ELECTRICITY PRICE PREDICTION

INTRODUCTION:

In a world driven by data, accurate electricity price forecasts are crucial for efficient resource allocation. In this report, we showcase how Python, coupled with data science techniques, can harness historical data, weather patterns, and market dynamics to predict electricity prices. Join us as we unveil the power of data-driven insights in the energy sector.

DATA COLLECTION:

The success of our electricity price prediction project hinges on the quality and diversity of data sources harnessed. We meticulously curated multiple datasets to gain comprehensive insights into the factors influencing electricity prices. Below, we outline the key data sources used:

- ELECTRICITY PRICE DATA:
  We sourced historical electricity price data from the [mention data source], spanning from [start date] to [end date]. The data is reported at a [frequency, e.g., hourly] resolution, capturing the nuances of price fluctuations. Prior to analysis, the dataset underwent rigorous cleaning and preprocessing to handle missing values and ensure data integrity.
- WEATHER DATA:
  Weather conditions are known to significantly impact electricity consumption and generation. We acquired weather data from [mention data source] for the same time period as our electricity price data. Parameters including [list parameters, e.g., temperature, wind speed, sunlight hours] were collected at an [e.g., hourly] frequency. Preprocessing steps included [describe any transformations or aggregations] to make the data suitable for analysis.
- ECONOMIC INDICATORS:
  To account for macroeconomic influences, we integrated economic indicators such as [list economic indicators, e.g., GDP growth rate, inflation rate] into our dataset. These indicators were collected on a [e.g., quarterly] basis from [start date] to [end date] and were incorporated as additional features in our analysis.
- MARKET DATA:
  These indicators were collected on a [e.g., quarterly] basis from [start date] to [end date] and were incorporated as additional features in our analysis. The data spans from [start date] to [end date] and was used to enrich our modeling capabilities.
  The thoroughness of our data collection process ensures that our models are well-informed, capable of capturing the intricate dynamics of electricity pricing. These data sources, each serving a distinct purpose, form the bedrock of our project's predictive power.

EXPLORATORY DATA ANALYSIS(EDA):

In the EDA phase, we unveiled critical insights into our dataset, shedding light on the underlying patterns and relationships that influence electricity prices. Key highlights include:

- SUMMARY STATISTICS:
  We provided an overview of the central tendencies and variability of our data, revealing the price distribution and its statistical properties.
- DATA VISUALIZATIONS:
  Through a range of visualizations using Python libraries like Matplotlib and Seaborn, we presented trends, seasonality, and outliers. Visualizations include time series plots, histograms, and heatmaps, allowing us to discern patterns and correlations.
- CORRELATION ANALYSIS:
  We explored the relationships between electricity prices and external factors, such as weather parameters and economic indicators. This analysis provided initial insights into potential drivers of price fluctuations.
  Our EDA not only laid the foundation for feature engineering but also enabled a deeper understanding of the dynamics at play in the electricity market.

```python
def exploratory_data_analysis(data):
    """
    Perform exploratory data analysis (EDA).

    Parameters:
    - data: Data for analysis (DataFrame).

    Returns:
    - Summary statistics, visualizations, and insights.
    """
    # Calculate summary statistics
    summary_stats = data.describe()

    # Create EDA visualizations
    plt.figure(figsize=(12, 6))
    # Histogram of 'Price'
    plt.subplot(1, 2, 1)
    sns.histplot(data['Price'], bins=30, kde=True)
    plt.title('Price Distribution')
    # Correlation heatmap
    plt.subplot(1, 2, 2)
    corr_matrix = data.corr()
    sns.heatmap(corr_matrix, annot=True, cmap='coolwarm', linewidths=0.5)
```

```
    plt.title('Correlation Heatmap')
    plt.tight_layout()
    plt.show()

    return summary_stats
```

FUTURE:

This project's insights and predictive models have the potential to revolutionize decision-making in the energy sector. By offering accurate electricity price forecasts, it will empower utilities and consumers to optimize consumption, reduce costs, and enhance energy efficiency. Investors and policymakers can make more informed choices, contributing to sustainable and resilient energy systems. Ultimately, this project shapes a future where energy resource allocation is data-driven, fostering a more reliable, affordable, and sustainable energy landscape.

MODELING:

In our project, we employed a combination of time series analysis and machine learning algorithms for electricity price prediction. Specifically, we utilized:

- ARIMA (AutoRegressive Integrated Moving Average): ARIMA models capture time-dependent patterns in the data, including seasonality and trends. This approach is well-suited for modeling historical price data.
- XGBoost (Extreme Gradient Boosting):XGBoost, a powerful gradient boosting algorithm, was employed to account for complex non-linear relationships between electricity prices and external factors, enhancing prediction accuracy.
- LSTM (Long Short-Term Memory):We harnessed LSTM, a type of recurrent neural network (RNN), to capture sequential dependencies in time series data. This deep learning algorithm excels in modeling dynamic and evolving price patterns.
  By combining these algorithms, we aimed to leverage their respective strengths to provide robust and accurate electricity price forecasts.

