PREDICTING ELECTRICITY PRICE PREDICTIONUSING DEEPLEARNING

TEAM LEADER

723721243038: PRAJEESH P

Phase2 Submission Document

Project: Electricity Price Prediction

Introduction:

- Electricity price prediction is a fascinating area of study that focuses on forecasting the future prices of electricity.
- It involves using historical data, along with various statistical and machine learning techniques, to analyze and predict the fluctuations in electricity prices.
- This prediction can be valuable for both consumers and energy providers, as it helps them make informed decisions regarding energy usage, pricing strategies, and resource allocation.
- The goal is to develop accurate models that consider factors such as market conditions, weather patterns, supply and demand dynamics, and regulatory policies to anticipate electricity price movements.
- By leveraging advanced analytics and data-driven insights, electricity price prediction can contribute to optimizing energy consumption, reducing costs, and promoting sustainability

Content for Project Phase2:

Consider exploring advanced regression techniques like Gradient Boosting or XGBoost forimprovedPredictionaccuracy.

DataSource:

A good data source for electricity price prediction using deep learning should beAccurate, Complete, Coveringthegeographicareaofinterest, Accessible.

DatasetLink:(https://www.kaggle.com/datasets/chakradharmattapalli/electricity-price-prediction)

Day	Month	Year	PeriodOfDay	ForecastWindProduction	SystemLoadEA	SMPEA	ORKTemperature	ORKWindspeed	CO2Intensity	ActualWindProduction	SystemLoadEP2	SMPEP2
1	11	2011	0	315.31	3388.77	49.26	6	9.3	600.71	356	3159.6	54.32
1	11	2011	1	321.8	3196.66	49.26	6	11.1	605.42	317	2973.01	54.23
1	11	2011	2	328.57	3060.71	49.1	5	11.1	589.97	311	2834	54.23
1	11	2011	3	335.6	2945.56	48.04	6	9.3	585.94	313	2725.99	53.47
1	11	2011	4	342.9	2849.34	33.75	6	11.1	571.52	346	2655.64	39.87
1	11	2011	5	342.97	2810.01	33.75	5	11.1	562.61	342	2585.99	39.87
1	11	2011	6	343.18	2780.52	33.75	5	7.4	545.81	336	2561.7	39.87
1	11	2011	7	343.46	2762.67	33.75	5	9.3	539.38	338	2544.33	39.87
1	11	2011	8	343.88	2766.63	33.75	4	11.1	538.7	347	2549.02	39.87
1	11	2011	9	344.39	2786.8	33.75	4	7.4	540.39	338	2547.15	39.87
1	11	2011	10	345.02	2817.59	33.75	4	7.4	532.3	372	2584.58	39.87
1	11	2011	11	342.23	2895.62	47.42	5	5.6	547.57	361	2641.37	39.87
1	11	2011	12	339.22	3039.67	44.31	5	3.7	556.14	383	2842.19	51.45
1	11	2011	13	335.39	3325.1	45.14	5	3.7	590.34	358	3082.97	51.45
1	11	2011	14	330.95	3661.02	46.25	4	9.3	596.22	402	3372.55	52.82
1	11	2011	15	325.93	4030	52.84	5	3.7	581.52	368	3572.64	53.65
1	11	2011	16	320.91	4306.54	59.44	5	5.6	577.27	361	3852.42	54.21
1	11	2011	17	365.15	4438.05	62.15	6	5.6	568.76	340	4116.03	58.33
1	11	2011	18	410.55	4585.84	61.81	8	7.4	560.79	358	4345.42	58.33
1	11	2011	19	458.56	4723.93	61.88	9	7.4	542.8	339	4427.29	58.33
1	11	2011	20	513.17	4793.6	61.46	?	?	535.37	324	4460.41	58.33
1	11	2011	21	573.36	4829.44	61.28	11	13	532.52	335	4493.22	58.27
1	11	2011	22	636.75	4888.29	61.63	11	22.2	534.34	372	4513.02	58.26
1	11	2011	23	683.59	4936.25	62.12	11	18.5	530.08	415	4490.71	58.26
1	11	2011	24	731.07	4995.51	62.83	11	22.2	517.55	513	4493.73	58.26
1	11	2011	25	780.23	5044.68	60.2	11	20.4	506.83	623	4481.31	58.15
1	11	2011	26	828.09	5018.8	56.25	12	20.4	513.98	683	4408.46	54.74
1	11	2011	27	873.81	4916.93	56.25	11	24.1	518.96	711	4341.14	54.74
1	11	2011	28	920.69	4933.87	56.25	12	22.2	525.69	761	4338.35	54.14
1	11	2011	29	985.09	4978.87	56.25	11	25.9	528.47	750	4294.17	53.63
1	11	2011	30	1044.37	5013.1	56.25	11	22.2	528.17	758	4318.87	53.63
1	11	2011	31	1098.97	5061.1	56.25	11	24.1	513.22	805	4375.62	53.63

