Institute of Technology of Cambodia

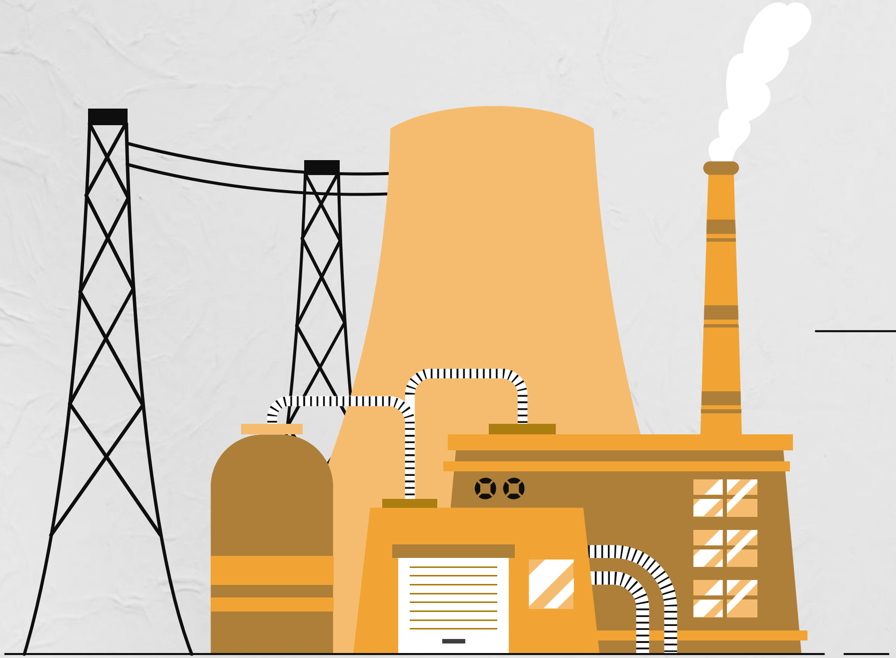
Department of Applied Mathematics and Statistics

Group: I3-AMS-TP-B (1)

Lecturers: Dr. PHAUK Sokkey (Course)
Mr. PEN Chentra (TP)

Presented by:

MON Sreylin	e20221701
PHYRUN Pichchhorda	e20220895
SOM Chann Reaksmey	e20220981
SONGSEANG Pisey	e20220225
SOPHON Rachana	e20220725



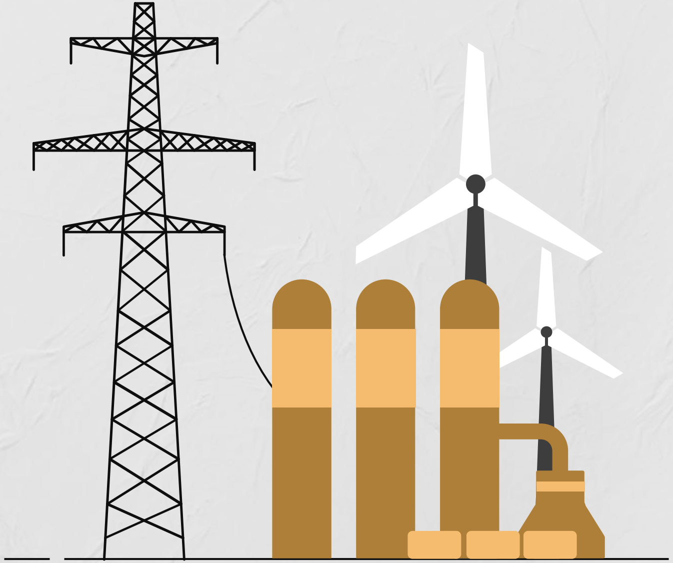
Electricity Price Prediction

Here is where our presentation begins

Table of contents

01

Introduction

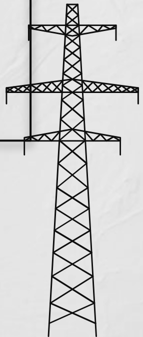


Overview

This project focuses on building models to analyze and predict electricity prices in the United States using historical data. By identifying key factors such as price, revenue, and sales across various sectors and states, the study aims to provide accurate and interpretable predictive models that can guide multiple stakeholders.

Dataset

The dataset provided is from the kaggle site that contains various information about many sectors across different states in the United States. The data spans multiple years and months (01-2001 to 01-2024), capturing key metrics such as price, revenue, and sales for each sectors.



