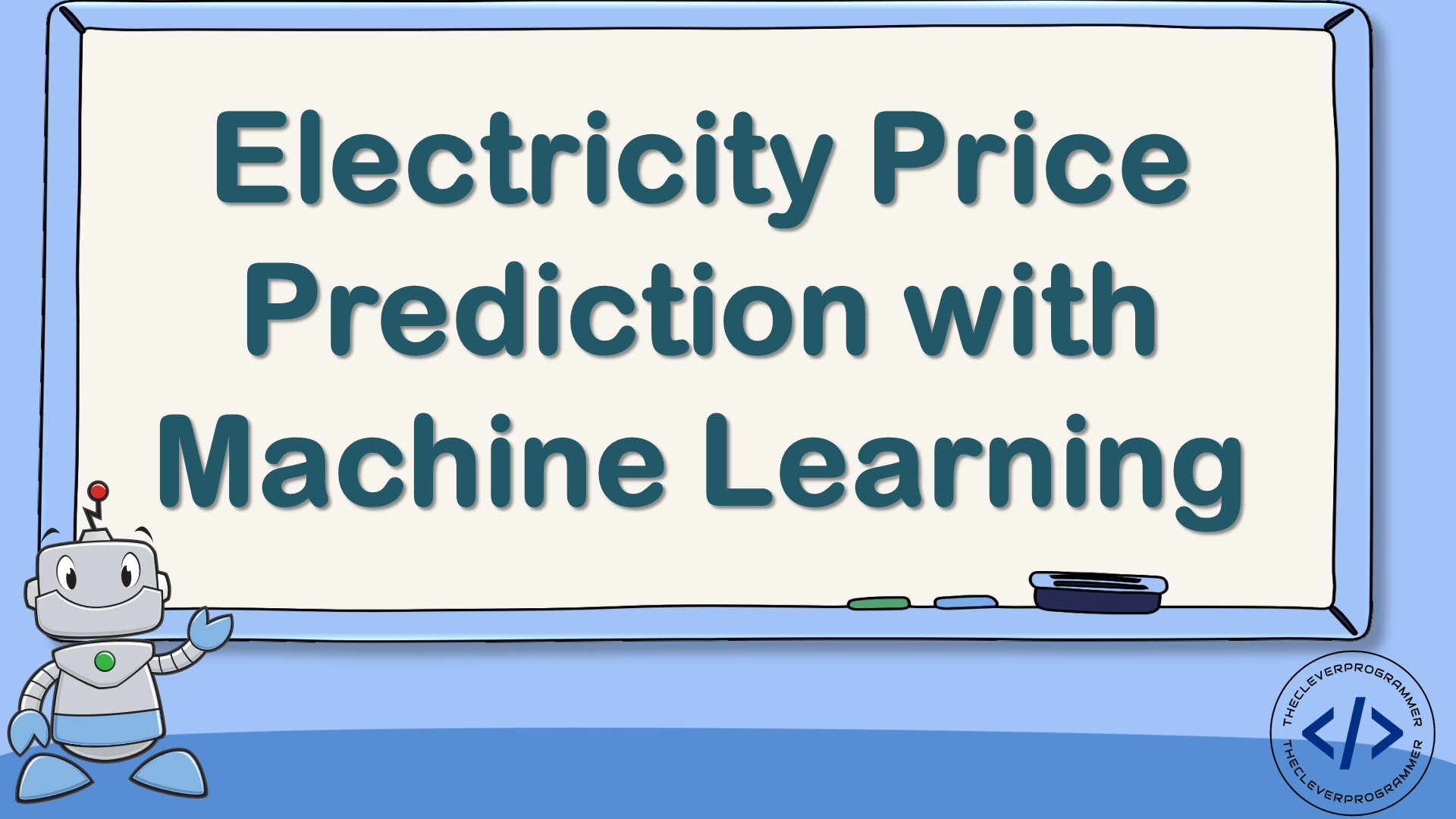
**APPLIED DATA SCIENCE:**

**ELECTRICITTY PRICE PREDICTION**

PHASE 3: DEVELOPMENT PART1

Topic: *Start building the electricity price prediction model by loading and pre-processing the dataset*



Electricity price predictions

# Introduction

* The price of electricity depends on many factors.
* Predicting the price of electricity helps many businesses understand how much electricity they have to pay each year.
* The Electricity Price Prediction task is based on a case study where you need to predict the daily price of electricity based on the daily consumption of heavy machinery used by businesses.
* The task of electricity price prediction with machine learning using python

**Problem definition:**

Suppose that your business relies on computing services where the power consumed by your machines varies throughout the day.

