Electricity Price Prediction

Problem Statement:

There are a number of issues that arise due to manual billing which includes incorrect computation / calculations, improper meter reading, delayed bill delivery, rounding off issues etc. Another major drawback of manual billing is the storage of the bills and maintaining a history of electricity consumption.

The user is not bound to pay excesses amount of money, user must pay according to their requirement. It can reduce problems associated with billing consumer living in isolated areas and reduce deployment of manpower for taking meter readings.

Real time and Accurate billing information

Design Thinking Process:

Design thinking is a problem-solving approach that focuses on understanding users' needs and creating innovative solutions. It typically involves the following stages:

1.Empathize: In this phase, you aim to understand the users' needs and gather insights into the problem. This often involves interviews, surveys, and observations to build empathy with users.

2.Define: After understanding the problem, you define it in a way that captures the core issues. This is where you create a clear problem statement.

3.Ideate: In this phase, you brainstorm and generate a wide range of potential solutions to the defined problem.

4.Prototype: You create prototypes or mock-ups of potential solutions to test and refine your ideas. This could be in the form of sketches, physical models, or digital prototypes.

5.Test: You test the prototypes with users to understand how well they address the problem. This iterative process helps refine and improve the solution.

Phases of Development:

1.Conceptualization: Defining the concept and scope of the project, including high-level requirements and objectives.

2.Design and Planning: Creating detailed designs, architecture, and project plans, including technical specifications, user interface design, and resource allocation.

3.Development: Building the actual product or system based on the design and plans. This involves coding, hardware development, or other activities required to create the solution.

4. **Testing and Quality Assurance:** Rigorous testing is conducted to ensure the product works as intended and

meets quality standards. This includes functional testing, performance testing, and user acceptance testing.

5.Deployment: Rolling out the solution to the target audience or environment. This can involve pilot launches, soft launches, or full-scale deployments.

6.Monitoring and Maintenance: Continuously monitoring the solution's performance, addressing issues, and making updates and improvements as necessary.

Dataset Description:

Dataset Link:

https://github.com/Akash-Jeyachandran/Electricity-price-prediction/Electricity.csv

1.DateTime:

This is a crucial feature for time series analysis. It allows the model to capture temporal patterns and seasonality in electricity prices. For example, electricity prices often vary by time of day, day of the week, and month.

2.Holiday:

Holidays can impact electricity demand and supply patterns. Prices may change during holidays due to altered consumption behaviour or reduced industrial activity.

3.Day Of Week:

Electricity price patterns often differ by day of the week. For instance, prices may be higher on weekdays when industrial and commercial demand is higher.

4.Week Of Year:

The week of the year can help capture seasonal variations in electricity prices, such as higher prices during peak summer or winter weeks.

5.Day:

The day of the month may reveal any monthly billing cycles or patterns in consumer behaviour.

6. Month:

Different months often have distinct electricity consumption patterns. For instance, prices may rise during the summer months due to increased air conditioning use.

7. Year:

Trends in electricity prices may change over the years, and this feature can help account for long-term shifts.

8.Period Of Day:

Dividing the day into periods (e.g., morning, afternoon, evening) can capture variations in demand and supply at different times.

9.Forecast Wind Production:

Information about forecasted wind power production can be valuable for understanding the supply side of the electricity market, as wind energy can impact prices.

10.SystemLoadEA:

The system load in a specific area (EA) is an essential feature, as it represents the overall demand for electricity.

11.SMPEA:

The spot market price in a specific area (EA) reflects market conditions and can be used to identify price trends.

12.ORKTemperature:

Temperature data can influence electricity demand (e.g., heating or cooling needs) and may correlate with price fluctuations.

13.ORKWindspeed:

Wind speed data is relevant if wind power generation is a significant part of the energy mix in the region.

14.CO2Intensity:

Carbon dioxide intensity is an environmental factor that can affect electricity prices, particularly in regions with carbon pricing or emissions targets.

15.ActualWindProduction:

Actual wind power production is relevant for assessing the supply of renewable energy sources.

16.SystemLoadEP2:

Similar to SystemLoadEA, this feature represents electricity demand in a specific area (EP2).

17.SMPEP2:

The spot market price in a specific area (EP2) can provide insights into regional price variations

Data Preprocessing Steps:

1. Data Cleaning: Remove or impute missing values, correct errors, and handle outliers in the dataset.

2. Feature Selection: Choose relevant features and transform them as needed. Feature engineering can involve scaling, one-hot encoding, and more.

3.Data Splitting: Divide the dataset into a training set and a test set to evaluate the model's performance.

