

□ **Practical 1: Excel Data Analysis**

1. **Conditional Formatting** – Highlight cells greater than a value using Conditional Formatting > Highlight Cell Rules > Greater Than.
 2. **Pivot Table** – Use Insert > PivotTable, drag columns to rows and values to analyze data.
 3. **VLOOKUP** – Use formula like `=VLOOKUP(value, table, column_number, TRUE/FALSE)` to fetch data.
 4. **Goal Seek (What-if Analysis)** – Use Data > What-If Analysis > Goal Seek to find input for a desired output.
-

□ **Practical 2: DataFrames & Preprocessing**

1. **Read CSV/JSON** – Use `pandas.read_csv()` or `pandas.read_json()`.
 2. **Handle Missing Data** –
 - Fill missing: `df.fillna(0)`
 - Drop missing: `df.dropna()`
 3. **Data Manipulation** –
 - Filter: `df[df['col'] == value]`
 - Sort: `df.sort_values(by='col')`
 - Group: `df.groupby('col').mean()`
-

□ **Practical 3: Feature Scaling & Dummification**

1. **Scaling** –
 - MinMax: `MinMaxScaler()`
 - Standard: `StandardScaler()`

2. Dummification (Categorical \rightarrow Numeric) – Use

`LabelEncoder()` or `pd.get_dummies()`.

□ Practical 4: Hypothesis Testing

1. **T-test** – Compare two groups using
`scipy.stats.ttest_ind(group1, group2)`.
 2. **Chi-square** – Use for categorical data using
`scipy.stats.chi2_contingency()`.
-

□ Practical 5: ANOVA

1. **One-way ANOVA** – Use `scipy.stats.f_oneway()` to compare more than two groups.
 2. **Tukey's Test** – Post-ANOVA test using
`pairwise_tukeyhsd()` to see which groups differ.
-

□ Practical 6: Regression

1. **Linear Regression** – Predict values using
`LinearRegression()` from `sklearn`.
 2. **Metrics** – Use `mean_squared_error()` and `r2_score()` to evaluate model.
-

□ Practical 7: Logistic Regression & Decision Tree

1. **Logistic Regression** – For binary classification.

2. **Decision Tree** – Use `DecisionTreeClassifier()` for predictions.
 3. **Metrics** – Use accuracy, precision, recall, and classification report.
-

□ **Practical 8: K-Means Clustering**

1. **Preprocess** – Scale data with `MinMaxScaler()`.
 2. **Cluster** – Use `KMeans(n_clusters=k)` and apply Elbow method to find best k.
-

□ **Practical 9: PCA (Principal Component Analysis)**

1. **Standardize** – `StandardScaler()`
 2. **Apply PCA** – Use `PCA()` and plot explained variance to reduce dimensions.
-

□ **Practical 10: Data Visualization & Storytelling**

1. **Scatter Plot** – `plt.scatter(x, y)`
2. **Bar Chart** – `sns.countplot()`
3. **Heatmap** – `sns.heatmap(correlation_matrix)`
4. **Storytelling** – Describe what the charts show.