You do not know the actual cost of the electricity consumed by the machines throughout the day, but the organization has provided you with historical data of the price of the electricity consumed by the machines.

Below is the information of the data we have for the task of forecasting electricity prices:

**DateTime:** Date and time of the record

**Holiday:** contains the name of the holiday if the day is a national holiday

**HolidayFlag:** contains 1 if it’s a bank holiday otherwise 0

**DayOfWeek:** contains values between 0-6 where 0 is Monday

**WeekOfYear**: week of the year Day: Day of the date Month: Month of the date Year: Year of the date

**PeriodOfDay**: half-hour period of the day

**ForcastWindProduction**: forecasted wind production SystemLoadEA forecasted national load

**SMPEA**: forecasted price

**ORKTemperature:** actual temperature measured

**ORKWindspeed:** actual windspeed measured

**CO2Intensity:** actual C02 intensity for the electricity produced

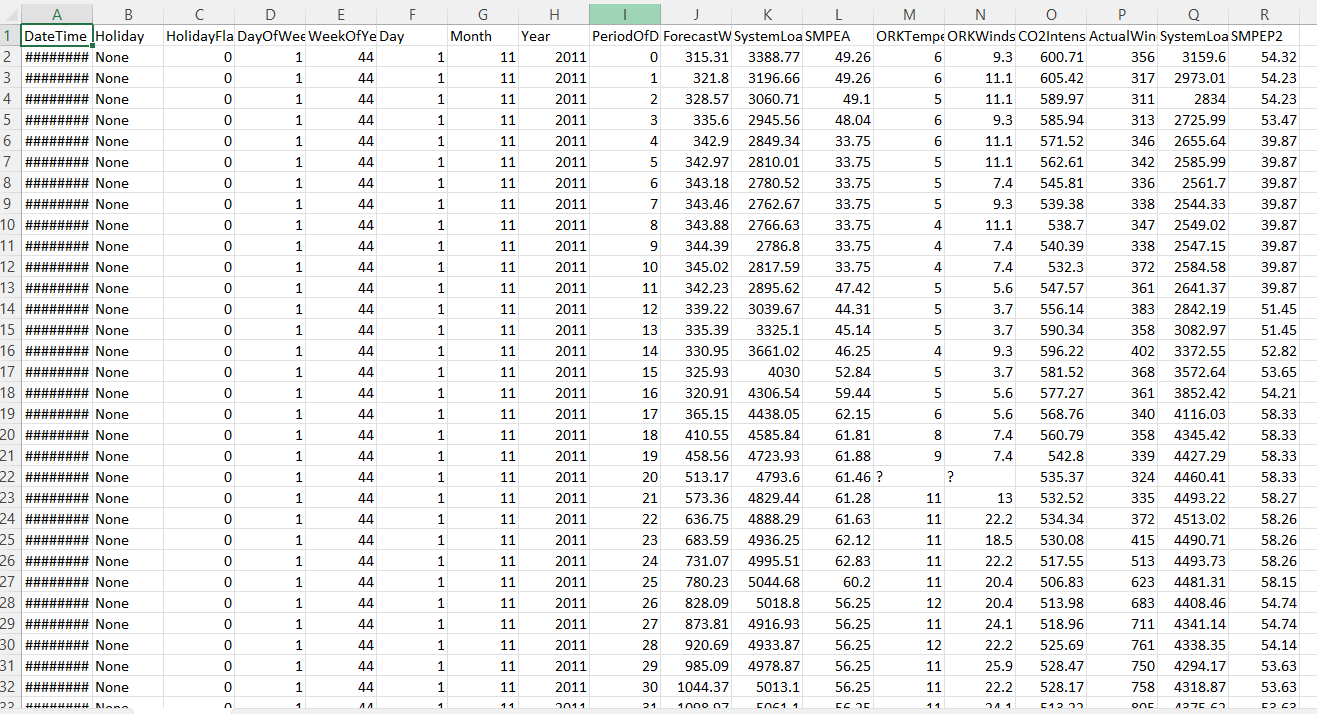
**ActualWindProduction:** actual wind energy production

**SystemLoadEP2:** actual national system load

SMPEP2: the actual price of the electricity consumed (labels or values to be predicted)

So your task here is to use this data to train a machine learning model to predict the price of electricity consumed by the machines. In the section below, I will take you through the task of electricity price prediction with machine learning using Python.