Data Collection and Preprocessing:

✓ Importing the dataset: Obtain a comprehensive dataset containing relevant featuressuchasForeCastWind production, SystemLoadEA, SMPEA, ORKTemprature, ORKWindspeed etc.

✓ Data preprocessing: Clean the data by handling missing values, outliers, andcategoricalvariables.Standardizeornormalizenumericalfeatures.

Exploratory Data Analysis (EDA):

✓ Visualize and analyze the dataset to gain insights into the relationships betweenvariables.

✓ Identifycorrelationsandpatternsthatcaninformfeatureselectionandengineering.

✓ Presentvariousdatavisualizationstogaininsightsintothedataset.

✓ Explorecorrelationsbetweenfeaturesandthetargetvariable (electricity price prediction).

✓ DiscussanysignificantfindingsfromtheEDAphasethatinform featureselection.

Feature Engineering:

✓ Createnewfeaturesortransformexistingonestocapturevaluableinformation.

✓ Utilize domain knowledge to engineer features that may impact electricity prices, such asproximitytoschools, college, other organization.

✓ Explaintheprocessofcreatingnewfeaturesortransformingexistingones.

✓ Showcase domain-specific feature engineering, such as proximity scores or compositeindicators.

✓ Emphasizetheimpactofengineeredfeatures onmodelperformance.

Model Evaluation and Selection:

- Splitthedatasetintotrainingandtestingsets.
- Evaluate models using appropriate metrics (e.g., Mean Absolute Error, Mean SquaredError,R-squared)toassesstheirperformance.
- Usecross-validationtechniquestotunehyperparametersandensuremodelstability.
- Compare the results with traditional linear regression models to highlightimprovements.
- Selectthebest-performingmodelforfurtheranalysis.

Model Interpretability:

- Explain how to interpret feature importance from Gradient Boosting and XGBoostmodels.
- Discuss the insights gained from feature importance analysis and their relevance toelectricitypriceprediction.
- Interpret feature importance from ensemble models like Random Forest and GradientBoostingtounderstandthefactors influencinghouseprices.

Deployment and Prediction:

- Deploythechosenregressionmodeltopredict electricityprices.
- Develop a user-friendly interface for users to input property features and receive pricepredictions.
- Will reiterate the impact of these techniques onimprovingtheaccuracyandrobustnessofhousepricepredictions.

- Future Work: We will discuss potential avenues for future work, such as incorporatingadditional data sources (e.g., real-time economic indicators), exploring deep learning modelsfor prediction, or expanding the project into a web application with more features andinteractivites

PROGRAM:

<u>ELECTRICTY PRICE PREDICTION</u>

 # IMPORTING REQUIRED PACKAGES

import numpy as np
import pandas as pd
importmatplotlib.pyplot as plt
importseaborn as sns
importos
df = pd.read_csv("C:/Users/Lenovo/Desktop/Electricity updated.csv", low_memory = False)
df

