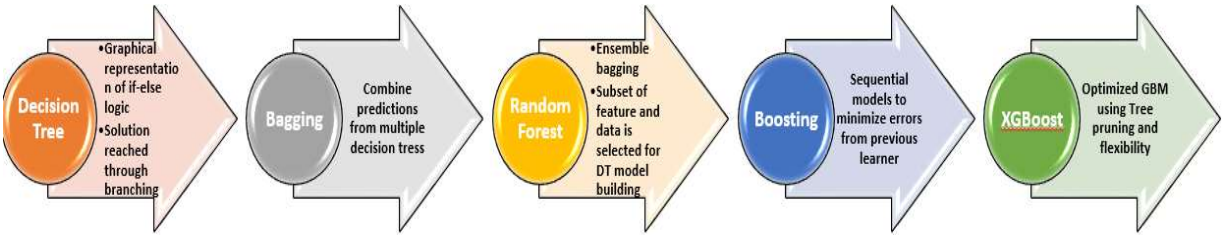
1. Linear Regression: It is the most widely used modelling technique, which assumes a linear connection between an independent variable and dependable variable. It employs

a best-fit line, also known as a regression line. The linear regression model is simple (with just one variable) or complex (multiple dependent and independent variables).

2. eXtreme Gradient Boost (XGBoost): XGBoost is a Machine Learning Algorithm from the decision tree family that operates with a boosting mechanism. This algorithm is primarily concerned with memory and computation resource efficiency. The primary purpose of this approach is to make the most effective use of available modeling resources. XGBoost is an ensemble tree approach that employs the gradient descent architecture to boost weak learners.

   Features of XGBoost:

   - Parallelized tree building
   - Efficient missing data handling
   - Tree pruning using Depth First Search
   - Capability of inbuilt Cross validation



3. Light Gradient Boosting Machine (LGBM): LightGBM is a decision tree-based gradient boosting framework that improves model efficiency while reducing memory utilisation. It employs two innovative techniques: Gradient-based One Side Sampling and Exclusive Feature Bundling (EFB), which address the drawbacks of the histogram-based approach employed in most Gradient Boosting Decision Tree frameworks. The properties of the LightGBM Algorithm are formed by the two methodologies of GOSS and EFB.. They work together to make the model run smoothly and give it an advantage over competing GBDT frameworks.

4. CatBoost: is a decision tree-based machine learning algorithm. CatBoost is derived from the term "categorical boosting." The built in feature allows it to handle categorical variables natively, allowing it to improve models that require very few category transformations. It employs the same gradient boosting algorithm as XGBoost. Gradient boosting fits the decision trees progressively, allowing the fitted trees to learn from the mistakes of previous trees and therefore reduce model errors. The method is repeated until the loss function can no longer be minimised. It can handle null values in data and deal with various data types. CatBoost has a lot of versatility when dealing

with Energy Consumption Prediction with heterogeneous, sparse, and category data. There is overfitting detector feature, which prevents algorithm from building the new trees by understanding if the data is being overfit

Performance Evaluation:

We have used the following metrics to evaluate the model after training: 1. RMSLE (Root Mean Squared Log Error) 2. RMSE (Root Mean Squared Error) 3. MAE (Mean Absolute Error)

Hyper Parameter tuning:

We used hyperparameters like learning rate, the number of leaves to prevent overfitting, regularisation parameter lambda, iterations, n estimators, columns by a tree, feature fraction, and early stopping rounds.

CatBoost:

```
model=CatBoostRegressor(iterations=num_iters,
                        depth=n_depth, learning_rate=learning_rate,
                        loss_function='RMSE')
```

Light Gradient Boost Model (LGBM):

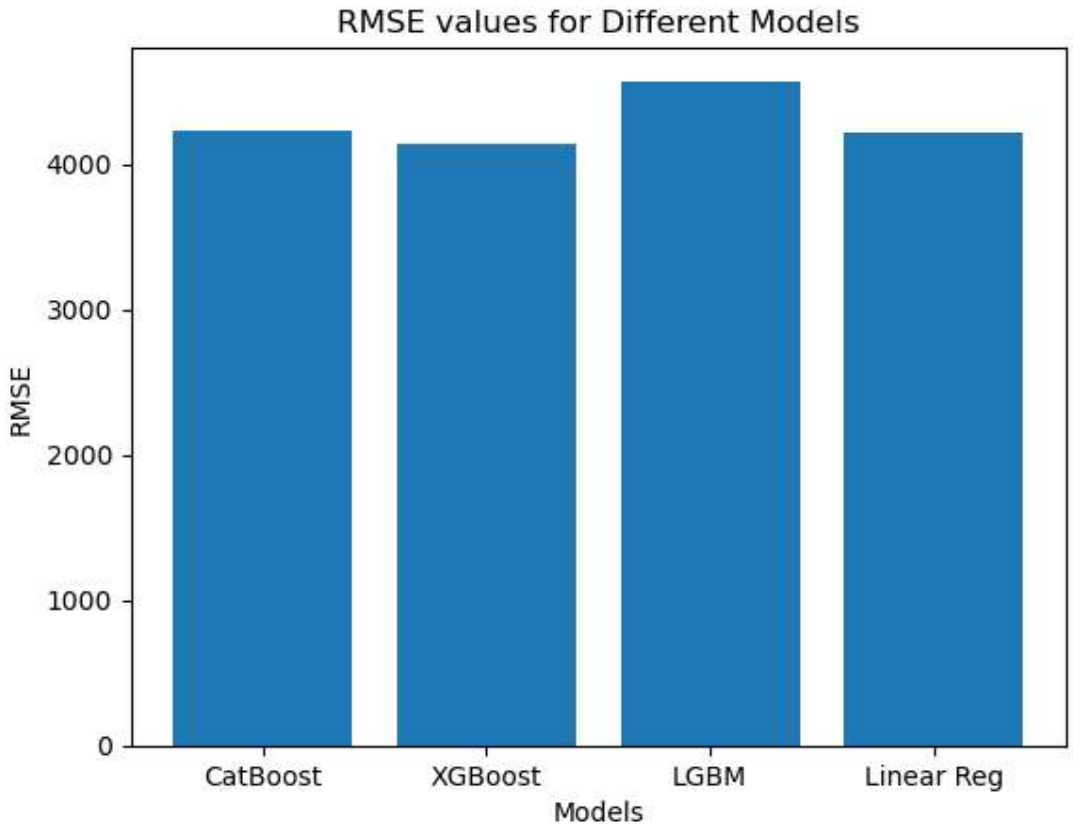
```
categorical_features = ["building_id", "site_id", "meter", "primary_use", "weekend"]

params = {
    "objective": "regression",
    "boosting": "gbdt",
     "num_leaves": 1280,
    "learning_rate": 0.05,
     "feature_fraction": 0.85,
     "reg_lambda": 2,
    "metric": "rmse",
}

model_LGBM=LGBM(categorical_features,params,3,features,target)
```

