Data Acquisition and Preprocessing Strategy for Data Analytics Using Python

# 1. Task Objective

This document outlines a strategic approach for acquiring and preprocessing data for a Python-based data analytics project. It includes identifying relevant data sources, ensuring data quality, and implementing robust cleaning and transformation techniques to prepare the data for analysis.

# 2. Data Sources and Selection Criteria

Potential public data sources include:  
- Kaggle Datasets  
- UCI Machine Learning Repository  
- Government Open Data Portals (e.g., data.gov)  
- Company CRM or web scraping (if applicable)  
  
Selection Criteria:  
- Relevance to the project goal  
- Completeness and granularity  
- Data freshness and update frequency  
- Licensing and accessibility

# 3. Data Extraction and Expected Challenges

Data Extraction Methods:  
- Direct CSV download  
- API access (e.g., OpenWeatherMap, Twitter API)  
- Web scraping using Python libraries like BeautifulSoup or Selenium  
  
Expected Challenges:  
- API rate limits  
- Missing or inconsistent formats  
- Encoding and character set issues  
- Unstructured formats (PDFs, images)

# 4. Ensuring Data Quality

Steps to ensure data quality include:  
- Handling missing values (drop, mean/mode/median imputation)  
- Detecting and treating outliers using IQR or Z-score methods  
- Removing duplicate entries  
- Standardizing column names and formats  
- Consistency checks across columns (e.g., logical dependencies)

# 5. Python Libraries and Preprocessing Methods

Selected Libraries:  
- Pandas: For data manipulation and cleaning  
- NumPy: For numerical operations and arrays  
- Scikit-learn: For preprocessing like scaling and encoding  
  
Justification:  
These libraries are well-documented, widely used in the industry, and provide powerful functions for preprocessing tasks.

# 6. Timeline for Preprocessing Phases (Approx. 8–10 Hours)

- Data Source Identification: 1 hour  
- Data Extraction: 2 hours  
- Initial Cleaning (nulls, duplicates): 2 hours  
- Feature Transformation & Encoding: 2 hours  
- Final Validation and Export: 2 hours

# 7. Preprocessing Workflow (Pseudocode)

1. Import necessary libraries  
2. Load data into Pandas DataFrame  
3. Explore data (df.info(), df.describe())  
4. Handle missing values  
5. Remove duplicates  
6. Standardize data formats  
7. Encode categorical variables  
8. Scale numeric features  
9. Save cleaned data for analysis