Given dataset:



Steps to process the given dataset:

1. Prediction using Python I will start the task of electricity price prediction by importing the necessary Python libraries and the dataset that we need for this task for data manipulation and machine learning. Common libraries include pandas, numpy, and scikit-learn.

**Program:**

**Load the Dataset:**

* Use pandas to load the diabetes dataset into a DataFrame.

**For example:**

import pandas as pd

# Load the dataset

data = pd.read\_csv('electricity\_price\_prediction.csv')

pd.read()

DateTime Holiday HolidayFlag DayOfWeek WeekOfYear Day Month \

0 01/11/2011 00:00 None 0 1 44 1 11

1 01/11/2011 00:30 None 0 1 44 1 11

2 01/11/2011 01:00 None 0 1 44 1 11

3 01/11/2011 01:30 None 0 1 44 1 11

4 01/11/2011 02:00 None 0 1 44 1 11

Year PeriodOfDay ForecastWindProduction SystemLoadEA SMPEA \

0 2011 0 315.31 3388.77 49.26

1 2011 1 321.80 3196.66 49.26

2 2011 2 328.57 3060.71 49.10

3 2011 3 335.60 2945.56 48.04

4 2011 4 342.90 2849.34 33.75

ORKTemperature ORKWindspeed CO2Intensity ActualWindProduction SystemLoadEP2 \

0 6.00 9.30 600.71 356.00 3159.60

1 6.00 11.10 605.42 317.00 2973.01

2 5.00 11.10 589.97 311.00 2834.00

3 6.00 9.30 585.94 313.00 2725.99

4 6.00 11.10 571.52 346.00 2655.64

SMPEP2

0 54.32

1 54.23

2 54.23

3 53.47

4 39.87

**Data Exploration:**

* Explore the dataset to understand its structure and characteristics. This helps you identify any issues that need to be addressed in the preprocessing phase.
* Use data.head() to view the first few rows of data.
* Use data.info() to check for data types, missing values, and the number of non-null entries.
* Use data.describe() to get summary statistics of the dataset.

**For example:**

Data.head()

#Check for missing values

Print(df.isnull().sum())

#explore statistics

Print(df.describe())

<class 'pandas.core.frame.DataFrame'>

RangeIndex: 38014 entries, 0 to 38013

Data columns (total 18 columns):

# Column Non-Null Count Dtype

--- ------ -------------- -----

0 DateTime 38014 non-null object

1 Holiday 38014 non-null object

2 HolidayFlag 38014 non-null int64

3 DayOfWeek 38014 non-null int64

4 WeekOfYear 38014 non-null int64

5 Day 38014 non-null int64

6 Month 38014 non-null int64

7 Year 38014 non-null int64

8 PeriodOfDay 38014 non-null int64

9 ForecastWindProduction 38014 non-null object

10 SystemLoadEA 38014 non-null object

11 SMPEA 38014 non-null object

12 ORKTemperature 38014 non-null object

13 ORKWindspeed 38014 non-null object

14 CO2Intensity 38014 non-null object

15 ActualWindProduction 38014 non-null object

16 SystemLoadEP2 38014 non-null object

17 SMPEP2 38014 non-null object

dtypes: int64(7), object(11)

memory usage: 5.2+ MB

**Data Preprocessing:**

* Prepare the data for modeling by handling missing values, outliers, and encoding categorical variables:
* Handle missing values using techniques such as imputation or removing rows with missing data.
* Check for and address outliers if they exist.
* Encode categorical variables into numerical format (e.g., one-hot encoding for binary categories).

I can see that so many features with numerical values are string values in the dataset and not integers or float values. So before moving further, we have to convert these string values to float values:

data["ForecastWindProduction"] = pd.to\_numeric(data["ForecastWindProduction"], errors= 'coerce')

data["SystemLoadEA"] = pd.to\_numeric(data["SystemLoadEA"], errors= 'coerce')

data["SMPEA"] = pd.to\_numeric(data["SMPEA"], errors= 'coerce')

data["ORKTemperature"] = pd.to\_numeric(data["ORKTemperature"], errors= 'coerce') data["ORKWindspeed"] = pd.to\_numeric(data["ORKWindspeed"], errors= 'coerce') data["CO2Intensity"] = pd.to\_numeric(data["CO2Intensity"], errors= 'coerce') data["ActualWindProduction"] = pd.to\_numeric(data["ActualWindProduction"], errors= 'coerce') data["SystemLoadEP2"] = pd.to\_numeric(data["SystemLoadEP2"], errors= 'coerce') data["SMPEP2"] = pd.to\_numeric(data["SMPEP2"], errors= 'coerce')