eXtreme Gradient Boost (XGBoost):

```
pars = {
    'colsample_bytree': 0.8,
    'learning_rate': 0.1,
    'max_depth': 5,
    'subsample': 0.8,
    'objective': 'reg:squarederror',
}
XGB_model=XGBoost(X_train,y_train,X_test,y_test,pars,5)
```

RESULTS:

Model	Data	RMSE	RMSLE	MAE
XGBoost	Test	4134.247	1.357806	311.372
	Train	4259.768	1.357577	312.962
Linear Regression	Test	4221.512	1.780983	367.490
	Train	4345.176	1.780404	369.097
LGBM	Test	4570.9	2.7	639
	Train	4449	2.5	670.23
CatBoost	Test	4229.658	1.785	395.795
	Train	4353.126	1.784	397.416



DISCUSSIONS:

From the results above, we can see that XGBoost outperforms all other models with lowest RMSLE, RMSE and MAE values.

We have tried to detect the patterns of different attributes of weather, buildings, and temperatures for each site for Data Analysis.

We discovered that XGBoost outperformed all other models, including CatBoost, LGBM, and linear regression, in terms of predicting energy consumption. This finding is consistent with prior research suggesting that XGBoost is a powerful algorithm for regression problems ([11]Chen & Guestrin, 2016). Our interpretation of these results is that incorporating weather data, especially wind speed and cloud coverage, can help improve the accuracy of energy consumption predictions. This observation is supported by previous studies that have also highlighted the significance of weather data in developing accurate energy consumption models ([10]. Mel Keytingan, Nor Azuana, Lilik J. Awalin).

When comparing our results to the reference paper, we observed some differences in the chosen methodologies. While our project utilized CatBoost, XGBoost, LGBM, and linear regression,

the reference paper focused on k-Nearest Neighbor, Support Vector Machine with Radial Basis Function kernel, and Artificial Neural Network with Multilayer Perceptron model. Despite these differences in the employed algorithms, both studies aimed to reduce energy consumption in buildings through accurate predictive models. The reference paper found that the Support Vector Machine algorithm provided the most promising results, while our study demonstrated that XGBoost had the best performance.

Some limitations of our study include the restricted number of buildings studied and the potential influence of external factors that were not considered in our models. Future research could focus on incorporating additional features that impact energy consumption, such as building occupancy and usage patterns, and exploring alternative machine learning algorithms. Moreover, investigating the potential of hybrid or ensemble methods, as recommended by the reference paper, could lead to further improvements in prediction accuracy.

CONCLUSION:

In this project, we developed a predictive energy consumption model to improve HVAC systems and reduce Carbon Dioxide emission. We compared the performance of four machine learning algorithms, CatBoost, XGBoost, LGBM, and linear regression, in predicting energy consumption and evaluated the effect of weather data on the models' performance. The results showed that XGBoost had the best performance, followed by CatBoost and linear regression, and LGBM had the least performance. Furthermore, the inclusion of weather data improved the models' performance, especially in the case of CatBoost and XGBoost.

This project can contribute to reducing carbon emissions and promoting sustainable practices in the building sector by helping building owners and energy providers to identify energy-efficient practices and reduce energy consumption. Future research can focus on developing more accurate and reliable models for predicting energy consumption in buildings, incorporating additional features that affect energy consumption, and exploring different machine learning algorithms.

REFERENCES:

[1]. https://www.sciencedirect.com/science/article/pii/S136403211000153X

[2]. https://www.sciencedirect.com/science/article/pii/S0038092X00000955

[3].https://www.sciencedirect.com/science/article/pii/S2352484723001026#bb134

[4].https://interiordesign.net/projects/8-sustainably-designed-and-architecturally-significant-buildings-in-singapore/

[5].https://towardsdatascience.com/https-medium-com-vishalmorde-xgboost-algorithm-long-she-may-rein-edd9f99be63d

[6].https://www.displayr.com/what-is-linear-regression/

[7].https://www.jeremyjordan.me/hyperparameter-tuning/

[8].https://towardsdatascience.com/bayesian-optimization-and-hyperparameter-tuning-6a22f14cb9fa

[9].https://arxiv.org/abs/2007.06933

[10] https://www.sciencedirect.com/science/article/pii/S266616592030034X

[11] https://dl.acm.org/doi/abs/10.1145/2939672.2939785