jNAME: ASHNA

REGISTRATION NUMBER-25BSA10138

1.INTRODUCTION:

This project signifies the importance of artificial intelligence and machine learning concepts to solve the real world problem like email spam detection.We use the concept of Naives Bayes and Bayesian theorem and its algorithm due to its efficiency, accuracy and speed in word classification.

2. PROBLEM STATEMENT:

To build a highly-efficient classification system that separates incoming emails into spam or Ham.

3.Functional Requirement(FR)

FR1 : Data Input and processing: input and clean a labeled dataset, and then giving string labels('ham', 'spam') to integers(0, 1)

FR2 : Feature Engineering: transform raw text data into numerical features using the CounterVectorizer.

FR3 Classification and Prediction: Training of Naïve Bayes model and give predictions on testing dataset.

4.Non-Functional Requirement

NFR1 : Performance1: training time very fast, within 5 seconds due to the simplicity of Naive Bayes.

NFR2 : Usability: Concise, simple, clear based output showing all results and predictions which are live.

NFR3 : Maintainibility: with the help of distinct functions available in modular python for each steps like Preprocessing, Vectorization, Training, Evaluation).

6.SYSTEM ARCHITECTURE: This architecture follows a series machine learning pipeline:

1.Data input and Processing

2. Feature Extraction(CountVectorizer)

3.Model Training(Multinomial Naïve Bayes)

4. Assessment and Prediction

7.Workflow Diagram

This whole process involves a direct flow: 1.Load Data 2.Split 3.Fit Vectorizer (Train Data) 4.Transform Data 5. Train Model 6. Predict 7. Assessment.

Component Diagram

This system is made up of:Pandas(Data), CountVectorizer(Feature Engineering), MultinomialNaives Bayes(Model), Scikit-learn Metrices which is the output.

8.Design Decisions and rationale:1. Algorithm: Multinomial naïve Bayes is selected for its advantages for count-based data, accuracy and strong performace in text classification.

2.Vectorization: provide a simple, high-dimensional representation of text frequencies

9. Implementation Details: 1.Implemented in modular, single python script i.e (spam_classifier.py)

10. Screenshots

11. Testing Approach

By the use of hold-out-cross-validation which is 70% as train data and 30 % as test data. Testing includes a functional demo and formal assessment of the test data, coded spam and ham messages.

12.Challenges Faced

Preventing data leakage by confirming the vectorize’s vocabulary was learned from the training data and interpreting the difference between the classification metrices.

13.Learning

Learning the core ML pipeline, and interpreting text to data conversion (NLP), and successfully implementing and evaluated a probablilistic classification model.

14.ENHANCEMENTS IN FUTURE

1.Implementing TF-IDF vectorization for feature weighting.

2.Analyze naïve bayes performance against logistic regression or Support Vector Machines

3.Tune the Naïve Bayes smoothing parameter for optimal results.

15. References

1.Scikit-learn Documentation for components machine learning

2. Dataset