1

data.isnull().sum()

DateTime 0

Holiday 0

HolidayFlag 0

DayOfWeek 0

WeekOfYear 0

Day 0

Month 0

Year 0

PeriodOfDay 0

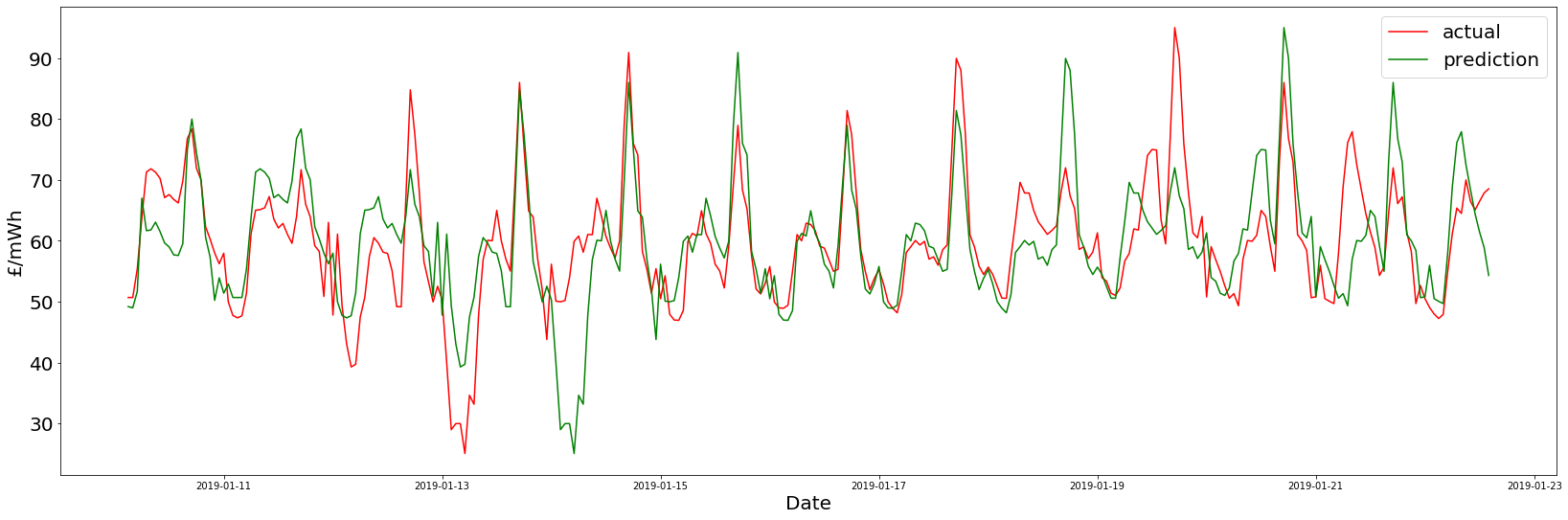
ForecastWindProduction 5

SystemLoadEA 2

SMPEA 2

ORKTemperature 295

ORKWindspeed 299



**Feature Selection:**

* Select relevant features for your diabetes prediction model. You can use various techniques for this, such as:
* Correlation Analysis: Identify features that are strongly correlated with the target variable (diabetes status).
* Feature Importance: Use machine learning models like Random Forest or XGBoost to rank features based on their importance.
* Domain Knowledge: Consult experts or medical literature to understand the relevance of different features.

**Data Splitting:**

* Split the dataset into a training set and a testing set. A common split ratio is 70-80% for training and 20-30% for testing. This allows you to evaluate your model's performance.

**Save Preprocessed Data**:

* Save the preprocessed data into a new file (e.g., a CSV file) so that you can easily access it in the modeling phase.

**Documentation:**

* Document your data preprocessing and feature selection steps. This documentation is essential for reproducibility and future reference.

**Further Data Preparation:**

* Depending on the specific dataset and project requirements, you might need to perform additional steps like feature scaling, dimensionality reduction, or creating new features based on domain knowledge.