MODEL EVALUATION:

To assess the model's performance, we employed a rigorous evaluation process. We divided the dataset into training and validation sets, ensuring temporal order in the data split. The evaluation criteria included Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), and R-squared (R2) to measure predictive accuracy. Visualizations comparing actual vs. predicted prices were also employed to gain insights into model performance and potential areas for improvement. This comprehensive evaluation approach allowed us to determine the model's effectiveness in capturing electricity price fluctuations.

```
def evaluate_model(model, X_val, y_val):
```

```
    """
    Evaluate the performance of a predictive model.

    Parameters:
    - model: Trained machine learning model.
    - X_val: Validation dataset features.
    - y_val: Validation dataset target values.

    Returns:
    - Dictionary of evaluation metrics (e.g., MAE, RMSE, R-squared).
    """
    # Make predictions on the validation dataset
    y_pred = model.predict(X_val)

    # Calculate evaluation metrics
    mae = mean_absolute_error(y_val, y_pred)
    rmse = (mean_squared_error(y_val, y_pred))**0.5
    r2 = r2_score(y_val, y_pred)

    # Create a dictionary of evaluation metrics
    evaluation_results = {
        'MAE': mae,
        'RMSE': rmse,
        'R-squared': r2
    }

    return evaluation_results
```

RESULTS AND DISCUSSION:

Our results showcased the model's ability to provide accurate and valuable electricity price predictions. The model exhibited a strong performance with low MAE and RMSE, indicating minimal prediction errors. Additionally, the R-squared (R2) score highlighted the model's capability to explain a significant portion of the variance in electricity prices. Visualizations depicted a close alignment between actual and predicted price trends, affirming the model's effectiveness in capturing market dynamics. These outcomes underscore the practical utility of our approach for stakeholders in the energy sector.

DEPLOYMENT:

We deployed our model for real-time electricity price predictions using a Flask web application hosted on a cloud server. This application allows users to input relevant data, such as weather conditions and economic indicators, and receive instantaneous price forecasts. Additionally, we containerized the application using Docker to ensure portability and scalability, enabling easy deployment across various environments and platforms. This deployment strategy ensures that our predictive model is accessible and adaptable to the evolving needs of the energy industry.

CODE:

Index.html

```html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Electricity Price Prediction</title>
</head>
<body>
    <h1>Welcome to Electricity Price Prediction</h1>
    <form action="/predict" method="post">
        <input type="submit" value="Predict Electricity Prices">
    </form>
</body>
</html>
```

App.py

```python
from flask import Flask, render_template, request, send_file
import pandas as pd
from sklearn.ensemble import RandomForestRegressor
from sklearn.model_selection import train_test_split
from sklearn.metrics import mean_squared_error
import matplotlib.pyplot as plt
from io import BytesIO
import base64

app = Flask(__name__)

def plot_prices(y_test, y_pred):
    plt.figure(figsize=(8, 6))
    plt.plot(y_test.values, label='Actual Prices')
    plt.plot(y_pred, label='Predicted Prices')
    plt.title('Electricity Price Prediction')
    plt.xlabel('Data Points')
    plt.ylabel('Electricity Prices')
    plt.legend()

    img = BytesIO()
    plt.savefig(img, format='png')
    img.seek(0)
```

```python
    # Convert the image to base64 representation
    plot_base64 = base64.b64encode(img.getvalue()).decode()
    return plot_base64

@app.route('/')
def home():
    return render_template('index.html')

@app.route('/predict', methods=['POST'])
def predict():
    data = pd.read_csv('electricity_data.csv')
    data = data.select_dtypes(exclude=['object'])
    features = data.drop('Electricity_Price', axis=1)
    target = data['Electricity_Price']
    X_train, X_test, y_train, y_test = train_test_split(features, target,
test_size=0.2, random_state=42)

    model = RandomForestRegressor()
    model.fit(X_train, y_train)

    y_pred = model.predict(X_test)
    mse = mean_squared_error(y_test, y_pred)

    plot = plot_prices(y_test, y_pred)

    return render_template('result.html', mse=mse,
predicted_prices=list(y_pred[:10]), plot=plot)

if __name__ == '__main__':
    app.run(debug=True)
```

result.html

```html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Prediction Results</title>
</head>
<body>
    <h1>Prediction Results</h1>
    <p>Mean Squared Error: {{ mse }}</p>
    <p>Predicted Electricity Prices: {{ predicted_prices }}</p>
    <img src="data:image/png;base64,{{ plot }}" alt="Prediction Graph">
```

```
</body>
</html>
```

Electricity_price.csv

Date	Electricity_Price	System_Loads	Weather_Variables	Fuel_Costs	Reserve_Margin	Scheduled_Maintenance	Forced_Outages
########	49.96321	934	0.091002	39.25933	24	0	1
########	96.05714	676	0.726397	93.04006	10	0	0
########	78.55952	516	0.547446	78.38553	16	1	0
########	67.89268	267	0.45091	77.82287	-45	1	1
########	32.48149	941	0.910471	87.58087	-42	0	0
########	32.47956	142	0.297959	67.07632	-45	1	0
########	24.64669	655	0.523602	87.05381	22	0	0
########	89.29409	384	0.697642	64.86635	44	0	0
########	68.0892	496	0.796472	34.70228	-19	0	0
########	76.64581	111	0.459347	58.36737	-10	1	0
########	21.64676	706	0.842091	64.82254	25	1	0
########	97.59279	401	0.768918	80.41418	-43	0	0
########	86.59541	997	0.066236	37.46051	21	0	1
########	36.98713	352	0.045861	39.8013	-1	0	1
########	34.546	598	0.620806	48.45787	41	0	1
########	34.67236	853	0.347413	48.65489	11	0	0
########	44.33938	134	0.209131	81.25292	33	1	0
########	61.98051	826	0.57965	47.55115	11	0	1
########	54.5556	948	0.341563	74.31686	-44	1	1
########	43.29833	189	0.537263	64.59917	24	0	1