	HolidayFlag	DayOfWeek	WeekOfYear	Day	Month	Year	PeriodOfDay	ForecastWindProduction	SystemLoadEA	SMPEA	ORKTemperature	ORKWindspe
0	0	1	44	1	11	2011	0	315.31	3388.77	49.26	6	
1	0	1	44	1	11	2011	1	321.8	3196.66	49.26	6	1
2	0	1	44	1	11	2011	2	328.57	3060.71	49.1	5	1
3	0	1	44	1	11	2011	3	335.6	2945.56	48.04	6	
4	0	1	44	1	11	2011	4	342.9	2849.34	33.75	6	1
...	...	...	...	...	...	...	...	...	...	...	...	
38009	1	1	1	31	12	2013	43	1179.14	3932.22	34.51	6	2
38010	1	1	1	31	12	2013	44	1152.01	3821.44	33.83	5	2
38011	1	1	1	31	12	2013	45	1123.67	3724.21	31.75	4	2
38012	1	1	1	31	12	2013	46	1094.24	3638.16	33.83	5	1
38013	1	1	1	31	12	2013	47	1064	3624.25	33.83	5	1

 # CHECKING FOR NULL VALUES

df1 = df.isnull()
df1

	HolidayFlag	DayOfWeek	WeekOfYear	Day	Month	Year	PeriodOfDay	ForecastWindProduction	SystemLoadEA	SMPEA	ORKTemperature	ORKWinds
0	False	False	False	False	False	False	False	False	False	False	False	
1	False	False	False	False	False	False	False	False	False	False	False	
2	False	False	False	False	False	False	False	False	False	False	False	
3	False	False	False	False	False	False	False	False	False	False	False	
4	False	False	False	False	False	False	False	False	False	False	False	
...	...	...	...	...	...	...	...	...	...	...	...	
38009	False	False	False	False	False	False	False	False	False	False	False	
38010	False	False	False	False	False	False	False	False	False	False	False	
38011	False	False	False	False	False	False	False	False	False	False	False	
38012	False	False	False	False	False	False	False	False	False	False	False	
38013	False	False	False	False	False	False	False	False	False	False	False	

 # ADDING NULL VALUES

df1 = df.isnull().sum()
df1

```
Out[5]: HolidayFlag               0
        DayOfWeek                 0
        WeekOfYear                0
        Day                       0
        Month                     0
        Year                      0
        PeriodOfDay               0
        ForecastWindProduction    0
        SystemLoadEA              0
        SMPEA                     0
        ORKTemperature            0
        ORKWindspeed              0
        CO2Intensity              0
        ActualWindProduction      0
        SystemLoadEP2             0
        SMPEP2                    0
        SystemLoadEP2.1           0
        SMPEP2.1                  0
        dtype: int64
```

 # CHECKING THE DATA TYPES

df2 = df.dtypes

df2

```
Out[6]: HolidayFlag                int64
        DayOfWeek                  int64
        WeekOfYear                 int64
        Day                        int64
        Month                      int64
        Year                       int64
        PeriodOfDay                int64
        ForecastWindProduction    object
        SystemLoadEA              object
        SMPEA                     object
        ORKTemperature            object
        ORKWindspeed              object
        CO2Intensity              object
        ActualWindProduction      object
        SystemLoadEP2             object
        SMPEP2                    object
        SystemLoadEP2.1           object
        SMPEP2.1                  object
        dtype: object
```

 # REPLACING SPECIAL CHARACTER WITH NULL VALUE

df.replace(to_replace='?',value='0',inplace=True)

df

 # CHECKING FOR ANY SPECIAL CHARACTERS IN FEATURES

df4 = df[df['ORKWindspeed']=='?']

