

1. Develop prediction model to determine electricity consumption

```
import pandas as pd

from prophet import Prophet

from sklearn.model_selection import train_test_split

from sklearn.ensemble import RandomForestRegressor

from sklearn.metrics import mean_absolute_error

# Load dataset

df = pd.read_csv("/content/Sets5_1_2_Household_electricity_Consumption_Datasets.csv")

# Use 'timestamp' column for datetime conversion, and then rename it to 'date' for consistency

df["date"] = pd.to_datetime(df["timestamp"])

# Create time features

df["hour"] = df["date"].dt.hour

df["day"] = df["date"].dt.day

df["month"] = df["date"].dt.month

df["weekday"] = df["date"].dt.weekday

# ----- Prophet Model -----

# The target column is 'energy_kwh', not 'consumption'

prophet_df = df.rename(columns={"date": "ds", "energy_kwh": "y"})[["ds", "y"]]

model_prophet = Prophet()

model_prophet.fit(prophet_df)

future = model_prophet.make_future_dataframe(periods=30) # Forecast next 30 days

forecast = model_prophet.predict(future)

print("Prophet Forecast (last 5 rows):")

print(forecast[["ds", "yhat"]].tail())

# ----- Random Forest Model -----"

X = df[["hour", "day", "month", "weekday"]]

y = df["energy_kwh"]
```

```

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, shuffle=False)

model_rf = RandomForestRegressor(n_estimators=100, random_state=42)

model_rf.fit(X_train, y_train)

predictions = model_rf.predict(X_test)

mae = mean_absolute_error(y_test, predictions)

print("\nRandom Forest MAE:", mae)

print("\nRandom Forest Predictions (first 5 rows):")

print(pd.DataFrame({"Actual": y_test.values, "Predicted": predictions}).head())

```

2] Perform exploratory data analysis on this dataset.

[https://drive.google.com/file/d/16pXCCVAPSgsA1ZgftUXLxCtQNVoJiRjm/view?usp=driv\\_e\\_link](https://drive.google.com/file/d/16pXCCVAPSgsA1ZgftUXLxCtQNVoJiRjm/view?usp=driv_e_link)

```

import pandas as pd

import matplotlib.pyplot as plt

df =
pd.read_csv("/content/Sets5_1_2_Household_electricity_Consumption_Datasets.csv")

df["date"] = pd.to_datetime(df["date"])

print(df.shape)

print(df.info())

print(df.describe())

print(df.isnull().sum())

df["hour"] = df["date"].dt.hour

df["day"] = df["date"].dt.day

df["month"] = df["date"].dt.month

df["weekday"] = df["date"].dt.weekday

plt.plot(df["date"], df["energy_kwh"])

plt.title("Electricity Consumption Over Time")

plt.show()

```

```
plt.hist(df["energy_kwh"], bins=50)
plt.title("Consumption Distribution")
plt.show()

hourly = df.groupby("hour")["energy_kwh"].mean()
plt.plot(hourly.index, hourly.values)
plt.title("Average Consumption by Hour")
plt.show()
```