AI Lab Project Report

Course: Artificial Intellegence **Instructor: Faiza Iqbal**

1. **M.Warid Ali (24108119)**
2. **Aimash Waheed (24108104)**
3. **Ammar Bin Yasser (24108105)**

1. Introduction

Artificial Intelligence (AI) is widely used today in language understanding, medical support systems, chatbots, and automated classification. In this project, our goal was to build a text-based classification model that predicts whether a given symptom description belongs to Class 0 or Class 1. Since the dataset provided by the instructor contained only numeric labels without medical meaning, our focus remained on the technical process rather than medical interpretation.

The complete workflow included:

1. Data Preprocessing
2. Exploratory Data Analysis (EDA)
3. Feature Extraction
4. Model Training
5. Model Evaluation
6. Final Predictions

Through these steps, we developed an NLP pipeline capable of classifying short symptom texts using Machine Learning techniques.

2. Dataset Description

The dataset provided by the instructor included:

1. Training dataset
2. Validation dataset
3. Test dataset

Each file contained two main columns:

text — the raw symptom description

labs — the class label (only 0 and 1 found in the dataset)

3. Task 1 — Data Preprocessing

Preprocessing is essential because raw text often contains noise such as punctuation, special symbols, unnecessary words, and inconsistent formatting. Machine learning models cannot understand such messy text directly.

The following preprocessing steps were performed:

1. Lowercasing

All words were converted to lowercase.

2. Removing unwanted characters

We removed:

1. Punctuation
2. Numbers
3. Extra spaces
4. URLs
5. Email-style patterns
6. Non-alphabetic characters

3. Stopword Removal

Common English words like:

“the”, “is”, “and”, “was”, “are”

4. Tokenization

Text was divided into individual words (tokens).

5. Lemmatization

Words were converted to their base form.

6. Saving Clean Datasets

After preprocessing, three new files were generated:

1. train_clean.csv
2. val_clean.csv
3. test_clean.csv

These cleaned datasets were used for EDA and model training.

4. Task 2 — Exploratory Data Analysis (EDA)

Exploratory Data Analysis helps us understand the dataset before building a model.

The following analyses were performed:

1. Class Distribution

We counted how many samples existed for each label (0 and 1).

2. Bar Plot for Class Counts

A simple bar chart was created to visually represent the number of examples in each class.

3. WordCloud Visualization

A WordCloud was generated using the entire cleaned text.

It shows the most frequently occurring words in a visually appealing way.

4. N-Gram Analysis

N-grams are word patterns:

1. Unigrams — 1 word (e.g., “stress”)
2. Bigrams — 2-word phrases (e.g., “feeling sad”)
3. Trigrams — 3-word phrases (e.g., “panic attack disorder”)

Top 10 most frequent n-grams were extracted for each category.

5. POS Tagging (Part-of-Speech Analysis)

POS Tagging assigns grammatical labels such as:

NOUN, VERB, ADJECTIVE

We extracted POS tags from a sample of the cleaned text.

The count of each POS category was calculated.

This helped understand the grammatical structure of the dataset.

A separate WordCloud of nouns only was also created to highlight important subjects mentioned in the symptoms.

This completes Task 2 by giving a deeper understanding of the dataset.

5. Task 3 — Model Training and Evaluation

After cleaning and understanding the data, the next step was to build a machine learning model capable of predicting whether a symptom description belongs to class 0 or class 1.

The following steps were performed:

1. Splitting Text and Labels

From the clean dataset:

X (features) = clean_text

y (labels) = labs

2. TF-IDF Feature Extraction

Text must be converted into numerical form for machine learning.

We used TF-IDF (Term Frequency – Inverse Document Frequency) for this purpose.

TF-IDF assigns importance scores to words based on:

1. How often a word appears in a document
2. How rare it is across all documents

3. Model Selection — Logistic Regression

Logistic Regression was chosen because:

1. It works extremely well for text classification
2. It is fast and interpretable
3. It performs well in binary classification tasks
4. It generalizes well for small-to-medium-sized datasets

During training, the model learned which words and patterns belong to class 0 and which belong to class 1.

4. Model Evaluation

The model's performance was measured using:

1. Accuracy — percentage of correctly predicted labels.
2. Precision — how many predicted labels were actually correct.
3. Recall — how many actual labels were correctly identified.
4. F1-Score — balance between Precision and Recall.
5. Confusion Matrix — shows how many class 0 and class 1 samples were correctly and incorrectly predicted.

This matrix allowed us to see if the model was confusing one class more than the other.

5. Prediction on Test Dataset

Finally, the trained model was applied to the test data.

A new column predicted_label was created, containing the predicted class (0 or 1) for each test sample.

The final output file was saved as:

'test_with_predictions.csv'

This completes Task 3.

6. Limitations

1. The dataset contains only two classes, limiting complexity.
2. Class meanings were not provided, restricting analysis.
3. The dataset size is moderate; a larger dataset may improve accuracy.
4. More complex models (like BERT) could achieve better results.
5. Medical interpretation is not possible without label definitions.

7. Future Work

1. Use deep learning models (BERT, RoBERTa).
2. Collect more diverse symptom datasets.
3. Expand from binary classification to multi-class classification.
4. Add semantic analysis and medical meaning for labels.
5. Improve preprocessing with domain-specific cleaning strategies.

8. Conclusion

In this project, we built a complete end-to-end symptom classification system using NLP and machine learning. The workflow included preprocessing, EDA, feature extraction using TF-IDF, training a logistic regression classifier, evaluating its performance, and generating predictions for unseen test data.

The project demonstrated that NLP techniques, combined with classical machine learning models, can effectively classify short medical symptom descriptions, even when labels are limited to simple numeric categories.

This completes the AI Lab Final Project with all required tasks successfully implemented.