# Data Science Project Flow

**1. Data Loading/Collection and Data Understanding**

* Reading data from various formats: CSV, Excel, JSON, SQL, etc.
  + pd.read\_csv(), pd.read\_excel(), pd.read\_json(), pd.read\_sql()
* Obtain the dataset from sources (databases, files, APIs)

**2. Data Cleaning**

* Handling missing values or NA
  + df.dropna(), df.fillna()
  + Remove Rows with Missing Values: df.dropna
  + Remove Columns with Missing Values: df.dropna(axis=1)
  + Impute missing values
  + with a constant - df.fillna(0, inplace=True)
  + with mean (if there are no outliers)
  + with median (if there are outliers)
  + with mode (for categorical data)
  + Conditional Imputation - Apply imputation only for specific groups in a column
  + Forward or backward filling - propagate the last or next valid observation forward
  + Interpolate Missing Values - Used for continuous or time series data (Linear method)
* Removing duplicates
  + df.drop\_duplicates()
  + Axis = 0 and Axis = 1
* String operations
  + df.str.split(), df.str.contains(), df.str.replace()
* Type conversion
  + df.astype()
  + Converting data types for consistency (e.g., pd.to\_datetime(), astype()).
* Renaming columns
  + df.rename()
* String Manipulation:
  + Handling text data, stripping spaces, and regular expressions for cleaning.
  + df['column'].str.replace(), str.split().
* Merging and Joining Data:
  + Combining datasets using merge, join, and concatenate functions (df.merge(), df.join()).
* Concatenation: Stacking data horizontally or vertically.
  + pd.concat().
* Reshaping data
  + df.melt(), df.stack(), df.unstack()

**3. Exploratory Data Analysis (EDA)**

* Understanding the data structure and content
  + df.info(), df.describe(), df.shape(), df.dtypes
  + df.head(), df.tail(), df.columns
* Accessing specific rows/columns
  + df.iloc[], df.loc[], df[]
* Filtering data
  + df[df['column'] > value]
* Sorting data
  + df.sort\_values()
* Grouping data
  + df.groupby()
* Checking for correlations between features, covariance also
  + df.corr()
  + Heat map to show Correlation
* Detecting outliers
  + Box Plots,
  + Z-scores,
  + IQR method
* Checking data types
  + df.dtypes
* Handling missing data
  + df.isna(), df.dropna(), df.fillna()
* Univariant,
  + Bar, Distribution and Pie Plots
* Bivariate
  + Numerical v/s Numerical - Scatter Plots
  + Categorical v/s categorical - Count Plots
* Multivariate analysis
  + Two Categorical, one numerical

**4. Data Preprocessing**

* Scaling and Normalization
  + Min-Max Scaling
  + Standard Scaling (Z-score)
  + Max Abs Scaling
  + Robust
  + Quantile
  + Power
* Encoding Categorical Data
  + Label Encoding
  + One-Hot Encoding
  + Binary
  + Target
  + Frequency
  + Ordinal
* Handling Outliers
  + Using IQR
  + Using Z Scores
  + Winsorizing
  + Log Tranformation
  + Capping
* Handle Imbalanced data - Techniques like oversampling or undersampling

**5. Data Transformation**

* Feature Transformation:
  + Log transformation,
  + Power transformation

**6. Data Aggregation and Summarization**

* Grouping and Aggregating:
  + Using groupby() for summarizing data (e.g., sum, mean).
* Pivot Tables and filtering
  + Creating pivot tables for multi-dimensional data.
  + pd.pivot\_table()
* Summarizing statistics
  + df.mean(), df.median(), df.mode(), df.std()
* Cross-tabulations
  + pd.crosstab()

**7. Feature Engineering/Feature Selection**

* Handling categorical columns (Encoding)
* Handling Numeric columns (Scaling)
* Handling Outliers
* Devire features
  + Transforms for group by (Transform vs aggregation)
  + Transforms for Rank
  + Transforms for Group-wise Differences
* Interaction features
* Binning / Discretization
  + Equal Width
  + Equal Frequency
* Time-Based Features
* Polynomial Features
* Feature Selection (Statistical Methods)
* Dimensionality Reduction (PCA)

**8. Data Visualization**

* Visualizing data distributions
  + df.plot(), seaborn.histplot(), sns.kdeplot()
* Scatter plots, bar plots, line plots, etc.
  + df.plot.scatter(), df.plot.bar(), df.plot.line()
* Customizing plots
  + plt.title(), plt.xlabel(), plt.ylabel()
* Analyzing distributions and patterns
  + df.hist(), sns.pairplot(), df.plot(kind='box')