Data Description

Entries

85870 entries

Numerical

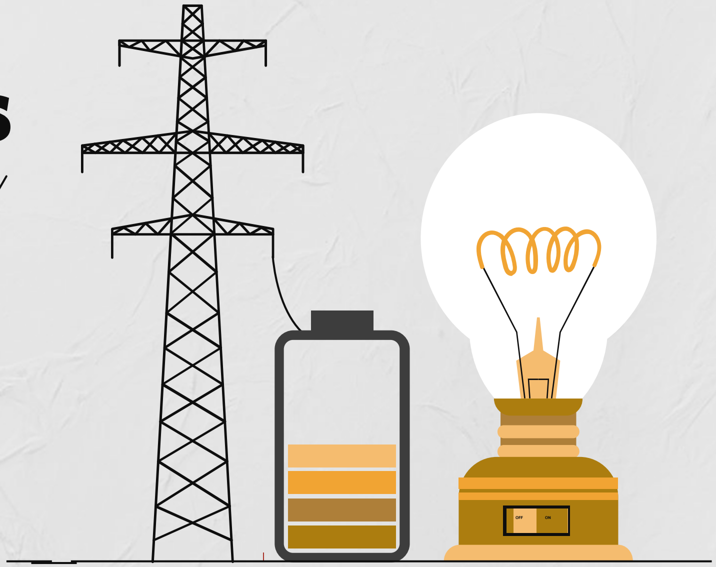
- Price
- Revenue
- Sales
- Year
- month

Categorical

- stateDescription
- sectorName

02

Exploratory Data Analysis



Data Cleaning

```
[19] dt.duplicated().sum()
```

```
0
```

```
dt.isnull().sum()
```

✓ 0.0s

```
year                0
month               0
stateDescription    0
sectorName          0
customers       26040
price               0
revenue             0
sales               0
dtype: int64
```

Handling Missing Values:
Null Values in Customers Column: 26,040
Action Taken: Dropped the customers column

```
dt.describe()
```

	year	month	price	revenue	sales
count	85870.000000	85870.000000	85870.000000	85870.000000	85870.000000
mean	2012.043321	6.480144	9.300193	586.627155	5980.048970
std	6.660304	3.461589	5.010382	2161.047702	21302.453181
min	2001.000000	1.000000	0.000000	-0.000010	0.000000
25%	2006.000000	3.000000	6.650000	29.475195	289.144572
50%	2012.000000	6.000000	8.840000	121.641500	1447.518085
75%	2018.000000	9.000000	11.380000	421.320628	4339.950965
max	2024.000000	12.000000	116.670000	52361.450970	391900.008970

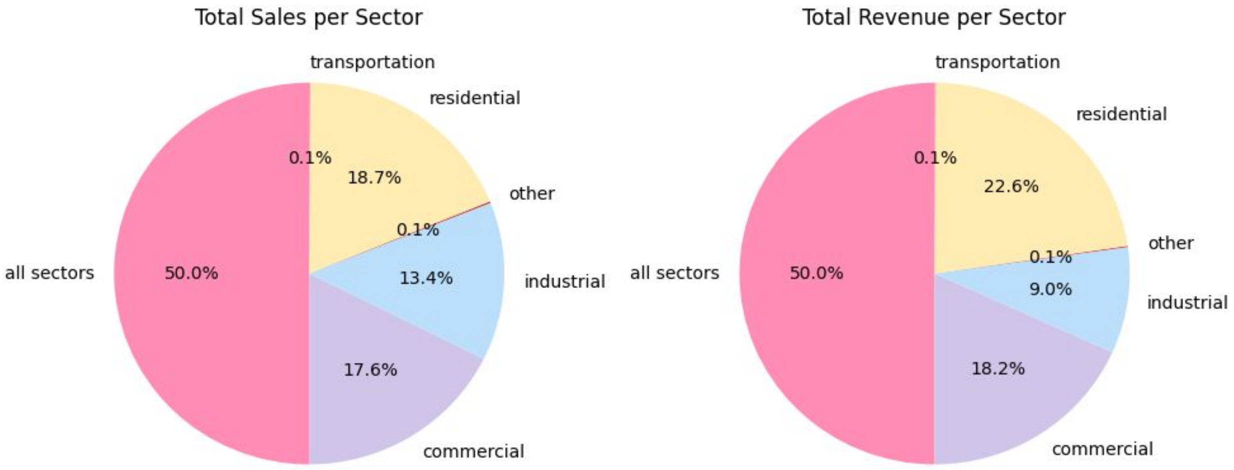
```
[ ] # Count negative revenue values
    negative_count = (dt['revenue'] < 0).sum()

    print(f"Number of negative revenue values: {negative_count}")
```

```
Number of negative revenue values: 2
```

