Action Plan for Sentiment Analysis of Text Reviews Project

1. Project Goal Definition

The primary objective of this project is to build a machine learning model that can accurately classify the sentiment of text reviews into three categories:

- **Positive**
- **Negative**
- **Neutral**

This project aims to apply Natural Language Processing (NLP) and machine learning techniques to analyze customer opinions and extract meaningful insights from textual data.

2. Data Acquisition

Dataset Selection:

The dataset will be selected from publicly available sources such as:

- **Amazon Product Reviews Dataset**
- **Yelp Reviews Dataset**

Steps to Acquire Data:

1. Visit the dataset source (Kaggle or official dataset website).
2. Download the dataset in CSV or text format.
3. Store the dataset in the project directory.
4. Load the dataset into Python using Pandas for further analysis.

3. Environment Setup

Software and Tools:

- Operating System: Windows / macOS / Linux
- Programming Language: Python (version 3.x)
- Jupyter Notebook for development

Required Libraries:

- Pandas
- NumPy
- NLTK or SpaCy
- Scikit-learn
- Matplotlib / Seaborn
- WordCloud (optional)
- Streamlit or Flask (optional web application)

Setup Steps:

1. Install Python and Jupyter Notebook.
2. Install required libraries using `pip install -r requirements.txt`.
3. Configure NLP libraries (download stopwords, tokenizer models).

4. Exploratory Data Analysis (EDA)

EDA will be performed to understand the structure and characteristics of the dataset.

Planned steps:

- Inspect dataset shape and data types.
- Display sample reviews from each sentiment class.
- Analyze the distribution of sentiment labels using bar charts or pie charts.
- Calculate statistics such as:
    - Average review length
    - Distribution of review lengths
    - Vocabulary size

- Identify missing or duplicate values.

5. Text Preprocessing

The raw text data will be cleaned and prepared using the following steps:

- Convert all text to lowercase.
- Remove punctuation, numbers, and special characters.
- Tokenization (split text into words).
- Remove stop words (common words like “the”, “is”, “and”).
- Apply stemming or lemmatization to reduce words to their root form.
- Reconstruct cleaned text for feature extraction.

6. Feature Extraction

To convert text into numerical features, the following techniques will be used:

- **TF-IDF (Term Frequency–Inverse Document Frequency)** for representing word importance.
- (Optional) **Word Embeddings** such as Word2Vec or GloVe for dense vector representations.

These features will serve as input to machine learning models.

7. Model Selection

The following machine learning models will be implemented and compared:

- Naive Bayes (Multinomial Naive Bayes)
- Support Vector Machine (SVM)
- Logistic Regression

These models are chosen due to their effectiveness in text classification tasks.

8. Model Training and Evaluation

Training:

- Split dataset into training and testing sets (80% training, 20% testing).
- Train each selected model on the training dataset.

Evaluation Metrics:

- Accuracy
- Precision
- Recall
- F1-score
- Confusion Matrix
- Classification Report

The performance of each model will be compared to identify the best-performing model.

9. Model Optimization (Optional)

Hyperparameter tuning will be performed to improve model performance using:

- GridSearchCV or RandomizedSearchCV

Examples:

- Tuning regularization parameters in Logistic Regression.
- Adjusting kernel and C values in SVM.

10. Web Application Development (Optional)

A simple web application will be developed to allow users to input text and receive sentiment predictions.

Planned steps:

1. Choose framework (Streamlit or Flask).
2. Create a user interface with a text input box and predict button.

3. Load trained model and TF-IDF vectorizer.
4. Apply preprocessing to user input.
5. Predict sentiment and display result (Positive/Negative/Neutral).

11. Documentation and Reporting

Documentation will be maintained throughout the project:

- Jupyter Notebook with code and comments.
- Research report explaining methodology and findings.
- Action plan document.
- Final results summary with charts and tables.
- README file for GitHub repository with setup instructions.

12. Timeline

Phase	Task	Duration
Week 1	Project planning & data acquisition	2 days
Week 1	Environment setup & library installation	1 day
Week 2	Exploratory Data Analysis (EDA)	2 days
Week 2	Text preprocessing	2 days
Week 3	Feature extraction (TF-IDF, embeddings)	2 days
Week 3	Model training & evaluation	2 days
Week 4	Model optimisation	1 day
Week 4	Web application development	2 days
Week 4	Documentation & GitHub setup	2 days

13. GitHub Repository Setup

Steps:

1. Create a GitHub repository named **SentimentAnalysisReviews**.
2. Upload:
    - Research Report
    - Action Plan document
    - Jupyter Notebook or scripts
    - Web application code (optional)
    - requirements.txt
    - README.md
3. Use Git for version control:
    - Commit changes regularly.
    - Maintain clear commit messages.
4. Share repository link as final deliverable.

Conclusion

This action plan provides a structured roadmap for completing the Sentiment Analysis of Text Reviews project. By following the defined steps, the project will successfully demonstrate the use of NLP and machine learning techniques for text classification, along with proper documentation and version control using GitHub.