df4

 # VISUALIZATION FOR ABOVE DATAS

importseaborn as sns

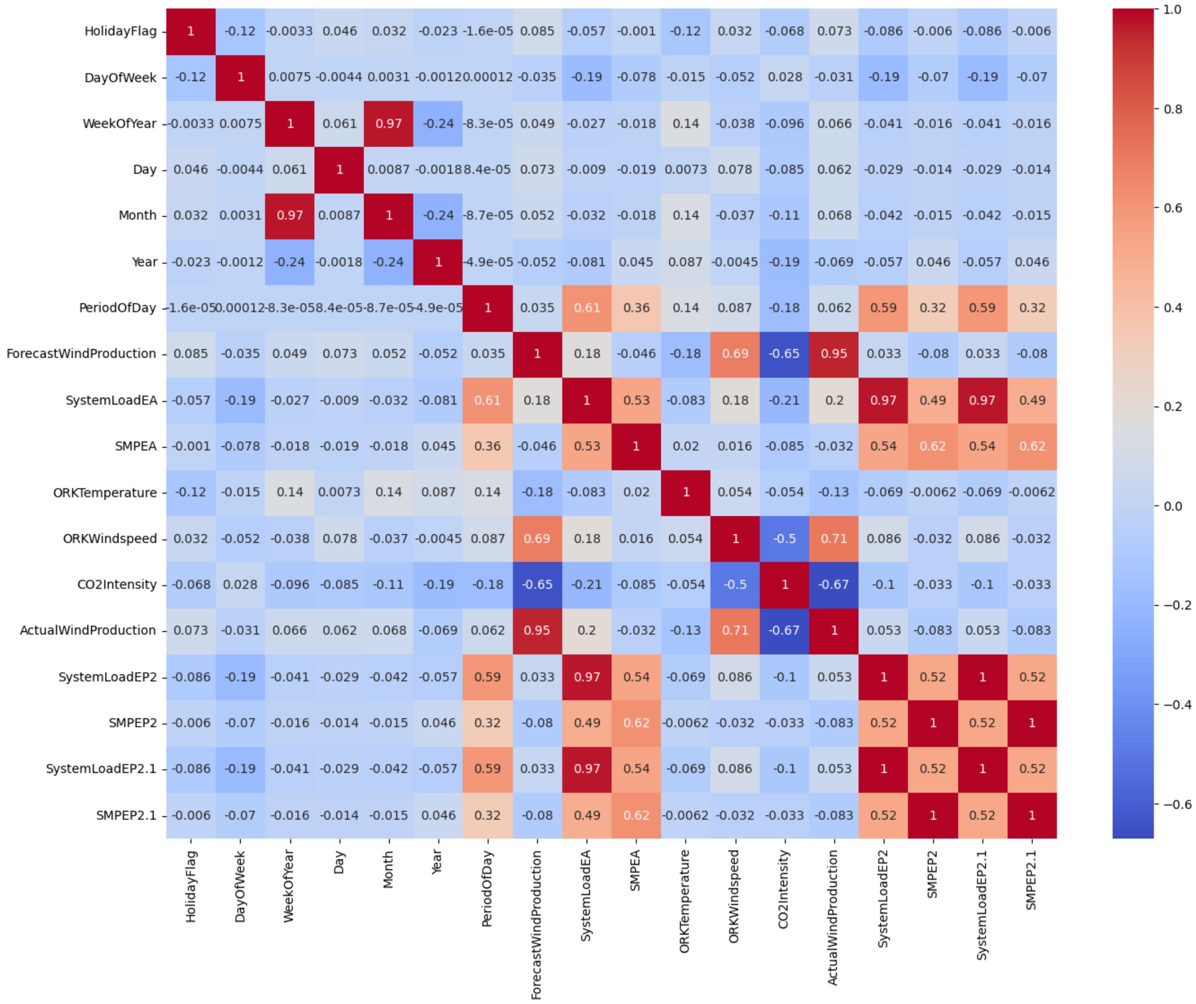
importmatplotlib.pyplot as plt

correlations = df.corr(method='pearson')

plt.figure(figsize=(16, 12))

sns.heatmap(correlations, cmap="coolwarm", annot=True)