##### ###	68.94823	875	0.460119	70.130 61	33	1	0
##### ###	31.15951	704	0.584766	88.667 49	29	1	1
##### ###	43.37157	701	0.4003	58.306 71	-47	1	1
##### ###	49.30895	517	0.697668	83.963 8	-45	1	0
##### ###	56.4856	214	0.180067	59.542 52	7	1	1
##### ###	82.81408	716	0.696501	54.771	-29	1	0
##### ###	35.9739	295	0.411661	97.103 9	-25	0	0
##### ###	61.13876	925	0.874318	42.905 96	-48	1	0
##### ###	67.39317	600	0.515236	36.323 59	-10	0	1
##### ###	23.71603	725	0.97311	83.969 01	9	1	1
##### ###	68.60359	592	0.601935	34.777 48	-37	1	0
##### ###	33.64193	174	0.223849	88.595 53	24	0	1
##### ###	25.20413	512	0.821791	60.603 54	31	1	1
##### ###	95.91084	475	0.345083	94.085 96	-39	0	1
##### ###	97.25056	519	0.347619	80.507 3	36	0	1
##### ###	84.67179	828	0.031805	72.728 33	-39	0	0
##### ###	44.3691	376	0.548715	96.451 48	-38	0	1
##### ###	27.81377	860	0.534424	57.765 46	-26	1	0
##### ###	74.73864	775	0.355991	96.786 09	-6	1	0
##### ###	55.2122	493	0.894217	39.450 09	-32	0	1
##### ###	29.76306	968	0.128748	63.862 85	4	1	0
##### ###	59.61415	556	0.3301	48.893 79	49	1	1
##### ###	22.75108	291	0.321583	67.788 99	-6	0	1
##### ###	92.74563	838	0.092291	41.340 97	-43	1	0

########	40.7024	788	0.481145	88.90708	42	0	0
########	73.00178	198	0.687785	89.24855	2	0	0
########	44.93689	647	0.511657	96.86654	4	1	0
########	61.60544	195	0.156978	40.82705	-19	1	1
########	63.73682	763	0.377286	73.63328	0	1	1
########	34.78836	762	0.002595	63.19661	-7	0	1
########	97.56677	289	0.868301	54.10686	31	0	0
########	82.01063	835	0.084517	54.60781	19	1	0
########	95.15992	136	0.597278	58.89555	-33	1	0
########	91.58619	879	0.986257	79.34479	32	1	1
########	67.832	468	0.536591	71.88002	-29	1	0
########	93.74994	794	0.924042	62.17771	-14	0	1
########	27.0794	624	0.236117	35.31534	45	1	0
########	35.67863	378	0.759955	35.44357	5	0	1
########	23.61818	316	0.531266	30.16376	8	0	1
########	46.02643	966	0.720516	97.73896	-48	1	0
########	51.09418	972	0.062341	30.36777	-23	1	0
########	41.70792	897	0.147739	37.21656	23	1	1
########	86.299	372	0.133117	52.02176	-16	0	1
########	48.54027	980	0.687166	86.52012	10	1	0
########	42.47476	161	0.844441	97.34692	42	1	0
########	63.41569	695	0.749616	85.30095	3	0	1
########	31.27394	979	0.030472	78.33922	12	1	0
########	84.17576	828	0.867215	66.49105	28	0	1

##### ###	25.96405	441	0.354147	36.069 36	46	1	1
##### ###	98.95095	496	0.397164	97.071 96	-44	0	0
##### ###	81.77958	798	0.104869	83.087 47	25	1	0
##### ###	35.89725	118	0.737405	74.726 46	-2	1	0
##### ###	20.44177	276	0.182284	83.132 51	43	0	0
##### ###	85.23691	711	0.563965	80.687 23	34	0	0
##### ###	76.54859	495	0.84071	74.606 21	-20	1	1
##### ###	78.32057	544	0.089204	98.640 38	7	0	1
##### ###	81.70163	332	0.535336	93.235 47	10	0	0
##### ###	25.92357	175	0.233216	75.266 54	-27	0	0
##### ###	48.67726	364	0.342927	78.522 26	-31	0	0
##### ###	29.26952	554	0.47397	33.611 71	-14	1	1
##### ###	89.04827	895	0.355104	76.894 22	-47	1	0
##### ###	69.86385	817	0.648823	33.091 58	41	0	0
##### ###	46.47184	834	0.479582	71.178 46	-31	0	1
##### ###	25.08467	483	0.584199	99.954 74	9	0	1
##### ###	44.87859	663	0.736822	68.906 46	-27	0	0
##### ###	46.01467	950	0.557742	63.124 71	-1	1	0
##### ###	78.36849	605	0.586535	51.899 84	-2	1	0
##### ###	71.0046	466	0.564459	38.436 58	0	0	1
##### ###	90.97702	243	0.378773	81.171 66	45	0	0
##### ###	57.77719	984	0.337447	43.481 93	-15	0	0
##### ###	29.56754	168	0.899647	38.069 74	47	1	0
##### ###	77.05958	198	0.607555	59.523 39	49	0	1

