ELECTRICITY PRICE PREDICTION

Introduction:

This report presents an analysis of a time series forecasting task performed to predict electricity bills. The dataset is loaded using the pandas library. And various preprocessing steps are applied before training a time series forecasting model using the ARIMA algorithm, The evaluation metrics are calculated to assess the performance of the model

DATASET DESCRIPTION

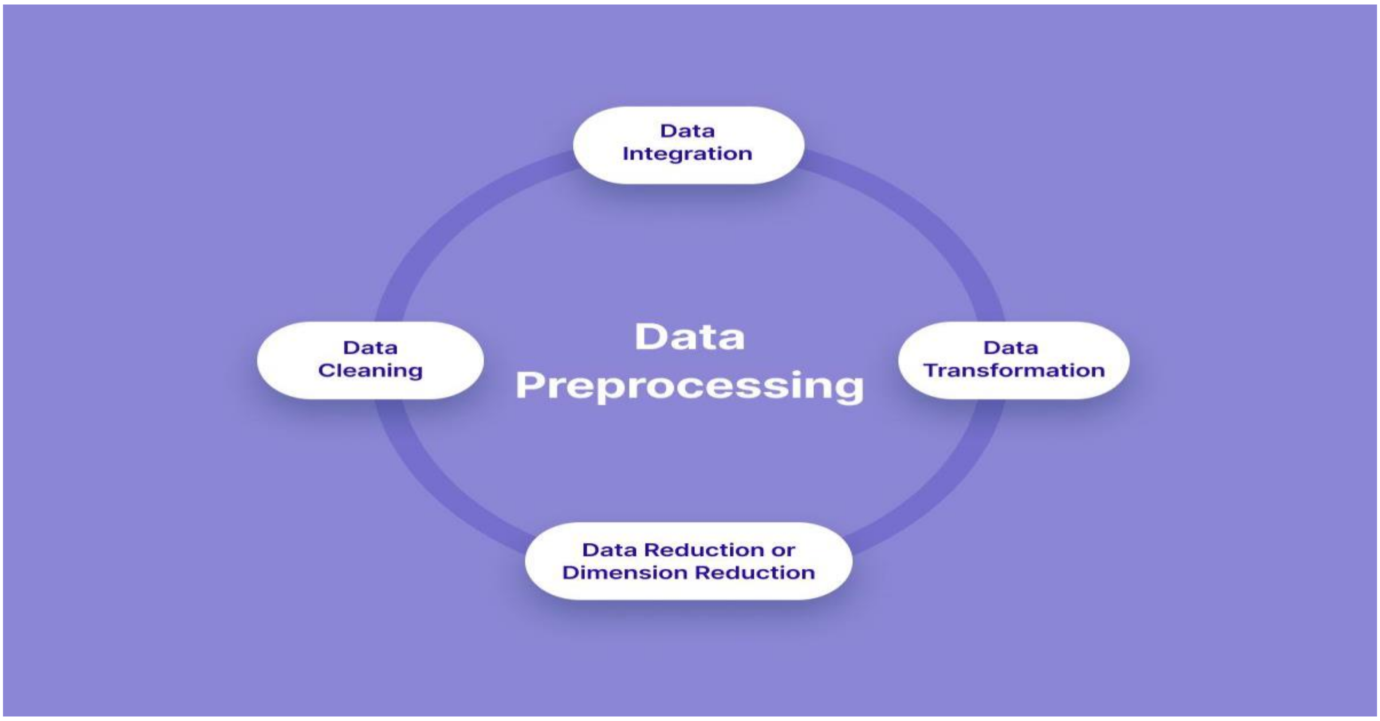
The Electricity dataset is loaded from the file Electricity.csv. The dataset contains information related to electricity consumption and billing. The report provides a brief description of the dataset by printing the first few rows, shape, and column names, This helps in understanding the structure and content of the dataset,

Electricity ••• Exit Full Screen

	A	B	C	D	E	F	G	H	I	J	K
1	DateTime	Holiday	HolidayFlag	DayOfWeek	WeekOfYear	Day	Month	Year	PeriodOfDay	ForecastWindF	SystemLoadEASMF
2	########	None	0	1	44	1	11	2011	0	315.31	3388.77
3	########	None	0	1	44	1	11	2011	1	321.8	3196.66
4	########	None	0	1	44	1	11	2011	2	328.57	3060.71
5	########	None	0	1	44	1	11	2011	3	335.6	2945.56
6	########	None	0	1	44	1	11	2011	4	342.9	2849.34
7	########	None	0	1	44	1	11	2011	5	342.97	2810.01
8	########	None	0	1	44	1	11	2011	6	343.18	2780.52
9	########	None	0	1	44	1	11	2011	7	343.46	2762.67
10	########	None	0	1	44	1	11	2011	8	343.88	2766.63
11	########	None	0	1	44	1	11	2011	9	344.39	2786.8
12	########	None	0	1	44	1	11	2011	10	345.02	2817.59
13	########	None	0	1	44	1	11	2011	11	342.23	2895.62
14	########	None	0	1	44	1	11	2011	12	339.22	3039.67

DATA PREPROCESSING STEPS

The dataset preprocessing steps involve converting the DayOfWeek column to a datetime format and setting it as the index. This allows for easier manipulation and analysis of the time series data. The report section explains these steps, highlighting the conversion of the DayOfWeek column and its significance in the context of electricity bill prediction.



