Final Project

Data Analysis and Visualization

Group Members:

Aroz Imran 21L-6246

Hashim Ali 21L-6230

Muneeb Bhatti 21L-6257

Title: Predicting YouTube Categories based on Titles

Statement of contribution to the project:

Scarping was done by all the group members in stages.

Aroz Imran:

Scaping and deep learning model(rnn,cnn)

Hashim Ali:

Scraping and machine learning (multinomial naive based)

Muneeb Bhatti:

Scraping and machine learning(gaussian)

1. Data Collection:

Web Scraping with Selenium:

Leveraging Selenium, our web scraping process efficiently navigated dynamic elements on YouTube, extracting video titles seamlessly. Utilized Selenium for web scraping YouTube titles and categories.Collected data points by navigating through YouTube pages and extracting information.

2. Data Preprocessing:

Text Cleaning:

Removed HTML tags, special characters, and irrelevant information.

Performed tokenization to break down titles into individual words.

Removed stop words to focus on meaningful content.

3. Feature Engineering:

Bag-of-Words (BoW):

Represented titles using the Bag-of-Words model.

Converted text data into **numerical vectors.**

4. Model Implementation:

Multinomial Naive Bayes:

Introduction:

Multinomial Naive Bayes is a probabilistic classification algorithm based on Bayes' theorem. It's well-suited for text classification tasks, including predicting YouTube categories based on titles. It assumes that the features (words in this case) are conditionally independent given the class label.

Implementation:

Data Preparation:

Features (X): Bag-of-Words representation of YouTube video titles.

Target Variable (y): YouTube video categories.

Train-Test Split:

Split the data into training and testing sets for model evaluation.

Model Initialization and Training:

Create an instance of the Multinomial Naive Bayes classifier and train it on the training data.

Prediction and Evaluation:

Make predictions on the test set.

Evaluate the model using accuracy and a detailed classification report.

Detailed Predictions and Exporting to CSV:

Extract titles with corresponding predictions.

Create a Data Frame with testing data and predictions.

Export predictions to a CSV file.

Logistic Regression:

Introduction:

Logistic Regression is a linear model for binary and multiclass classification. It models the probability of an instance belonging to a particular class. In this case, it's used to predict YouTube categories based on title features.

Implementation:

Data Preparation:

Features (X): Bag-of-Words representation of YouTube video titles.

Target Variable (y): YouTube video categories.

Train-Test Split:

Split the data into training and testing sets for model evaluation.

Model Initialization and Training:

Create an instance of the Logistic Regression model and train it on the training data.

Prediction and Evaluation:

Make predictions on the test set.

Evaluate the model using accuracy and a detailed classification report.

Detailed Predictions and Exporting to CSV:

Extract titles with corresponding predictions.

Create a Data Frame with testing data and predictions.

CNN (Convolutional Neural Network):

Introduction:

Convolutional Neural Networks are deep learning models commonly used for image classification but can also be adapted for sequential data like text. In this context, a CNN is applied to capture local patterns in the Bag-of-Words representation of YouTube titles.

Implementation:

Data Preparation:

Features (X): Bag-of-Words representation of YouTube video titles.

Convert the text data into numerical vectors, possibly using embeddings.

Model Architecture:

Design a CNN architecture with convolutional layers to capture local patterns, followed by pooling layers to reduce spatial dimensions.

Training:

Split the data into training and testing sets.

Train the CNN model using the training data.

Evaluation:

Evaluate the model's performance on a separate test set using metrics like accuracy, precision, recall, and F1 score.

Prediction:

Use the trained CNN model to predict the category of new YouTube video titles.

RNN (Recurrent Neural Network):

Introduction:

Recurrent Neural Networks are designed to capture sequential dependencies in data. In the context of YouTube titles, an RNN can be used to model the temporal relationships between words.

Implementation:

Data Preparation:

Features (X): Bag-of-Words representation of YouTube video titles.

Convert the text data into numerical vectors, possibly using embeddings.

Model Architecture:

Design an RNN architecture, possibly using LSTM layers to capture long-range dependencies in the sequential data.

Training:

Split the data into training and testing sets.

Train the RNN model using the training data.

Evaluation:

Evaluate the model's performance on a separate test set using metrics like accuracy, precision, recall, and F1 score.

Prediction: Use the trained RNN model to predict the category of new YouTube video titles.

5. Model Evaluation:

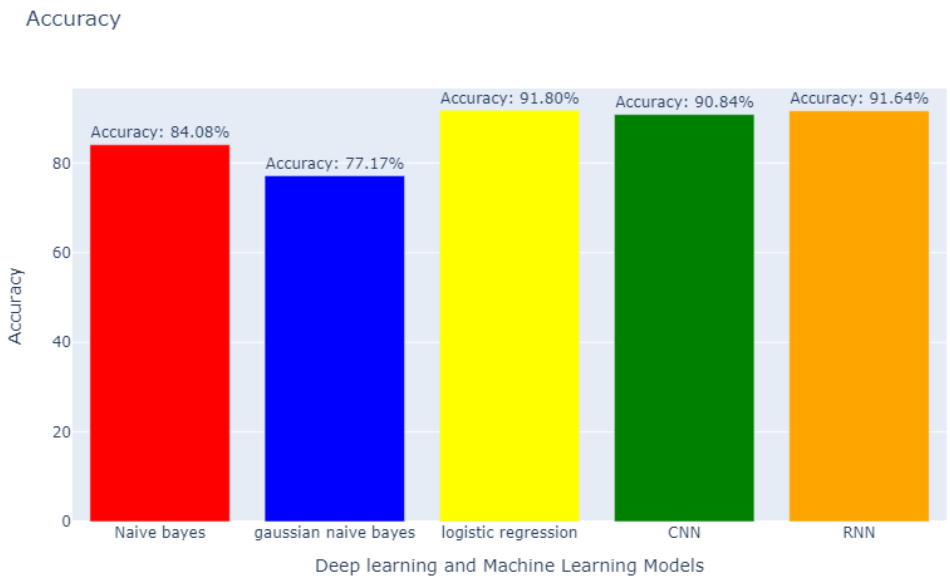
Split the data into training and testing sets for unbiased evaluation.

Employed accuracy, precision, recall, and F1 score metrics to assess model performance.

6. Results and Analysis:

Analyzed and compared the performance of each model.

Considered the strengths and weaknesses of different approaches.



7. Conclusion:

Summarized the findings and selected the best-performing model.

Discussed potential areas for improvement or future work.