Handling Outliers:
Negative Revenue Values: 2 instances (<0.002% of total data)
Action Taken: Retained for completeness (minimal impact on trends)

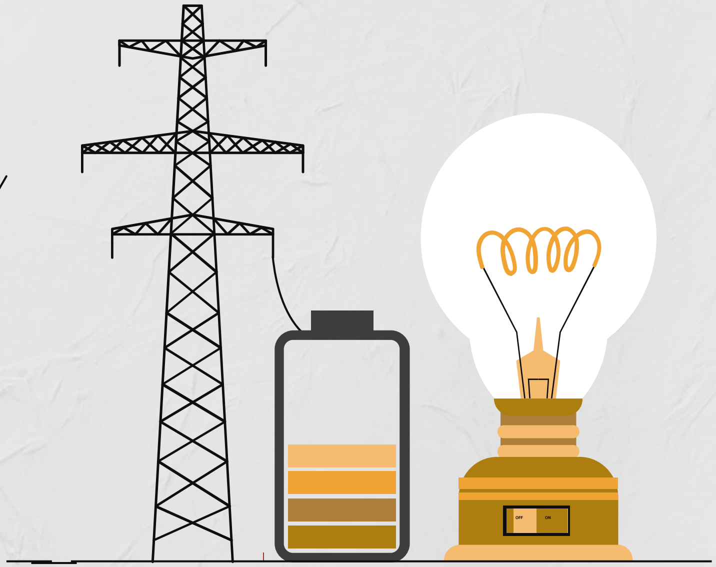
Data Visualization

2. Relationship Between Sectors and Numerical Variables



03

Data Preparation



Choosing Importance Features

```
from sklearn.ensemble import RandomForestRegressor

model = RandomForestRegressor()
model.fit(dt[['price', 'revenue', 'sales']], dt['price'])
print(model.feature_importances_)
```

✓ 12.0s

```
[9.98990346e-01 7.34540402e-04 2.75113378e-04]
```

```
from sklearn.feature_selection import RFE
from sklearn.linear_model import LinearRegression

estimator = LinearRegression()
selector = RFE(estimator, n_features_to_select=3)  # Select top 5 features
selector = selector.fit(dt[['price', 'revenue', 'sales']], dt['price'])
print(selector.support_)  # True for selected features, False otherwise
```

✓ 0.0s

```
[ True  True  True]
```

Random Forest Regressor Results:

- **Most Important Feature: Price** (highest importance score)
- **Second Most Important Feature: Revenue**
- **Least Important Feature: Sales** (lowest importance score)

RFE & Linear Regression Results:

- All features (**Price**, **Revenue**, **Sales**) are selected as relevant.

Splitting Data into Training and Testing Sets

Define feature X and variable target Y:
Feature (X): Predictor variables (**price, revenue, sales**)
Target Variable (Y): Price

```
# Define features (X) and target variable (y)
X = dt[['price', 'revenue', 'sales']]  #features
y = dt['price']  #target variable
```

Split Data: The data is split into training and testing sets using 80-20 split.

