Task4:-

🧠 **Objective:-**

Work with a messy real-world dataset (e.g., open gov or finance CSVs) and transform it into a clean, analyzable format for statistical modeling. You'll build a regression or classification model using statsmodels or SciPy, but do all data wrangling strictly in pandas.

📂 **Project Structure:-**

```
Data_modeling_project/
│
├── data/
│   └── raw_data.csv          # Original messy dataset
│   └── cleaned_data.csv      # Cleaned dataset (output)
│
├── data_prep.ipynb           # Notebook for cleaning & feature engineering
├── modeling.ipynb            # Notebook for statistical modeling and analysis
├── slides/
│   └── final_summary_slides.pptx  # Final presentation (5–7 slides)
│
├── requirements.txt
└── README.md
```

🔧 **Step-by-Step Tasks-**

🔷 **1. Data Cleaning (data_prep.ipynb)**

Goals-: Handle common data quality issues.

■Missing values:

●Use .fillna() or .interpolate() with context (e.g., time, forward-fill).

●Document percentage of missingness per column.

■Outlier detection:

●IQR-based or z-score filtering.

●Optional: Visualize with boxplots or scatterplots.

■Type conversion:

●Use .astype() and pd.to_datetime().

●Detect object columns that should be numeric, datetime, or categorical.

■Erroneous entries:

●Use .str.extract(), .replace(), .apply() to fix malformed strings or mixed data types.

🔷 2. Schema Inference & Normalization

Goals-: Convert raw format into structured and normalized layout.

■Reshape data:

●.melt() / .pivot() for wide ↔ long conversions.

●.stack() / .unstack() for multi-index management.

■Categorical handling:

●Convert strings to pd.Categorical.

●Sort or group by categories for optimization.

■Timestamp alignment (finance/time data):

●Use merge_asof() to align time series (e.g., prices vs. Events).

🔷 3. Feature Engineering

Goals-: Prepare for modeling.

■Add:

●Polynomial features: e.g., x**2, x**3

●Interaction terms: x1 * x2

●Lagged features (for time series)

●Group-based transformations: .groupby().transform()

■Encode categorical variables:

●.get_dummies()

●Or use statsmodels C(variable) formula syntax

📊 4. Statistical Modeling (modeling.ipynb)

- Use statsmodels for regression with inference.

●Linear regression (continuous target):

```
Import statsmodels.api as sm
Model = sm.OLS(y, X).fit()
Print(model.summary())
```

●Logistic regression (binary target):

```
Model = sm.Logit(y, X).fit()
```

●Get:-

▪Parameter estimates (β)

▪p-values, R^2, confidence intervals

▪Hypothesis tests for model significance

📌 **Use formulas:-**

```
Import statsmodels.formula.api as smf
Model = smf.ols('target ~ x1 + x2 + x1:x2 + I(x1**2)', data=df).fit()
```

🖼️ **5. Final slide desk**

●5–7 slides in PowerPoint or PDF

- Sections:

1. Problem Overview
2. Data Challenges (missingness, noise, schema issues)
3. Cleaning Pipeline
4. Feature Engineering
5. Model Results & Interpretation
6. (Optional) Limitations & Next Steps

Tools:- PowerPoint, Canva, Google Slides, or even nbconvert export.

📦 **Required Libraries-**

```
Pip install pandas numpy matplotlib
 seaborn statsmodels scipy
```

Optional-

Pip install jupyter pandas-profiling
scikit-learn openpyxl

✅ **Deliverables Recap**

File	Purpose
`data_prep.ipynb`	Cleaning, reshaping, and feature engineering
`modeling.ipynb`	Statistical modeling + interpretation
`cleaned_data.csv`	Final dataset used for modeling
`slides.pptx`	Summary of challenges, pipeline, results

✅ **1. Messy Synthetic Dataset (raw_data.csv)**

Theme:- Urban housing + energy usage (example use case)

Sample structure-

ID	City	Date	Energy_kWh	Temp
1	NYC	2022/01/01	120.5	3.2
2	la	Jan 5 2022	--	NA
3	Lon	2022-01-07	133.0	7.8
4	NYC	2022-01-08	140.1	4.1
...	...	...	...	...

Messiness includes:-

◇Inconsistent date formats

◇Categorical capitalization/noise

◇Missing and malformed numeric entries

◇Outliers (manually inserted)

◇Text noise in a “notes” column

◇Mixed case in Income_Level

✅ **2. Data_prep.ipynb Scaffold**

Includes:-

◇Imports and loading

◇Missingness overview

◇Data cleaning (commented sections)

◇Schema reshaping if needed

◇Feature engineering section (with stubs for interaction terms, polynomial features)

◇Marked TODOs where your logic/choices go

✅ **3. Modeling.ipynb with Example Regression**

◇Load cleaned data

◇Fit linear regression via statsmodels

◇Show parameter estimates and confidence intervals

◇Perform hypothesis tests

◇Interpret coefficients in markdown cells

◇Plot residuals and diagnostics

✅ **4. PowerPoint Slide Deck Template (final_summary_slides.pptx)**

Sections:-

1. Project Introduction
2. Raw Data Challenges (with placeholders for screenshots)
3. Data Cleaning Pipeline
4. Feature Engineering Summary

5. Modeling Results + Coefficient Table

6. Conclusions & Future Work

•Modern, clean, minimal style with editable text boxes and suggested chart slots.