CLASSIFICATION MODELS REPORT

1. **Classifying Emails into Ham and Spam using Naïve Bayes Classifier.**

Article written by Varun Kumar,
The article explains how to build an email classifier using the Naive Bayes Classifier to classify emails as either ham (non-spam) or spam. Here's a summary of the key points

Model:
The classifier is based on Bayes' Theorem, which calculates the probability that an email is ham or spam given its content.

The classifier compares probabilities:
$P(\text{ham} \mid \text{bodyText}) > P(\text{spam} \mid \text{bodyText}) \rightarrow$return 'ham'

Key probabilities include:

- P(ham): The fraction of ham emails.
- P(spam): The fraction of spam emails.
- P(bodyText | spam) and P(bodyText | ham): The likelihood of words in the email's body being spam or ham.

Training Set Preparation**:**
Two MySQL tables are used for training:

- trainingSet: Contains email text and its category (ham or spam).
- wordFrequency: Stores each word, its count, and the category.

Issue with New Words:
For words unseen in training data, Laplace smoothing and logarithmic transformation are used to avoid zero probabilities.

Testing the Classifier:
Examples:

- Spam: "Scan Paytm QR Code to Pay & Win 100% Cashback"
- Ham: "Re: Applying for Fullstack Developer"

2. **Breast Cancer Classification using Support Vector Machines.**

This article is written by Adebola Lamidi, Published in Towards Data Science, Nov 22, 2018 demonstrates how to train and optimize an SVM model for breast cancer classification, improving accuracy with data normalization. It highlights the importance of understanding machine learning models, their application in health diagnostics, and the significant improvement that data preprocessing can provide in model performance. Here's a summary of the key points

Key Concepts:

- **Breast Cancer** is the most common cancer among women, and early diagnosis is crucial for survival. Machine learning techniques can enhance diagnostic accuracy.
- **Dataset**: It includes ten real-valued features describing cell nuclei, such as **radius, texture, perimeter, area,** and **smoothness**.

Support Vector Machine (SVM):

- **SVM** is a powerful classification algorithm that finds the optimal decision boundary separating classes by maximizing the margin between the closest data points.
- **Advantages**: SVM performs well in **high-dimensional spaces**, works with **both small and medium-sized datasets**, and can use **various kernels for different data types**.
- **Disadvantages**: SVM is prone to **overfitting** with **too many features** and does not directly provide probability estimates.

Modeling:

- **Feature Selection:** Features like mean radius, perimeter, and texture are used to predict whether a tumor is malignant or benign.
- **Training and Testing**: The dataset is split into training (80%) and testing (20%) subsets. The model is trained on the training data and tested on the unseen testing data.

Model Evaluation:

- **Initially**, the model's performance is **poor**, achieving only 34% accuracy, with many false predictions.
- The model is **improved** by **normalizing the data**, scaling feature values into the [0,1] range.
- After normalization, the model's accuracy significantly improves to **98%**.

3. **Naïve Bayes to Classify Movie Reviews Based on Sentiment.**

This article was written by Mohana Pranadeep potti an al., Published in IJCRT on 1 March 2018.
The article discusses sentiment analysis (SA) on movie reviews using the Naïve Bayes (NB) classifier. Sentiment analysis, or opinion mining, involves determining whether a given piece of text expresses positive, negative, or neutral sentiments. This study focuses on classifying movie reviews into positive and negative categories using the Naïve Bayes approach, a probabilistic technique.. Here's a summary of the key points

Key points:

- **Sentiment Analysis**: It is used to analyze emotions, opinions, or attitudes expressed in text. In the context of movie reviews, this can help classify reviews as positive or negative.

- **Naïve Bayes Classifier:** The article prefers the Naïve Bayes algorithm due to its probabilistic approach, which uses the probability of words being positive or negative in the context of a movie review.

- **Approach**: The paper uses a dataset of movie reviews. Each review is analyzed by counting the occurrences of positive and negative words. The Naïve Bayes classifier calculates the posterior probabilities for positive and negative sentiments using Bayes' Theorem.

- **Implementation**: Reviews are first classified into positive or negative categories based on their words' probabilities. Then, the most frequent sentiment (either positive or negative) in a set of reviews determines the final classification for that movie.

- **Other Methods Considered:** The article mentions other classification techniques, such as Support Vector Machine (SVM), K-Nearest Neighbors (K-NN), and Decision Trees. However, the Naïve Bayes classifier is found to perform better due to its simplicity and efficiency.

- **Results and Conclusion**: The study shows that Naïve Bayes outperforms other methods (SVM and K-NN) for sentiment classification in movie reviews. The method provides a robust and optimized approach to classify the reviews based on the overall sentiment expressed in the reviews.

- **Applications**: This sentiment analysis can help both movie producers and viewers. Producers can gauge public opinion on movies, while consumers can use reviews to make informed decisions about what to watch.

4. **Wine Quality Prediction using Decision Trees.**

This article focuses on predicting wine quality using a decision tree classifier, utilizing the Wine Dataset from the UC Irvine Machine Learning Repository. The goal is to train a machine learning model to predict wine quality based on various attributes. Here's a summary of the key points.

Packages and Libraries used:

- **Numpy**: Used for accurate **mathematical calculations.**
- **Pandas**: Used for **handling datasets** in file formats like CSV and Excel.
- **Scikit-learn:** Used for implementing the **machine learning mode**l, including the Decision Tree Classifier.
- **Matplotlib/Seaborn:** (implied for data visualization) Can be used for **plotting results**.

Steps Involved:

1. Installing Required Packages:

2. numpy for mathematical accuracy, pandas for handling datasets, and scikit-learn for the classifier are installed via terminal/command prompt.

3. Importing Required Modules:
    - **numpy, pandas** for data manipulation.
    - **train_test_split** for splitting data into training and testing sets.
    - **preprocessing** for data normalization.
    - **DecisionTreeClassifier** for prediction.

4. Loading Dataset:
    - The Wine Dataset is imported using **pd.read_csv()**, with a semicolon separator for structure.

5. Analyzing Data:
    - The dataset's first five entries are inspected using the **head()** function.

6. Separating Features and Labels:
    - Features (attributes) are stored in **X**, and the target label (wine quality) is stored in y**.**

7. Splitting Data into Train and Test Sets:
    - Using t**rain_test_split()** to divide the data into **training and testing sets**, with **80%** for training and **20%** for testing.

8. Data Preprocessing:
    - **Data normalization** is done, converting feature values into a range of -1 to 1 for easier interpretation by the algorithm.

9. Training the Model:
    - A **DecisionTreeClassifier()** is instantiated and trained using the **fit()** method on the preprocessed training data.

10. Model Evaluation:
    - The model's **accuracy** is measured using the **score()** function, which returns a confidence score of **62.1875%**.

11. Making Predictions:
    - After training, predictions are made on the test data using the **predict()** function.
    - The first five predicted labels (**y_pred**) are **compared** with the expected labels (**y_test**).

12. Performance Comparison:
    - The accuracy of the model is estimated by comparing predicted results with expected values. A result of **62.1875% accuracy** is **achieved**, indicating the model's reliability.