Splitting ensures that the model generalizes well to unseen data, reducing the risk of overfitting.

```
# Split data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

Models Used

Linear Model

Strengths:

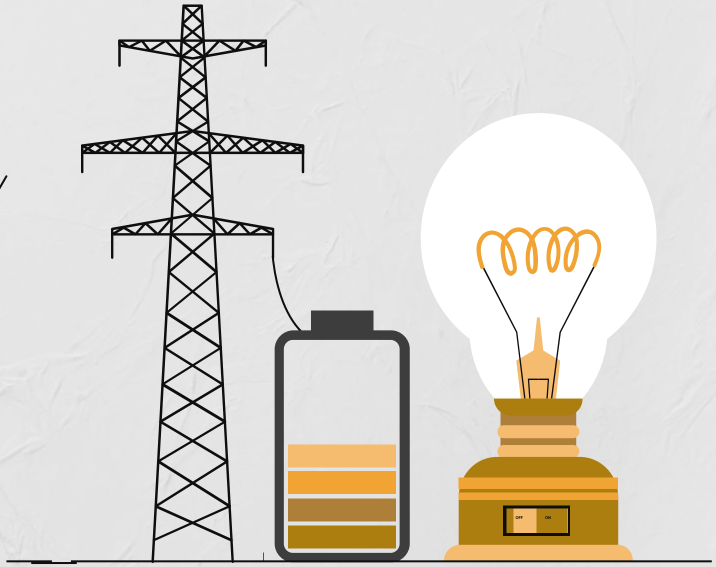
- Simplicity: Easy to implement and interpret.
- Efficiency: Quick training and prediction, suitable for smaller datasets.
- Baseline Comparison: Serves as a benchmark for evaluating more complex models.

Limitations:

- Linear Assumption: Assumes a linear relationship between features and the target variable, which may not always hold.
- Sensitivity to Outliers: Linear Regression is affected by extreme values, which can distort predictions.

04

Model Building



Linear Model

01

Split Data

X (features) and y (target)

02

Train Process

Choose Linear regression model for training

03

Make Prediction

Predict electricity prices for the testing dataset.

04

Evaluate model

Evaluate by finding R-squared and MSE

05

Analyze Error

Plot error term distribution and scatter y_test vs y_pred plot

Linear Model

01 Split Data

X (features) and y (target)

02 Train Process

Choose Linear regression model for training

03 Make Prediction

Predict electricity prices for the testing dataset.

```
# Define features (X) and target variable (y)
X = dt[['price', 'revenue', 'sales']]  # Example features
y = dt['price']  # Example target variable

# Split data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Initialize and train a model (example: Linear Regression)
model = LinearRegression()
model.fit(X_train, y_train)

# Make predictions on the test set
y_pred = model.predict(X_test)
```

Linear Model

04 Evaluate Model

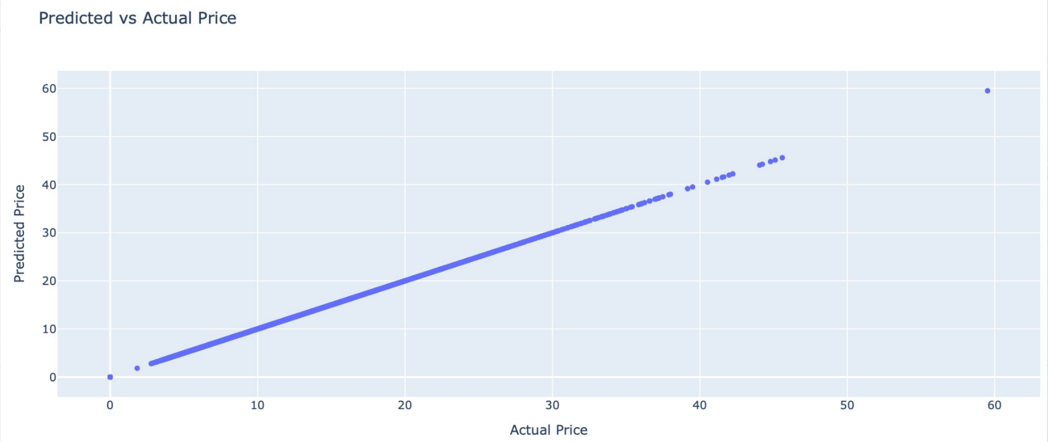
Evaluate by finding R-squared and MSE

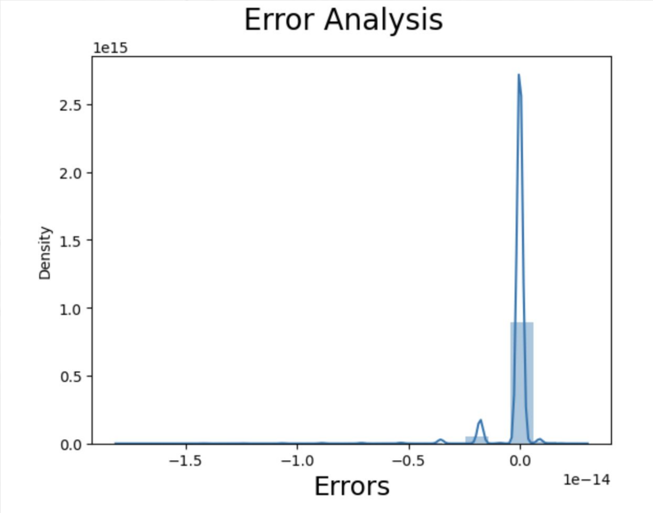
```
y_pred = model.predict(X_test)
mse = mean_squared_error(y_test, y_pred)
r2 = r2_score(y_test, y_pred)