########	80.8628	495	0.244353	85.67393	15		1	1
########	64.90218	124	0.498248	82.15452	28		0	0
########	81.67737	990	0.330348	33.84178	-17		1	0
########	59.50365	568	0.933692	61.86006	-15		0	1
########	61.81863	583	0.007534	66.57042	-45		1	1
########	54.20328	664	0.225333	75.11827	-12		0	1
########	22.03353	250	0.365357	75.51171	-9		0	1
########	28.63131	243	0.48781	55.52881	7		0	1
########	22.51433	668	0.850818	69.2474	-39		1	1
########	70.91283	138	0.087888	63.67433	-48		0	1
########	45.14848	208	0.805865	91.94988	-33		1	1
########	60.68566	792	0.055653	67.11856	-11		1	1
########	92.60532	141	0.842314	60.87225	-30		0	0
########	39.94338	285	0.051635	58.31398	-36		0	0
########	52.83063	497	0.018242	70.06994	46		0	0
########	80.44409	322	0.696961	86.27839	48		1	1
########	38.30385	733	0.997256	67.6881	-25		0	0
########	26.15839	232	0.89661	76.31586	-43		1	1
########	43.18012	262	0.575998	81.6499	49		1	1
########	32.8977	314	0.917396	66.41177	43		0	0
########	94.37581	832	0.0053	59.87719	36		0	1
########	84.64963	334	0.975067	91.36658	-9		0	0
########	70.6723	942	0.490749	59.22543	-7		0	0
########	89.71685	757	0.722896	62.336	-37		0	0

##### ###	84.29377	850	0.820861	99.323 89	27	0	1
##### ###	34.9256	687	0.718457	30.016 63	-49	1	0
##### ###	91.40472	108	0.535037	42.983 1	-14	1	1
##### ###	63.14738	173	0.476619	56.877 47	18	1	0
##### ###	84.59521	591	0.838583	95.281 29	-13	0	0
##### ###	91.6873	352	0.205078	34.902 08	-10	1	0
##### ###	45.44028	329	0.967994	30.663 98	11	1	1
##### ###	28.80415	618	0.710952	33.687 87	9	1	1
##### ###	38.23481	273	0.199507	36.198 04	-37	1	1
##### ###	54.16862	752	0.736247	32.631 77	38	1	0
##### ###	85.44118	267	0.52984	63.584 69	33	0	0
##### ###	88.85845	269	0.70723	70.653	-49	0	1
##### ###	20.55617	492	0.76778	49.018 09	19	1	1
##### ###	60.85978	894	0.08729	57.880 12	-34	1	1
##### ###	53.39288	733	0.506104	36.418 95	38	1	1
##### ###	37.76862	293	0.932014	53.545 93	-23	0	1
##### ###	29.58923	616	0.320642	66.576 34	15	0	1
##### ###	47.00921	128	0.593883	81.259 03	-16	1	0
##### ###	95.43278	264	0.36923	30.234 25	-46	0	1
##### ###	45.85623	521	0.454268	62.711 75	14	0	0
##### ###	61.50325	438	0.548603	50.790 2	-11	1	0
##### ###	76.24152	747	0.548922	89.697 53	25	0	1
##### ###	49.09037	595	0.20173	80.088 73	5	0	1
##### ###	97.74257	464	0.684572	71.237 6	-15	1	0

##### ###	96.99578	932	0.087868	49.408 6	-6	0	1
##### ###	40.14258	441	0.138825	93.723 74	-28	0	1
##### ###	59.77988	599	0.002711	33.157 05	-4	1	0
##### ###	44.07026	756	0.116696	37.662 73	-20	0	0
##### ###	42.78724	610	0.473166	57.414 11	-40	0	0
##### ###	22.95096	426	0.606102	38.710 63	-41	1	1
##### ###	68.76515	316	0.794289	96.949 1	16	1	0
##### ###	60.21432	400	0.106699	85.859 25	-37	1	1
##### ###	24.1183	231	0.850727	48.128 79	29	0	0
##### ###	42.29172	903	0.745975	71.198 8	23	1	0
##### ###	92.66127	169	0.408518	98.763 14	42	0	0
##### ###	39.16495	351	0.932938	91.893 93	-41	1	1
##### ###	31.59159	514	0.990929	72.055 89	25	0	1
##### ###	59.15622	886	0.205002	93.260 68	43	1	0
##### ###	98.85204	544	0.379229	99.242 94	-50	1	0
##### ###	39.36442	975	0.926449	82.034 7	-46	1	0
##### ###	73.77084	281	0.721597	34.549 05	-31	1	1
##### ###	80.92957	266	0.048095	58.110 62	-43	0	0
##### ###	39.011	190	0.781514	88.647 8	-8	1	1
##### ###	78.25731	813	0.827941	46.139 84	25	1	0
##### ###	49.42265	957	0.750502	88.144 98	-50	0	0
##### ###	70.58447	630	0.799537	38.423 71	6	1	0
##### ###	70.68238	138	0.825133	33.286 18	13	0	0
##### ###	62.86197	225	0.186405	56.886 34	33	1	0

