Essential Python Libraries for ML:

1. Pandas (Data Handling)

- **Loading Data:** `pd.read_csv`, `pd.read_excel`
- **Core Structures:** DataFrame (tables) vs. Series (columns)
- **Inspection:** `head()`, `shape`, `info()` (check types), `describe()` (stats)
- **Selection:** `iloc` (select by position) vs. `loc` (select by condition)
- **Analysis:** `value_counts()` (check class balance), `isnull().sum()` (find missing data), `corr()` (correlations)
- **Resource:** https://www.youtube.com/watch?v=2uvysYbKdjM&t=81s

2. NumPy (Numerical Operations)

- **Creating Arrays:** `np.array`, `np.zeros`, `np.ones`
- **Inspection:** `shape`, `ndim`
- **Reshaping:** `reshape`, `-1` trick (auto-calculate dim), `flatten()`, `np.expand_dims()` (add batch dim)
- **Statistical Ops:** `mean`, `sum`, `argmax` (find predicted class index)
- **Matrix Math:** Dot Product (`np.dot` or `@`), Broadcasting
- **Random:** `np.random.rand`, `np.random.randn` (initialize weights)
- **Resource:** https://www.youtube.com/watch?v=QUT1VHiLmmI

3. Matplotlib (Visualization)

- **Basic Setup:** `import matplotlib.pyplot as plt`
- **Line Plot:** `plt.plot()` (Visualizing Loss/Accuracy curves over epochs)
- **Scatter Plot:** `plt.scatter()` (Visualizing data clusters or regression points)
- **Histogram:** `plt.hist()` (Checking if data is normal/Gaussian distributed)
- **Subplots:** `plt.subplots()` (Displaying multiple images or graphs side-by-side)
- **Labels & Legends:** `plt.title()`, `plt.xlabel()`, `plt.ylabel()`, `plt.legend()` (Crucial for report clarity)
- **Display:** `plt.show()`
- **Resource:** `https://www.youtube.com/watch?v=3Xc3CA655Y4`

4. Scikit-Learn (Preprocessing & Model Building)

- **Splitting Data:** `train_test_split` (Separate Training vs. Testing data)
- **Preprocessing (Scaling):** `StandardScaler` (Z-score), `MinMaxScaler`
- **Preprocessing (Encoding):** `LabelEncoder` (Text to numbers)
- **Handling Missing Data:** `SimpleImputer`
- **The Workflow Pattern:** `.fit()` (learn patterns) vs. `.transform()` (apply patterns) vs. `.predict()` (make guesses)
- **Evaluation:** `accuracy_score`, `confusion_matrix`, `classification_report`
- **Resource:** https://www.youtube.com/watch?v=0B5eIE_1vpU