print(f"Mean Squared Error: {mse}")
print(f"R-squared: {r2}")
```

```
Mean Squared Error: 8.171903547157617e-31
R-squared: 1.0
```

05 Analyze Error

Plot error term distribution and scatter y_test vs y_pred plot





train_and_predict_for_each_feature

01

Filter the Dataset

Focus on data for a specific state

02

Segment the Data

Group by `sectorName` to build separate models for each sector

03

Prepare the data for the model

- `X` (features): `Date (Convert to numerical format)`
- `y` (target): `Price`

04

Choose a Model

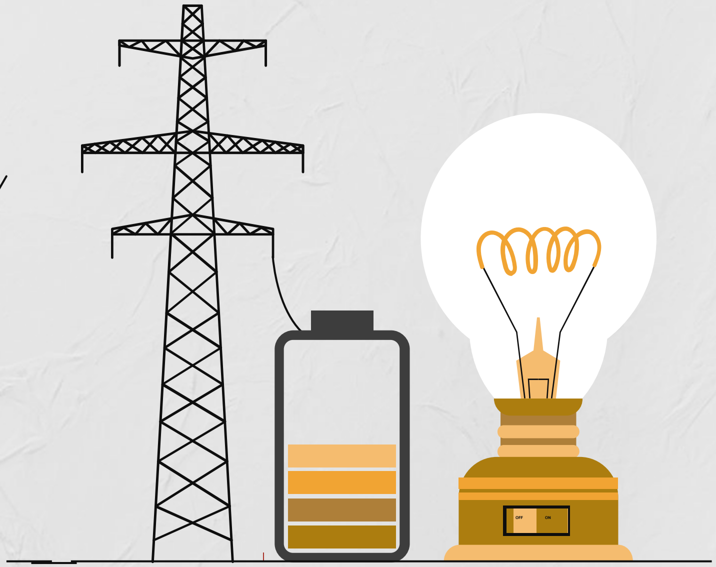
Linear Regression

05

Train, Make Prediction and Store to a DataFrame

05

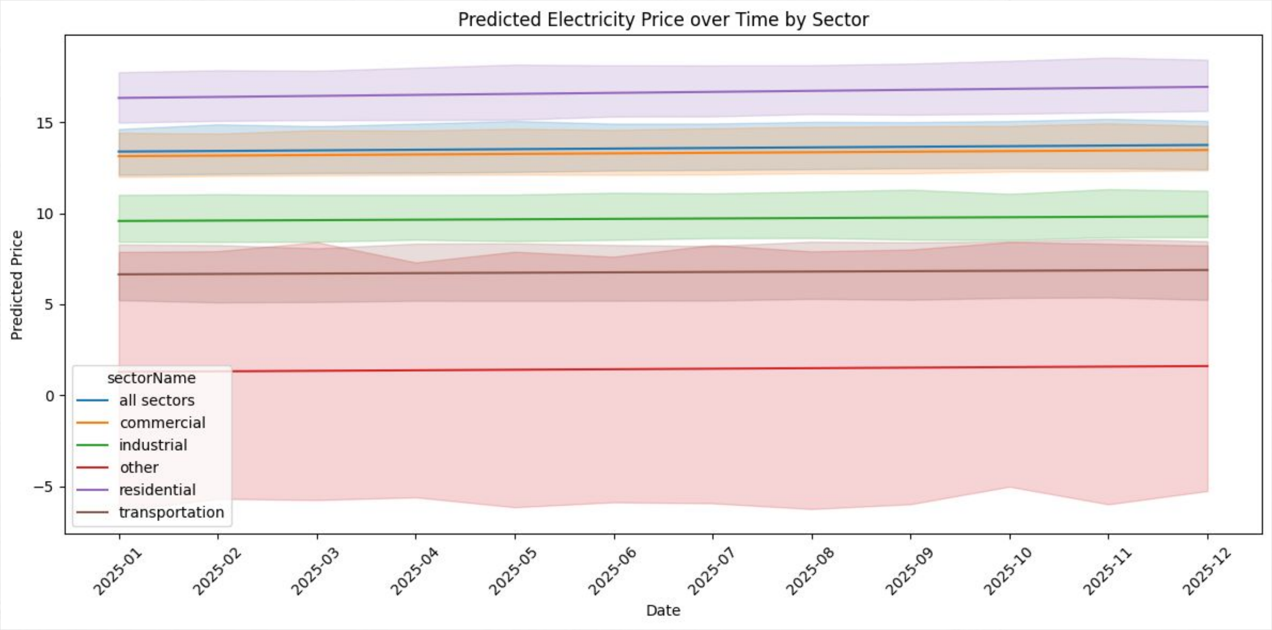
Result