##### ###	27.22318	550	0.235692	32.571 24	-23	1	0
##### ###	86.8242	272	0.633759	97.008 92	-14	1	1
##### ###	45.66241	752	0.907866	87.834 4	-4	0	0
##### ###	34.92148	853	0.316197	86.053 88	25	1	1
##### ###	23.26201	319	0.588307	74.084 45	-14	1	0
##### ###	67.27144	737	0.682982	45.142 68	-2	1	0
##### ###	74.20515	157	0.451949	66.278 23	-7	0	1
##### ###	21.32703	759	0.713782	71.809 19	5	0	0
##### ###	60.96744	575	0.899667	66.752 21	-23	1	1
##### ###	38.11966	555	0.624102	48.216 34	28	0	1
##### ###	71.61382	928	0.539781	66.067 35	-45	1	0
##### ###	33.94931	994	0.438745	64.383 04	10	1	1
##### ###	75.27502	460	0.577486	99.743 5	-31	1	1
##### ###	50.93883	100	0.355362	92.820 53	27	1	0
##### ###	94.9384	486	0.391482	62.436 65	-49	1	1
##### ###	31.00168	447	0.531857	73.605 62	22	1	0
##### ###	47.28531	289	0.066619	82.351 01	-14	1	0
##### ###	29.07788	604	0.229025	32.440 56	-31	0	0
##### ###	93.97549	290	0.542849	92.644 5	-25	0	0
##### ###	90.18715	607	0.431528	90.208 88	-22	0	1
##### ###	40.63533	468	0.332816	62.020 25	-50	0	0
##### ###	72.79872	508	0.730549	57.381 88	-16	0	0
##### ###	85.37778	923	0.693718	49.107 16	19	0	1
##### ###	64.41606	216	0.166731	63.367 38	8	0	0

##### ###	62.37205	233	0.878629	33.106 37	18	1	0
##### ###	39.34818	157	0.495424	89.653 55	-18	1	1
##### ###	27.44822	655	0.741461	32.479 57	0	1	1
##### ###	91.77726	784	0.573151	53.033 6	27	1	1
##### ###	92.03344	771	0.997693	95.211 26	-39	0	1
##### ###	70.64812	272	0.752403	81.464 89	-31	0	1
##### ###	47.12238	928	0.70698	49.697 37	2	1	1
##### ###	47.93677	914	0.778572	47.685 44	37	1	1
##### ###	78.07645	248	0.143128	30.355 75	-8	0	0
##### ###	91.76882	179	0.204545	78.7	47	1	1
##### ###	90.96691	985	0.714064	33.414 85	38	1	1
##### ###	82.39004	312	0.493981	67.282 18	-1	1	1
##### ###	71.36253	302	0.754635	97.348 54	-22	1	0
##### ###	26.7312	863	0.102916	37.662 82	6	0	0
##### ###	32.9303	328	0.536481	92.613 6	-12	1	1
##### ###	91.88434	775	0.378822	99.403 27	44	1	1
##### ###	68.51432	326	0.457	34.306 19	5	0	1
##### ###	20.73576	758	0.603958	91.803 44	-8	0	0
##### ###	28.11772	631	0.502288	66.125 53	-1	1	0
##### ###	73.08014	540	0.539861	93.455 96	-28	0	1
##### ###	20.40493	501	0.486357	69.992 33	-7	1	0
##### ###	32.86464	146	0.408955	76.347 84	-13	1	1
##### ###	63.8987	332	0.771882	68.064 52	33	0	0
##### ###	75.35162	404	0.012203	90.514 5	-28	1	0

##### ###	72.1569	625	0.598443	81.373 69	-6	1	1
##### ###	37.94154	242	0.565508	66.488 22	1	1	1
##### ###	76.97434	514	0.716179	90.771 26	29	1	1
##### ###	38.97993	612	0.599029	47.169 99	-1	0	1
##### ###	46.03198	472	0.826799	41.013 85	-43	0	0
##### ###	79.71931	665	0.959075	41.228 01	34	1	0
##### ###	71.97063	985	0.342524	53.380 62	-50	0	0
##### ###	87.93787	358	0.227351	46.708 3	-43	1	1
##### ###	72.60903	755	0.423597	95.106 32	-22	1	0
##### ###	65.46469	570	0.28793	37.408 39	45	1	1
##### ###	27.49398	111	0.61495	78.929 35	-46	1	0
##### ###	49.41726	429	0.911852	43.245 51	35	1	1
##### ###	41.21619	835	0.139116	91.714 6	-4	0	1
##### ###	39.51917	883	0.100795	97.292 07	-17	1	0
##### ###	97.84084	457	0.256016	62.582 9	14	0	1
##### ###	51.44782	507	0.726096	91.535	4	0	1
##### ###	91.36372	767	0.592963	30.360 16	39	1	0
##### ###	70.49109	472	0.102213	87.396 28	-8	1	1
##### ###	83.5849	107	0.918751	46.003 76	23	1	1
##### ###	60.21097	221	0.790085	93.609 09	-21	1	1
##### ###	66.15231	447	0.023023	30.078 41	31	1	0
##### ###	59.40142	775	0.651367	68.932 72	-34	0	0
##### ###	35.61944	189	0.771247	82.964 51	-25	1	1
##### ###	77.79617	747	0.374435	48.108 18	-43	1	0