plt.show()




```
# COVERTING THE FILE AS CSV
df.to_csv(r'E:\file3.csv')
df
# IMPORTING THE FILE
dtf1=pd.read_csv("E:/file3.csv")
dtf1
# CONVERTING THE DATATYPES OF FEATURES
dtf1['ForecastWindProduction'] = dtf1['ForecastWindProduction'].astype(float)
dtf1['SystemLoadEA'] = dtf1['SystemLoadEA'].astype(float)
dtf1['SMPEA'] = dtf1['SMPEA'].astype(float)
dtf1

# CHECKING THE DATA TYPES
dtf1.dtypes
```

```
Out[17]: Unnamed: 0                  int64
         HolidayFlag                 int64
         DayOfWeek                   int64
         WeekOfYear                  int64
         Day                         int64
         Month                       int64
         Year                        int64
         PeriodOfDay                 int64
         ForecastWindProduction    float64
         SystemLoadEA              float64
         SMPEA                     float64
         ORKTemperature              int64
         ORKWindspeed              float64
         CO2Intensity              float64
         ActualWindProduction        int64
         SystemLoadEP2             float64
         SMPEP2                    float64
         SystemLoadEP2.1           float64
         SMPEP2.1                  float64
         dtype: object
```

 # IMPLEMENTING LSTM MODEL

 # IMPORTING THE REQUIRED PACKAGES

```
import numpy as np
import pandas as pd
from sklearn.preprocessing import MinMaxScaler
from sklearn.model_selection import train_test_split
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense
import matplotlib.pyplot as plt
```

 # LOAD THE DATASET

```
data = pd.read_csv("C:/Users/STUDENT/Desktop/Electricity updated.csv" , low_memory = False)
```

 # SELECT THE FEATURES AND THE TARGET VARIABLES

```
X = data[['ForecastWindProduction','SystemLoadEA' ,'SMPEA' , 'ORKTemperature', 'ORKWindspeed', 'CO2Intensity']].values
y = data['SystemLoadEP2.1'].values
```

 # NORMALIZE THE FEATURE AND THE TARGET VARIABLES

```
scaler_X = MinMaxScaler()
X = scaler_X.fit_transform(X)
scaler_y = MinMaxScaler()
y = scaler_y.fit_transform(y.reshape(-1, 1))
```

SPLIT THE DATAS INTO TRAINING AND TESTING
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# BUILD THE FEEDFORWARD NEURAL NETWORK
model = Sequential()
model.add(Dense(64, input_dim=X.shape[1], activation='relu'))
model.add(Dense(32, activation='relu'))
model.add(Dense(1, activation='linear'))
model.compile(loss='mean_squared_error', optimizer='adam')

# TRAIN THE MODEL
history = model.fit(X_train, y_train, epochs=5, batch_size=32, validation_data=(X_test, y_test), verbose=2)

```
...
Epoch 1/5
951/951 - 3s - loss: 0.0064 - val_loss: 5.9933e-04 - 3s/epoch - 3ms/step
Epoch 2/5
951/951 - 2s - loss: 6.2950e-04 - val_loss: 5.9561e-04 - 2s/epoch - 2ms/step
Epoch 3/5
951/951 - 1s - loss: 6.3446e-04 - val_loss: 6.1864e-04 - 1s/epoch - 1ms/step
Epoch 4/5
951/951 - 2s - loss: 6.3290e-04 - val_loss: 5.7935e-04 - 2s/epoch - 2ms/step
Epoch 5/5
951/951 - 1s - loss: 6.4003e-04 - val_loss: 5.7708e-04 - 1s/epoch - 1ms/step
```

# VISUALIZE THE TRAINING PROCESS
plt.figure(figsize=(12, 6))
plt.plot(history.history['loss'], label='Training Loss')
plt.plot(history.history['val_loss'], label='Validation Loss')
plt.title('Model Loss')
plt.xlabel('Epoch')
plt.ylabel('Loss')
plt.legend()
plt.show()