Electricity Price Prediction with Machine Learning

TEAM MEMBER

312621243011: Dilli Babu.D

Phase-1 Document Submission

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. So if you want to learn how to predict the price of electricity, then this article is for you. In this article, I will walk you through the task of electricity price prediction with machine learning using Python.

Electricity Price Prediction (Case Study)

Problem statement

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:

Design and Thinking:

1. DateTime: Date and time of the record
2. Holiday: contains the name of the holiday if the day is a national holiday
3. HolidayFlag: contains 1 if it's a bank holiday otherwise 0
4. DayOfWeek: contains values between 0-6 where 0 is Monday
5. WeekOfYear: week of the year
6. Day: Day of the date
7. Month: Month of the date
8. Year: Year of the date
9. PeriodOfDay: half-hour period of the day
10. ForcastWindProduction: forecasted wind production
11. SystemLoadEA forecasted national load
12. SMPEA: forecasted price
13. ORKTemperature: actual temperature measured

14. ORKWindspeed: actual windspeed measured
15. CO2Intensity: actual C02 intensity for the electricity produced
16. ActualWindProduction: actual wind energy production
17. SystemLoadEP2: actual national system load
18. 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.

Electricity Price 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:

Coding:

```
# 1. Data Source: Load your electricity price
dataset import pandas as pd data =
pd.read_csv('Electricity.csv')
```

```
# 2. Data Preprocessing print(data.describe())  #
Summary statistics print(data.isnull().sum())  #
Check for missing values

# Handle missing values (if any)
data.fillna(data.mean(), inplace=True)

# Remove duplicate values (if any)
data = data.drop_duplicates()

# 3. Feature Engineering # Time-based features
data['Date'] = pd.to_datetime(data['Date'])
data['Year'] = data['Date'].dt.year
data['Month'] = data['Date'].dt.month
data['DayOfWeek'] =
data['Date'].dt.dayofweek

# Lagged variables data['ElectricityPrice_Lag1']
= data['ElectricityPrice'].shift(1)
data['ElectricityPrice_Lag7'] =
data['ElectricityPrice'].shift(7)
```

```
# 4. Model Selection from
statsmodels.tsa.arima_model import ARIMA #
Define the ARIMA order (p, d, q) p = 1  # Example
value d = 1  # Example value q = 1  # Example
value

# Create the ARIMA model model =
ARIMA(data['ElectricityPrice'], order=(p, d, q))

# Fit the model to the data model_fit
= model.fit()

# Print the summary of the model
print(model_fit.summary())

# 5. Model Training
# Split the data into training and testing sets
train_size = int(len(data) * 0.8) train, test =
data['ElectricityPrice'][:train_size],
data['ElectricityPrice'][train_size:]
```

```
# Initialize and fit the ARIMA model on the
training data model = ARIMA(train, order=(p, d,
q)) model_fit = model.fit()

# Print the summary of the model
print(model_fit.summary())

# 6. Evaluation
# Make predictions on the test set predictions =
model_fit.forecast(steps=len(test))

# Calculate MAE, MSE, RMSE (import necessary
libraries) from sklearn.metrics import
mean_absolute_error, mean_squared_error
import math

mae = mean_absolute_error(test, predictions)
mse = mean_squared_error(test, predictions)
rmse = math.sqrt(mse)

# Print the evaluation results print(f'Mean
Absolute Error (MAE): {mae}') print(f'Mean
```

Squared Error (MSE): {mse}') print(f'Root Mean

Squared Error (RMSE): {rmse}')

conclusion:

Load and price forecasting shows future trends. Accurate electricity forecasting is the key for the secureness of grid. In this paper author focus on literature in last few years. This paper concerns to provide a review on forecasting in terms of price and load with different ML approaches. The author classifies the papers regarding problems, solutions, constraints, and future challenges. The objective of this paper to provide a survey on price and load and compare their evaluation metrics to check which approach gives best solution.

For future, survey of different deep learning approaches like CNN, ResNet, LSTM and their computational time need to be considered, though computational time is critical in deep learning applications