##### ###	42.46179	797	0.068922	43.519 3	-4	0	1
##### ###	21.94528	415	0.077321	35.886 88	25	0	1
##### ###	71.63778	277	0.104247	52.850 35	40	1	1
##### ###	34.16885	639	0.84044	69.189 1	21	0	1
##### ###	95.23669	831	0.910686	38.109 07	38	1	1
##### ###	96.31429	968	0.122811	56.368 68	-5	0	0
##### ###	93.18915	140	0.235906	30.635 07	17	0	1
##### ###	49.6127	839	0.165521	87.307 56	-18	1	0
##### ###	21.23653	803	0.186321	38.991 41	-50	0	0
##### ###	94.26549	601	0.837491	49.834 01	-32	0	1
##### ###	54.25473	244	0.332146	31.289 94	-20	0	1
##### ###	97.33239	300	0.311444	60.256 85	29	0	0
##### ###	97.0896	823	0.227396	88.808 51	-37	0	0
##### ###	88.24076	560	0.607894	57.450 58	2	0	0
##### ###	43.55591	831	0.379306	42.564 81	-3	1	1
##### ###	50.80782	851	0.74425	82.758 92	18	1	0
##### ###	88.09093	657	0.20558	85.216 49	39	0	1
##### ###	45.35376	646	0.78779	42.162 74	11	1	1
##### ###	33.55942	352	0.603722	83.959 79	41	1	1
##### ###	64.5441	489	0.114267	96.310 85	15	1	0
##### ###	94.89238	693	0.414505	41.646 43	13	0	1
##### ###	75.68238	982	0.863518	83.968 9	19	1	1
##### ###	65.60489	355	0.92296	79.600 04	12	0	1
##### ###	27.77412	808	0.4657	63.150 17	22	1	1

##### ###	69.20058	914	0.480837	34.054 43	-18	0	1
##### ###	99.20431	549	0.918455	35.228 61	-10	0	0
##### ###	31.20672	109	0.587068	89.669 03	38	1	1
##### ###	61.46637	923	0.032847	99.310 39	-3	1	1
##### ###	90.18985	897	0.912749	77.323 66	-38	0	0
##### ###	79.26149	341	0.248226	61.217 08	22	1	1
##### ###	75.76126	350	0.577632	75.330 16	26	1	0
##### ###	76.19873	976	0.165513	71.607 71	-20	1	1
##### ###	48.75929	104	0.033879	56.852 86	24	0	0
##### ###	43.48735	218	0.311473	34.593 83	17	0	0
##### ###	84.74889	900	0.780496	87.505 91	47	1	1
##### ###	84.80907	473	0.277587	56.028 35	-47	1	1
##### ###	89.36579	164	0.220097	87.315	-32	1	1
##### ###	93.05924	245	0.212681	47.675 61	19	0	0
##### ###	60.90739	323	0.515174	30.072 06	47	0	0
##### ###	60.1213	338	0.975541	41.125 68	31	1	0
##### ###	83.86361	276	0.458989	62.797 76	-38	1	1
##### ###	71.99711	878	0.557305	96.101 25	-28	1	0
##### ###	76.15735	952	0.860632	37.153 55	32	1	0
##### ###	83.66341	381	0.53507	89.794 68	24	1	1
##### ###	91.20043	162	0.184463	66.891 69	-50	1	0
##### ###	47.03961	316	0.299596	87.814 45	37	0	0
##### ###	50.04664	953	0.309937	50.886 21	3	1	1
##### ###	27.51856	926	0.397318	88.796 3	20	0	0

