Assignment No:2

Title:

Classify the email using the binary classification method. Email Spam detection has two states: a) Normal State – Not Spam, b) Abnormal State – Spam. Use K-Nearest Neighbors and Support Vector Machine for classification. Analyze their performance.

Dataset link: The emails.csv dataset on the Kaggle

Experiment Objective:

The objective of this lab is to classify emails into two states: "Normal State" (Not Spam) and "Abnormal State" (Spam) using binary classification methods. We will implement K-Nearest Neighbors (KNN) and Support Vector Machine (SVM) algorithms for classification and analyze their performance using various evaluation metrics.

Dataset Description:

The csv file contains 5172 rows, each row for each email. There are 3002 columns. The first column indicates Email name. The name has been set with numbers and not recipients' name to protect privacy. The last column has the labels for prediction : 1for spam, 0 for not spam. The remaining 3000 columns are the 3000 most common words in all the emails, after excluding the non-alphabetical characters/words. For each row, the count of each word (column) in that email(row) is stored in the respective cells. Thus, information regarding all 5172 emails are stored in a compact dataframe rather than as separate text files.

Link:https://www.kaggle.com/datasets/balaka18/email-spam-classification-dataset-csv

Tools and Libraries:

We will use Python programming language and the following libraries:

- Pandas: For data manipulation and preprocessing.
- Scikit-learn: For implementing KNN and SVM classifiers.
- Matplotlib/Seaborn: For data visualization and result analysis.

Theory:

1. **Supervised Learning - Binary Classification:**
    - In binary classification, the goal is to classify instances into one of two classes or categories. In this case, we aim to classify emails as either "Normal State" (not spam) or "Abnormal State" (spam).
2. **Email Spam Detection:**
    - Email spam detection involves the automatic identification of unsolicited and unwanted email messages that are sent in bulk. The objective is to distinguish between legitimate (normal) emails and spam emails.
3. **K-Nearest Neighbors (KNN):**
    - KNN is a non-parametric classification algorithm that assigns a class label to a data point based on the majority class labels of its k nearest neighbors. It measures the distance between instances to determine similarity.
4. **Support Vector Machine (SVM):**
    - SVM is a powerful supervised learning algorithm used for binary classification. It separates classes by finding the optimal hyperplane that maximizes the margin between the classes.
5. **Dataset:**
    - The "emails.csv" dataset available on Kaggle is used for this experiment. It contains a collection of emails along with their corresponding labels indicating whether they are spam or not. The dataset consists of features (e.g., email content, subject, sender) and the target variable (spam or not spam).
6. **Data Preprocessing:**
    - Preprocess the dataset by loading it into a suitable data structure (e.g., Pandas DataFrame).
    - Perform any necessary data cleaning steps, such as removing duplicates or irrelevant columns.
    - Convert categorical variables into numerical representations using techniques like one-hot encoding or label encoding.
    - Split the dataset into training and testing sets for model evaluation.

7. **Model Training and Evaluation:**
    - Implement KNN and SVM classification models using appropriate libraries (e.g., scikit-learn).
    - Train the models using the training set.
    - Make predictions on the testing set using the trained models.
    - Evaluate the models' performance using various metrics:
        - Accuracy: The proportion of correctly classified instances.
        - Precision: The ability of the model to correctly classify positive instances (spam emails).
        - Recall: The ability of the model to correctly identify all positive instances.

Implementation steps.

Software required:

Anaconda with Python 3.7

1. **Preprocessing the Dataset:**
    - Download the "emails.csv" dataset from Kaggle.
    - Load the dataset into a Pandas DataFrame.
    - Perform necessary data cleaning steps, such as removing duplicates or irrelevant columns.
    - Convert categorical variables into numerical representations using one-hot encoding or label encoding.
    - Split the dataset into training and testing sets (e.g., 80% for training and 20% for testing).
2. **Implementing K-Nearest Neighbors (KNN):**
    - Import the required libraries, such as scikit-learn.
    - Create an instance of the KNN classifier.
    - Train the KNN model using the training data.

- Make predictions on the testing data.
- Evaluate the KNN model's performance using accuracy, precision.

3. **Implementing Support Vector Machine (SVM):**
    - Import the necessary libraries, such as scikit-learn.
    - Create an instance of the SVM classifier.
    - Train the SVM model using the training data.
    - Make predictions on the testing data.
    - Evaluate the SVM model's performance using accuracy, precision, recall, and F1 score.
4. **Comparing Performance:**
    - Compare the performance of the KNN and SVM models based on the evaluation metrics.
    - Analyze the results to determine which model performs better in classifying emails as spam or not spam.

Conclusion:

- Summarize the results and performance of both the KNN and SVM models.
- Discuss the effectiveness of each model in classifying email spam.