**Written Report**

Assignment: Group project (NLP)

Group number: #3

Course number: COMP 237 004

**Task #2: Result of the basic data exploration**

1. **Dataset Information:**

* The dataset contains 350 entries.
* There are 5 columns: COMMENT\_ID, AUTHOR, DATE, CONTENT, and CLASS.
* All columns are non-null, indicating there are no missing values in the dataset.

1. **Descriptive Statistics:**

For the CLASS column (the only numerical column):

* Count: 350 entries.
* Mean: 0.5, indicating a balanced distribution between the two classes.
* Standard Deviation: Approximately 0.5.
* Min and Max values are 0 and 1, suggesting binary classification (likely spam and not spam).

1. **Missing values:**

* There are no missing values in any of the columns.

1. **Class Distribution:**

* The distribution of the CLASS column is perfectly balanced, with 175 entries for each class (0 and 1).

1. **Sample Comments:**

* A random sample of comments provides insight into the nature of the content in the data set. These comments range from promotional messages to personal requests and discussions.

The CONTENT and CLASS columns are identified as the primary columns for the text classification task.

**Task #4: Highlights of the output**

**Shape of the data:**

* The transformed data has a shape of (350, 1310). This indicates that from the 350 comments in the dataset, CountVectorizer has extracted 1310 unique features (tokens). Each feature represents a distinct word or token present in the comments.

**Types of words:**

* The initial features include different tokens. As seen in the sample feature names like '00', '000', '02' and so on, these features include numerical values and some alphanumeric strings. This wide range of tokens is due to CountVectorizer treating any sequence of characters separated by whitespace as a valid token.
* The presence of such different tokens can be explained by the nature of online comments, where people often use numbers, URLs or special character combinations.

**Tokenization:**

* The default tokenization used by CountVectorizer includes all sorts of characters and does not differentiate between words and numbers. However, it might also include tokens that are not very useful for some text analysis tasks.

**Task #5: TF-IDF Transformation**

**Shape of the TF-IDF transformed data:**

* The shape of the data remains the same (350, 1310) after applying TF-IDF. This means we still have 350 comments (documents) and 1310 unique features (words or tokens), just like after the CountVectorizer step.

**Nature of the TF-IDF features:**

* The 1310 features are the same words or tokens as before, but now their values have changed. Each value represents the relative importance of that word in a comment, considering all comments in the dataset.
* Common words across all comments will have lower TF-IDF scores, as their high frequency makes them less significant for distinguishing between different comments.
* Unique or rare words in specific comments will have higher scores, making them more important for analysis.

**Usefulness:**

* The TF-IDF transformation is useful for text classification models because it highlights the words that are most distinctive for each comment. This can improve the performance of machine learning models by focusing on the most relevant features.

**Task #12: Results and Conclusions**

**Data Loading and Exploration:**

* Successfully loaded "Youtube01-Psy.csv" file with 350 comments.
* Dataset consists of 5 columns: COMMENT\_ID, AUTHOR, DATE, CONTENT, and CLASS.
* CONTENT (comments) and CLASS (spam or not spam) are key columns for analysis.
* No missing values in the data, ensuring completeness.

**Data Preparation:**

* Transformed comments into numerical data with CountVectorizer, resulting in 1310 unique features.
* Applied TF-IDF transformation, maintaining 1310 features, focusing on word importance.

**Model Building and Evaluation:**

* Trained a Multinomial Naive Bayes classifier on 75% of the data (262 training samples).
* Cross-validation on training data showed a mean accuracy of about 93.87%.
* Model tested on 25% of the data (88 testing samples) achieved an accuracy of 95.45%.
* Confusion matrix indicated good model performance with 42 true positives, 42 true negatives, 3 false positives, and 1 false negative.

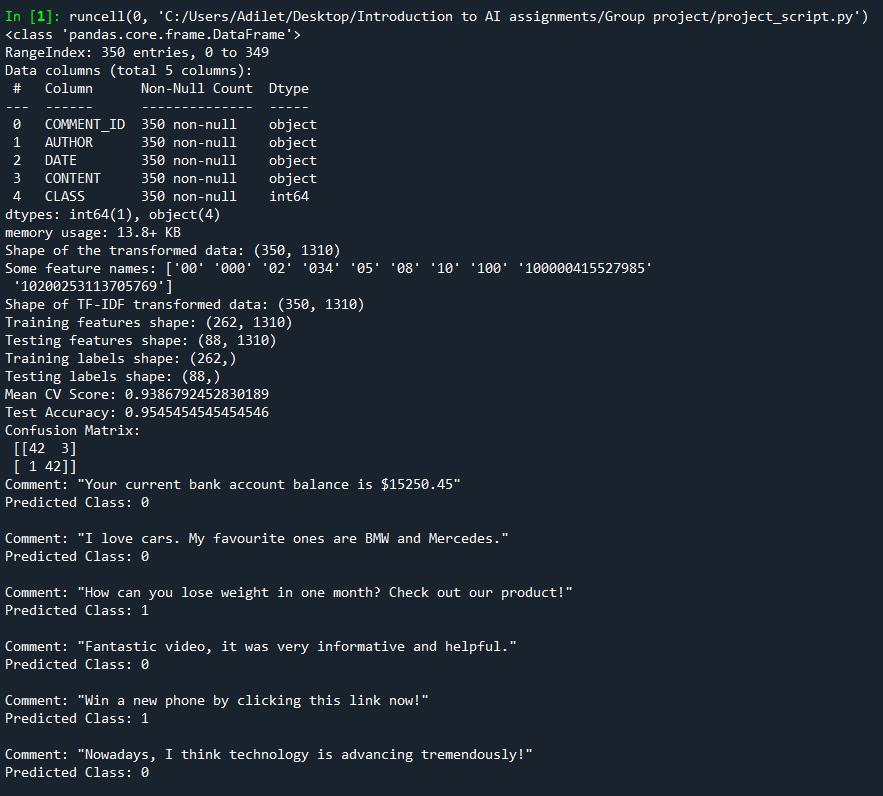
**Model Testing with New Comments:**

* Successfully identified both non-spam (bank account balance, love for cars, fantastic video, comment on technology) and spam comments (weight loss product, link to win a phone).

**Conclusions:**

* The Naive Bayes classifier is highly effective in classifying comments as spam or non-spam.
* High accuracy in both cross-validation and test sets indicates robust model performance.

**Output of the code:**

****

**Overall Conclusion:**

Our project demonstrates a strong application of text classification in identifying spam comments, with high accuracy and effectiveness. The results show that the model we built can reliably differentiate between spam and non-spam comments, making it a valuable tool for moderating or analyzing online content.