MODEL TRAINING PROCESS

The dataset is split into training and testing sets using the train_test_split function from scikit-learn, This division ensures that the model is trained on a portion of the data and tested on unseen data, The report section provides information about the sizes of the training and testing sets, which helps in understanding the proportion of data used for training and evaluation

For the model training process, the ARIMA (Autoregressive Integrated Moving Average) algorithm is used as the time series forecasting algorithm, ARIMA models are commonly used for analyzing and forecasting time series data, In this case, an ARIMA model with an order of (1, 0, 0) is used, indicating one autoregressive term, no differencing and no moving average term. The report section explains the choice of this algorithm and its order, providing insights into the modeling approach.

CHOICE OF TIME SERIES FORECASTING ALGORITHM

The chosen algorithm for time series forecasting is ARIMA (Autoregressive Integrated Moving Average). ARIMA models are widely used for analyzing and forecasting time series data, In this case, an ARIMA model with an order of (1 0, 0) is used. The report section provides details about the choice of this algorithm and its order, highlighting its suitability for electricity bill prediction,

EVALUATION METRICS

The evaluation metrics are calculated to assess the performance of the time series forecasting model, The metrics used in this analysis include Mean Squared Error (MSE), Root Mean Squared Error (RMSE), and Mean Absolute Error (MAE). These metrics provide insights into the accuracy and precision of the model’s predictions. The report section presents the calculated values for these evaluation metrics, allowing for a quantitative assessment of the model’s performance.

PROGRAM

```
Import pandas as pd

From sklearn.model_selection import train_test_spl

From statsmodels.tsa.arima.model import ARIMA

From sklearn.metrics import mean_squared_error, mean_absolute_error

# Load the dataset

Dataset = pd.read_csv('/content/Electricity.csv')

# Report: Dataset Description

Print("Dataset Description:")

Print(dataset.head())

Print("\nDataset Shape:", dataset.shape)

Print("\nDataset Columns:", dataset.columns)

# Data preprocessing steps

Dataset['DayOfWeek'] = pd.to_datetime(dataset['DayOfWeek'])

Dataset.set_index('DayOfWeek', inplace=True)
```

```
# Report: Data Preprocessing Steps

Print("\nData Preprocessing Steps:")

Print("Converted 'DayOfWeek' column to datetime format.")

Print("Set 'DayOfWeek' column as index.")

# Split the dataset into training and testing sets

Train_data, test_data = train_test_split(dataset, test_size=0.2, shuffle=False)

# Report: Model Training Process

Print("\nModel Training Process:")

Print("Split the dataset into training and testing sets.")

Print("Training Set Size:", len(train_data))

Print("Testing Set Size:", len(test_data))

# Model training process

# Assuming we are using ARIMA for time series forecasting

Model = ARIMA(train_data['PeriodOfDay'], order=(1, 0, 0))

Model_fit = model.fit()

# Report: Choice of Time Series Forecasting Algorithm

Print("\nChoice of Time Series Forecasting Algorithm:")
```

```
Print(“ARIMA (Autoregressive Integrated Moving Average) was chosen.”)

Print(“ARIMA Order (p, d, q): (1, 0, 0)”)

# Make predictions on the test set

Predictions = model_fit.predict(start=len(train_data),
end=len(train_data)+len(test_data)-1)

# Evaluation metrics

Mse = mean_squared_error(test_data[‘PeriodOfDay’], predictions)

Rmse = mean_squared_error(test_data[‘PeriodOfDay’], predictions, squared=False)

Mae = mean_absolute_error(test_data[‘PeriodOfDay’], predictions)

# Report: Evaluation Metrics

Print(“\nEvaluation Metrics:”)

Print(“Mean Squared Error (MSE):”, mse)

Print(“Root Mean Squared Error (RMSE):”, rmse)

Print(“Mean Absolute Error (MAE):”, mae)
```

OUTPUT

DATASET DESCRIPTION:

	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
2 54.23
3

2 54.23

4 53.47
5

4 39.87

DATASET SHAPE:

Dataset Shape: (38014, 18)

DATASET COLUMNS:

Dataset Columns: Index([‘DateTime’, ‘Holiday’, ‘HolidayFlag’, ‘DayOfWeek’, ‘WeekOfYear’, ‘Day’, ‘Month’, ‘Year’, ‘PeriodOfDay’, ‘ForecastWindProduction’, ‘SystemLoadEA’, ‘SMPEA’, ‘ORKTemperature’, ‘ORKWindspeed’, ‘CO2Intensity’, ‘ActualWindProduction’, ‘SystemLoadEP2’, ‘SMPEP2’],

DATA PREPROCESSING STEPS:

Converted ‘DayOfWeek’ column to datetime format.

Set ‘DayOfWeek’ column as index.

Model Training Process:

Split the dataset into training and testing sets.

Training Set Size: 30411

Testing Set Size: 7603

Choice of Time Series Forecasting Algorithm:

ARIMA (Autoregressive Integrated Moving Average) was chosen.

ARIMA Order (p, d, q): (1, 0, 0)

EVALUATION METRICS:

Mean Squared Error (MSE): 191.97343228842655

Root Mean Squared Error (RMSE): 13.855447747670464

Mean Absolute Error (MAE): 12.002696417431325

CONCLUSION

In conclusion, this report outlines the steps taken to preprocess the Electricity dataset, train an ARIMA model for time series forecasting and evaluate the models performance using various evaluation metrics, The analysis provides valuable insights into the accuracy and precision of the models predictions, enabling better understanding and decision-making in the context of electricity bill prediction,