4.Normalization/Standardization: Scale the features to have zero mean and unit variance, which can improve model convergence.

5.Handling Imbalanced Data: Address class imbalances if present, for example, by oversampling, under sampling , or using synthetic data generation techniques.

6.Text or Image Data Processing: For text or image data, preprocessing can include tokenization, stemming/lemmatization, vectorization or resizing/rescaling for images.

7.Data Augmentation: For image data, data augmentation techniques like rotation, flipping, and cropping can be applied to increase the diversity of the training data.

Model Training Process:

1.Select a Model: Choose an appropriate machine learning or deep learning model for your task. The choice depends on the nature of the problem, the dataset, and your goals.

2.Model Architecture: Define the architecture of the model, including the number of layers, units, and activation functions. In deep learning, this is often done using frameworks like TensorFlow or PyTorch.

3.Loss Function: Define a suitable loss function that the model will minimize during training. The choice of loss function depends on the problem, such as mean squared error for regression or cross-entropy for classification.

4. Optimization Algorithm: Select an optimization algorithm to minimize the loss function

5.Training: Train the model on the training data using the chosen optimization algorithm. This involves iterating through the training dataset for multiple epochs.

6.Evaluation: Use the test set to evaluate the model's performance, using metrics like accuracy, precision, recall, F1-score, or mean squared error, depending on the task.

7. Hyperparameter Tuning: Fine-tune hyperparameters like learning rate, batch size, and regularization strength to improve model performance.

8.Deployment: If the model performs satisfactorily, deploy it in a production environment for making predictions on new data.

Time Series Forecasting Algorithm:

1. ARIMA (AutoRegressive Integrated Moving Average):

- ARIMA is a classic and widely used time series forecasting model. It's suitable when the electricity price data exhibits a clear temporal structure, including trends and seasonality.

- ARIMA comprises three main components: AutoRegressive (AR), Integration (I), and Moving Average (MA). These components capture past values, differencing to achieve stationarity, and the influence of past forecast errors.

- Use ARIMA when you observe seasonality and autocorrelation in the data. It's particularly effective for short to medium-term predictions.

Algorithm:

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from statsmodels.tsa.arima_model import ARIMA
from statsmodels.graphics.tsaplots import plot_acf, plot_pacf
np.random.seed(42)
time = pd.date_range(start='2023-01-01', periods=100, freq='D')
data = np.cumsum(np.random.randn(100))
ts = pd.Series(data, index=time)
plt.figure(figsize=(12, 6))
plt.plot(ts)
plt.title("Sample Time Series Data")
plt.xlabel("Date")
plt.ylabel("Value")
plt.show()
plot_acf(ts, lags=40)
plot_pacf(ts, lags=40)
plt.show()
```

```
p, d, q = 1, 1, 1
model = ARIMA(ts, order=(p, d, q))
model_fit = model.fit(disp=0)
print(model_fit.summary())
forecast_steps = 10
forecast, stderr, conf_int =
model_fit.forecast(steps=forecast_steps)
plt.figure(figsize=(12, 6))
plt.plot(ts, label="Original Data")
plt.plot(pd.date_range(start='2023-04-10',
periods=forecast_steps, freq='D'), forecast,
label="Forecast", color='red')
plt.fill_between(pd.date_range(start='2023-04-10',
periods=forecast_steps, freq='D'),
                 forecast - stderr, forecast + stderr,
color='pink', alpha=0.3, label='Confidence Interval')
plt.title("ARIMA Forecast")
plt.xlabel("Date")
plt.ylabel("Value")
plt.legend()
```

```
plt.show()
```

Evaluation Metrics:

1.Mean Absolute Error (MAE):

- MAE measures the average absolute difference between the model's predictions and the actual electricity prices. It provides a straightforward and interpretable measure of the model's accuracy.

- Lower MAE values indicate better prediction accuracy.

2.Mean Squared Error (MSE):

- MSE calculates the average of the squared differences between the model's predictions and actual prices. It gives more weight to larger errors.

- Like MAE, lower MSE values indicate better accuracy, but it penalizes larger errors more severely.

3.Root Mean Squared Error (RMSE):

- RMSE is the square root of MSE. It provides an estimate of the standard deviation of prediction errors, which can be useful for understanding the spread of errors.

- As with MAE and MSE, lower RMSE values indicate better prediction accuracy.

4.R-squared (R2) Score:

- R2 measures the proportion of variance in electricity prices explained by the model. It provides insight into how well the model fits the data.

- An R2 score close to 1 indicates a good fit, while a score close to 0 suggests that the model doesn't explain much variance.