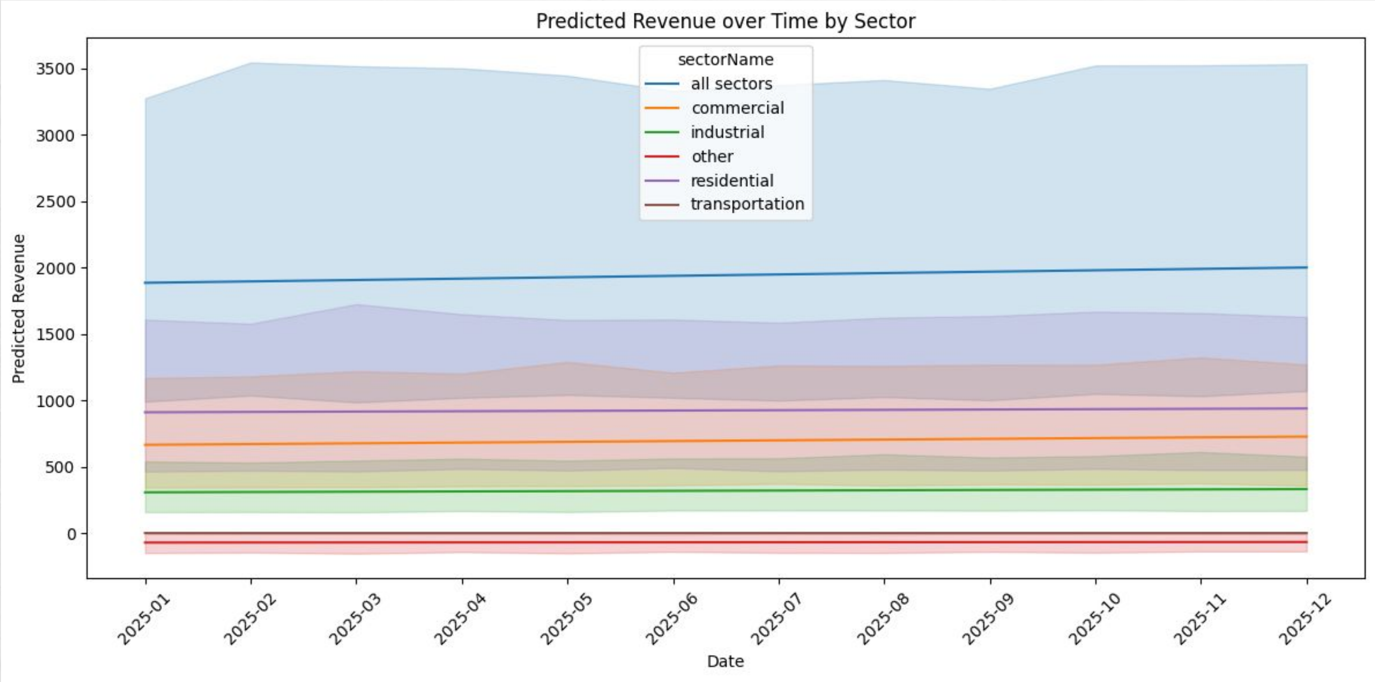
	stateDescription	sectorName	date	predicted_price	predicted_revenue	predicted_sales
0	Alabama	all sectors	2025-01	11.618125	843.626369	7273.747016
1	Alabama	all sectors	2025-02	11.650562	847.329642	7288.297147
2	Alabama	all sectors	2025-03	11.682998	851.032914	7302.847279
3	Alabama	all sectors	2025-04	11.715434	854.736186	7317.397410
4	Alabama	all sectors	2025-05	11.747871	858.439459	7331.947542
...	...	...	...	...	...	...
4459	Wyoming	transportation	2025-08	0.000000	0.000000	0.000000
4460	Wyoming	transportation	2025-09	0.000000	0.000000	0.000000
4461	Wyoming	transportation	2025-10	0.000000	0.000000	0.000000
4462	Wyoming	transportation	2025-11	0.000000	0.000000	0.000000