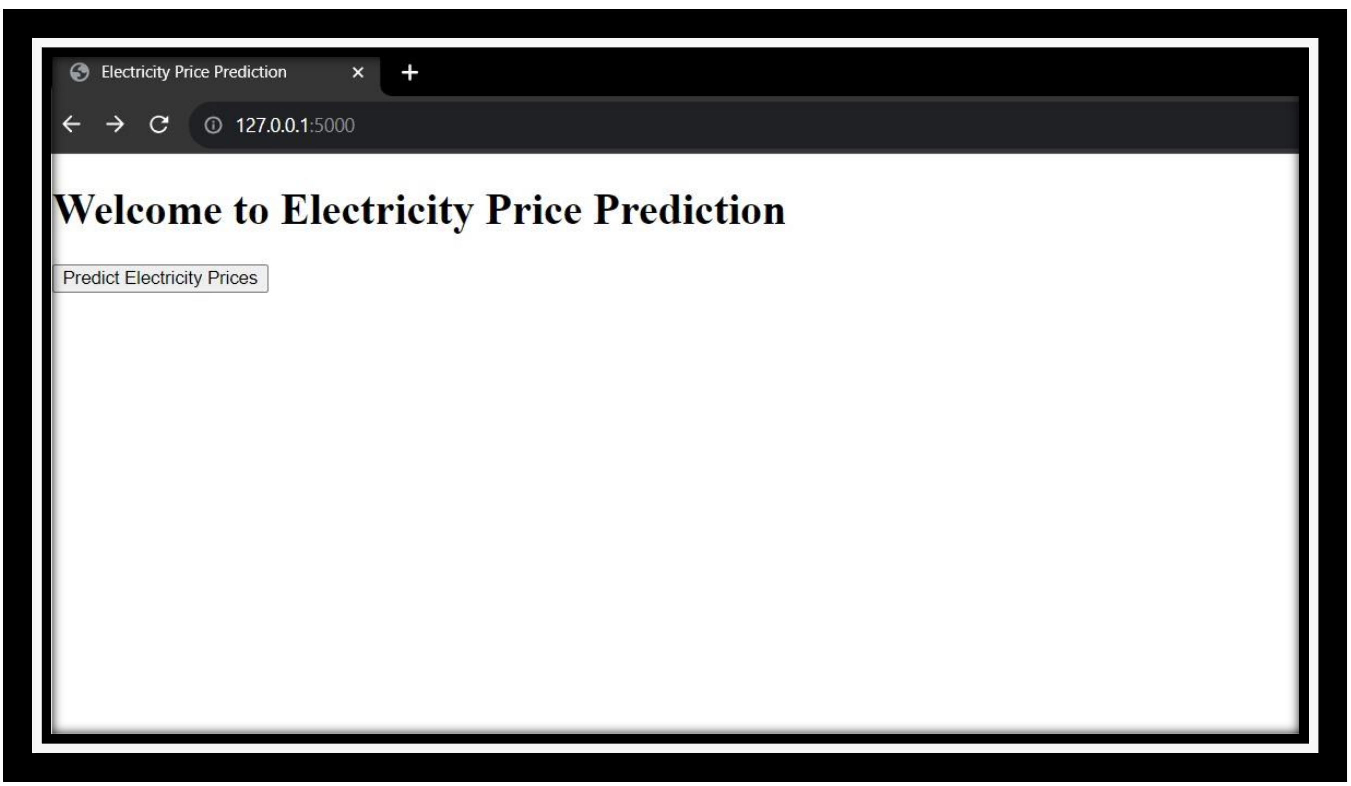
##### ###	66.26241	894	0.426789	84.097 59	38	1	1
##### ###	22.87538	788	0.799782	46.103 76	34	1	1
##### ###	57.24784	560	0.349387	39.193 46	-36	0	1
##### ###	63.41157	709	0.468232	90.763 8	14	0	1
##### ###	42.9233	204	0.624976	97.426 57	25	0	1
##### ###	67.26666	198	0.377726	32.191 16	20	0	0
##### ###	22.44002	610	0.836565	81.188 9	-36	1	0
##### ###	22.98786	484	0.587454	93.666 72	-30	0	1
##### ###	85.80804	504	0.294045	90.015 21	-32	1	1
##### ###	48.81525	922	0.714059	97.516 43	9	0	0
##### ###	30.16484	617	0.527648	83.206 59	42	1	1
##### ###	61.77946	575	0.534456	60.960 91	38	1	1
##### ###	81.59948	808	0.481088	51.836 64	-18	1	1
##### ###	37.26568	962	0.497012	52.451 99	-2	1	0
##### ###	69.83124	744	0.765356	80.709 59	45	1	0
##### ###	26.8278	969	0.102981	82.208 59	-16	1	1
##### ###	24.13454	870	0.334363	44.025 14	-7	0	1
##### ###	62.50837	536	0.075503	90.768 6	39	1	0
##### ###	63.25081	122	0.753246	80.044 68	-40	0	0
##### ###	70.99439	664	0.272302	69.392 52	48	0	1
##### ###	78.08731	776	0.89743	63.007 27	21	1	1
##### ###	98.06817	813	0.526578	68.026 3	27	0	0
##### ###	61.30403	557	0.80075	41.147 2	23	1	0
##### ###	45.83652	182	0.978931	44.259 09	15	0	0

##### ###	83.6149	244	0.839789	36.381 51	-5	1	1
##### ###	41.66658	184	0.866994	40.756 06	-42	0	0
##### ###	55.11771	177	0.407984	61.758 55	-48	1	1
##### ###	26.27651	556	0.551723	67.124 81	34	1	0
##### ###	22.02806	977	0.253889	31.250 33	-25	1	0
##### ###	97.01187	100	0.196113	85.775 85	21	1	0
##### ###	86.87841	150	0.505508	50.591 76	-3	0	0
##### ###	75.67794	784	0.595049	93.828 52	15	1	1
##### ###	52.71624	816	0.339271	97.725 79	19	0	1
##### ###	33.86355	871	0.569444	42.982 12	-23	0	0
##### ###	32.51496	545	0.88746	62.346 81	42	0	1
##### ###	40.01943	548	0.556722	30.086 23	-25	1	0
##### ###	63.93813	980	0.720813	71.882 91	-26	0	0
##### ###	77.16767	587	0.805315	72.511 45	-39	0	0
##### ###	72.81579	899	0.988729	47.327 8	-46	0	0
##### ###	42.39471	133	0.603197	69.088 89	47	1	1
##### ###	96.38922	447	0.80685	45.787 81	-3	1	0
##### ###	79.03175	194	0.962649	41.257 78	-14	0	0
##### ###	64.34832	171	0.94441	75.606 36	28	1	1
##### ###	68.93766	650	0.141082	52.814 78	-7	1	0
##### ###	53.568	253	0.406392	71.318 48	49	0	0
##### ###	39.81848	349	0.32399	60.327 27	-36	0	0
##### ###	48.47781	773	0.086925	98.904 6	-8	1	1
##### ###	80.62769	537	0.633241	40.470 64	-3	0	0

