**I, am doing my bachelor in software engineering from Mehran University Jamshoro. In this Degree program we have learned what is software engineering, there key concepts, and most importantly coding concept. We have learned multiple languages yet for example: Java, C++, Phython, SQl, Fluter, Data Science and Analysis, Agent Based Learning, Operating system and Data structure and algorithum. we have made multiple projects in our degree program that gave us key concepts of subjects. So this my final semester and my Final Year project is: Pulmonary Disease Detection and Classification Using Deep Learning Network. In this we used AI concept and build the model (RNN deep learning model) to detect and classified pulmonary diseases from lung sound. so in the end I have to write that my core interest now is to work on data science field and experience as much as i can. and make projects that will cover core concepts of Data Sciences field.**

**Report: Email Spam Classification**

**1. Introduction**

Email spam detection is an important real-world problem that helps in filtering unwanted and potentially harmful messages from a user’s inbox. In this project, the goal was to build a machine learning model that can classify messages as either **spam** (unwanted messages) or **ham** (legitimate messages). This task is a classic example of **binary classification** and is widely used in the field of Natural Language Processing (NLP).

**2. Methodology**

The project was completed step by step as follows:

**Step 1: Data Cleaning and Preprocessing**

* The dataset contained two main columns: **Category** (spam or ham) and **Message** (the text of the email).
* A preprocessing pipeline was applied to clean the text:
  + Converted all words to lowercase.
  + Removed special characters, numbers, URLs, and punctuation.
  + Tokenized text into words.
  + Removed common stopwords (e.g., “the”, “is”, “and”).
  + Finally, applied lemmatization to reduce words to their base form.
* The cleaned text was stored in a new column called **clean\_text**.

This step ensured that the raw text was transformed into a more structured and machine-readable form.

**Step 2: Vectorization (Converting Text to Numbers)**

* Since machine learning models cannot process raw text, the cleaned text was converted into numerical features using two approaches:
  + **Bag of Words (CountVectorizer)**
  + **TF-IDF Vectorizer (Term Frequency–Inverse Document Frequency)**
* For the main experiments, TF-IDF was chosen because it not only counts words but also gives higher weight to words that are more important for distinguishing spam from ham.
* The final feature matrix had **3000 features (words)**.

**Step 3: Model Building**

Three machine learning algorithms were trained on the TF-IDF vectors:

1. **Multinomial Naive Bayes** (commonly used for text classification).
2. **Logistic Regression** (a widely used linear classifier).
3. **Support Vector Machine (SVM)** (effective in high-dimensional text data).

The dataset was split into **80% training** and **20% testing** sets. Each model was trained on the training set and then tested on unseen data.

**Step 4: Model Evaluation**

The models were evaluated using the following metrics:

* **Accuracy**: overall percentage of correct predictions.
* **Precision**: proportion of predicted spam messages that were actually spam.
* **Recall**: proportion of actual spam messages correctly identified.
* **F1-Score**: balance between precision and recall.
* **Confusion Matrix**: detailed view of true positives, true negatives, false positives, and false negatives.

**3. Results**

**Naive Bayes**

* Accuracy: **97.5%**
* Strength: Perfect recall for ham (1.00).
* Weakness: Lower recall for spam (0.82), meaning some spam emails were missed.
* Confusion Matrix: 955 ham correct, 131 spam correct, 28 spam misclassified.

**Logistic Regression**

* Accuracy: **97.2%**
* Strength: High precision for spam (0.98).
* Weakness: Same issue as Naive Bayes with spam recall (0.82).
* Confusion Matrix: 953 ham correct, 130 spam correct, 29 spam misclassified.

**Support Vector Machine (SVM)**

* Accuracy: **98.7%** (highest).
* Strength: Best balance between precision (0.98) and recall (0.93) for spam.
* Weakness: Very few misclassifications overall.
* Confusion Matrix: 952 ham correct, 148 spam correct, 11 spam misclassified.

**4. Conclusion**

All three models performed strongly, with accuracies above 97%. However, **SVM clearly outperformed the other models**, achieving the highest accuracy (**98.7%**) and the best balance between precision and recall. This means SVM is more reliable at correctly detecting spam while minimizing false alarms.

In real-world spam detection, **recall is especially important** because missing spam emails can expose users to security risks. Based on this project, **SVM is the recommended model** for spam classification.