4463	Wyoming	transportation	2025-12	0.000000	0.000000	0.000000

4464 rows × 6 columns

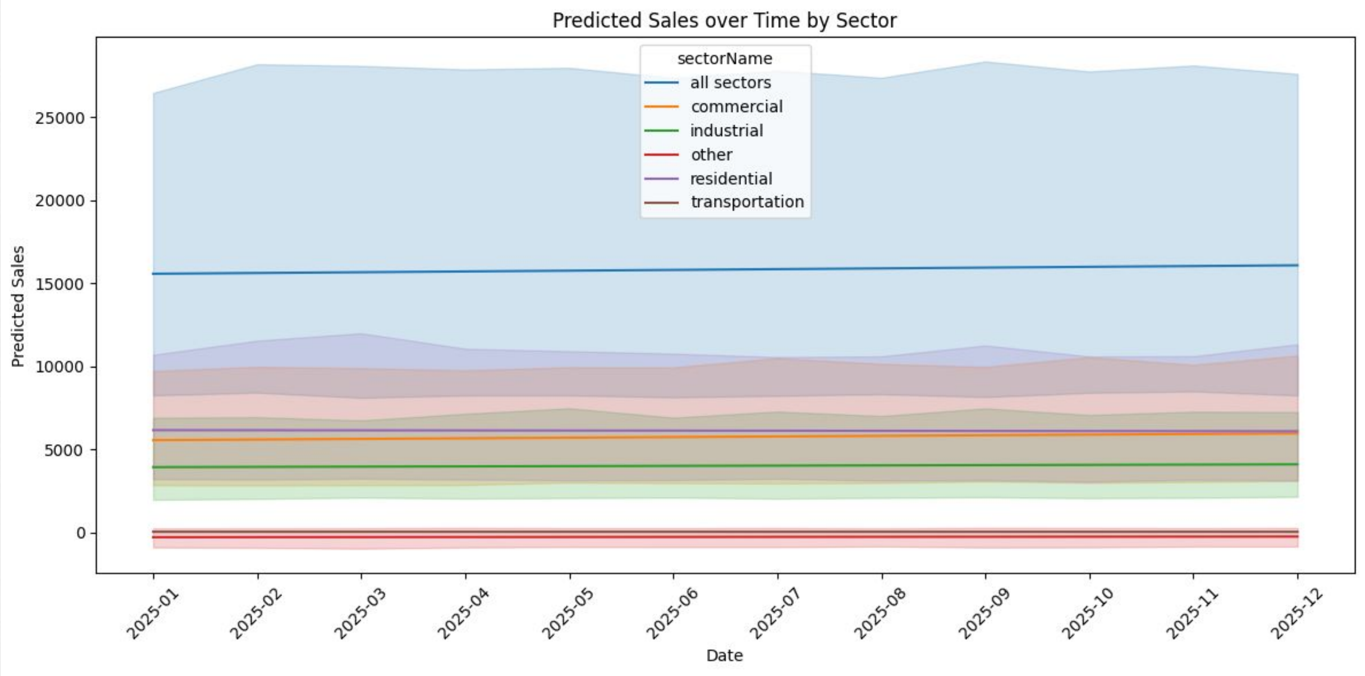
Visualize the Results



Visualize the Results

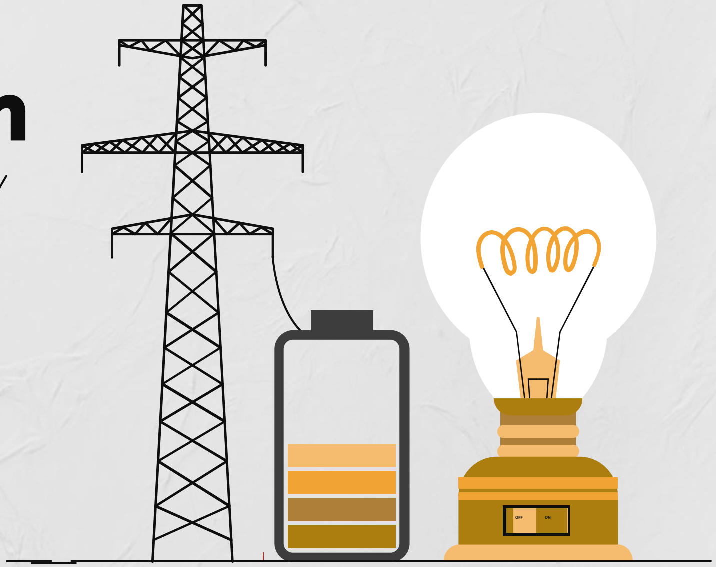


Visualize the Results



06

Discussion and Conclusion



Sector-Specific Trends

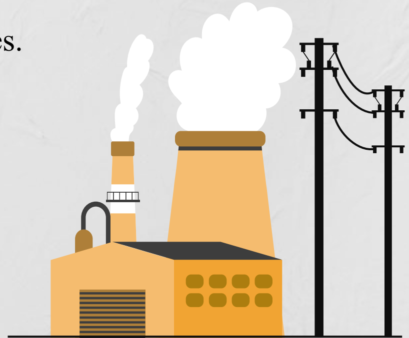
- **Residential Sector**: Higher prices and revenues due to consistent demand patterns.
- **Commercial & Industrial Sectors**: Lower prices reflect bulk energy usage discounts.

State-Level Insights

- Predictions align with historical trends, capturing **seasonal** and **sectoral variations** accurately.

Seasonality