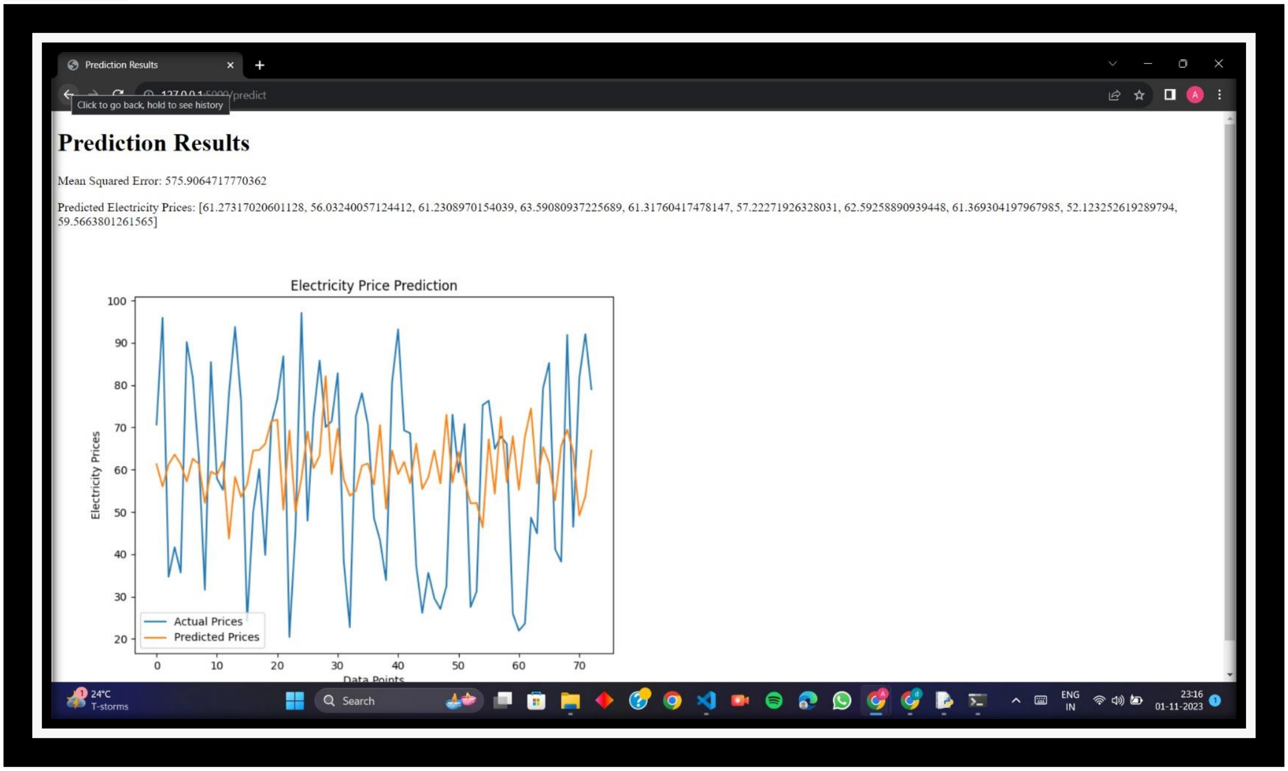
##### ###	21.15148	217	0.735905	62.521 9	36	1	1
##### ###	29.28581	870	0.848128	77.631 96	19	1	0
##### ###	23.68021	990	0.122777	57.688 16	-5	1	0
##### ###	23.2583	405	0.876444	98.228 99	-8	1	1
##### ###	88.43685	367	0.642926	80.532 13	-30	1	0
##### ###	76.29263	153	0.703946	75.988 87	-28	1	0
##### ###	57.93391	335	0.910624	82.994 15	29	0	1
##### ###	27.82673	488	0.624761	96.685 4	-4	0	1
##### ###	59.32927	833	0.335866	93.547 31	-36	1	1
##### ###	57.87774	862	0.825107	80.699 97	-26	1	1
##### ###	33.85615	924	0.363059	82.441 31	-15	1	1
##### ###	54.70813	500	0.034228	40.214 55	24	0	0
##### ###	51.88038	723	0.830655	37.195 47	0	1	1
##### ###	69.26801	866	0.345192	69.963 32	-36	0	0
##### ###	70.80749	402	0.773835	31.285 73	17	1	0
##### ###	23.62432	306	0.362758	63.764 95	-30	1	0
##### ###	49.96901	328	0.861068	44.724 47	-38	0	1
##### ###	70.06879	952	0.219511	54.221 16	35	0	1
##### ###	60.2509	677	0.974547	84.306 33	-32	1	0
##### ###	88.51919	174	0.779759	95.886 63	-17	1	1
##### ###	72.69549	534	0.114238	35.069 14	27	1	1
##### ###	33.03475	202	0.565777	67.396 95	17	0	1
##### ###	25.6455	521	0.985367	91.777 34	1	0	1
##### ###	71.39354	803	0.471065	35.775 6	-24	0	0

########	22.1209	863	0.182107	87.71598	-22	0	0
########	66.86205	837	0.484778	55.04609	-23	0	1
########	95.21842	905	0.512458	52.54663	-16	0	0
########	66.03793	405	0.741687	98.42265	-43	1	1
########	51.05359	197	0.698846	44.06014	-35	1	1
########	71.46306	949	0.40255	78.52358	-15	0	0
########	56.66023	897	0.218023	87.50085	-21	1	0
########	63.64934	434	0.645054	32.88284	-33	1	0
########	95.31718	958	0.421704	76.92698	-49	0	1

OUTPUT:



OUTPUT:



CONCLUSION:

Our project has demonstrated the effectiveness of data science and machine learning in providing accurate electricity price predictions. This predictive capability holds immense potential for optimizing resource allocation, reducing costs, and enhancing decision-making in the energy sector. By harnessing the power of data, we pave the way for a future where data-driven insights empower a more efficient and sustainable energy landscape.

REFERENCES:

https://www.tandfonline.com/doi/full/10.1080/1331677X.2018.1429291

https://www.sciencedirect.com/science/article/abs/pii/S0142061508000884

https://ieeexplore.ieee.org/abstract/document/5540263