- Seasonal peaks observed in **summer** and **winter**, especially in residential electricity prices.



Project Highlights

- Applied **Random Forest** and **RFE with Linear Regressor**.
- Predictions offer insights for **businesses**, **policymakers**, and **investors**.

Key Findings

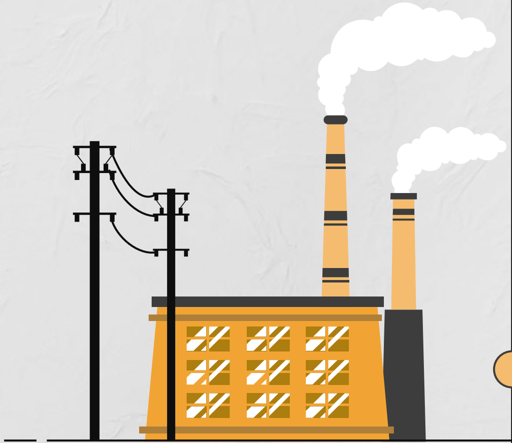
- **Seasonal demand** drives residential price peaks.
- **Bulk usage discounts** lower commercial/industrial prices.

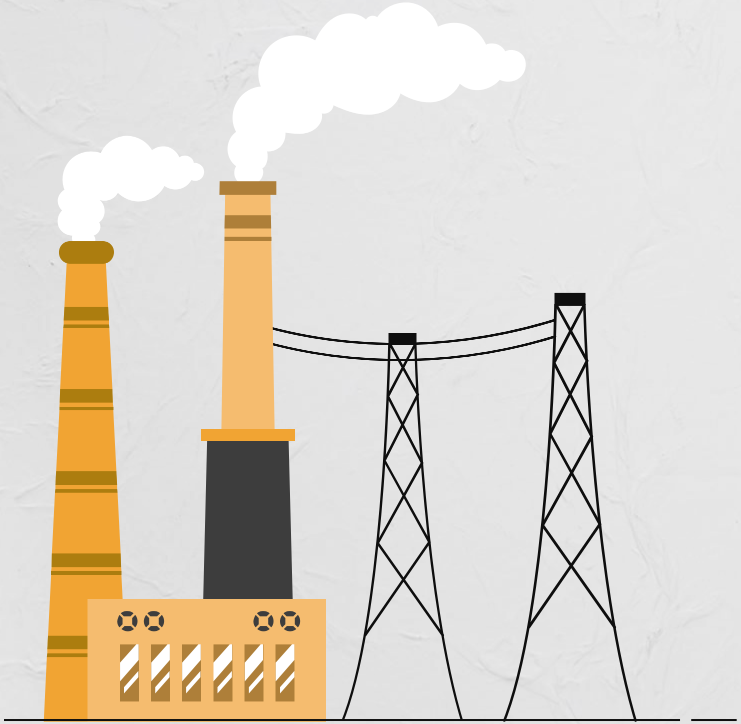
Conclusion & Future Directions

Predictive analytics provide real-world utility in the energy sector.

Next steps:

- Explore complex models and add predictors.
- Integrate **real-time data** for dynamic forecasting.





Thank You
For
Your Attention!!!