CS3 Case Study Rubric – South Park Character Line Classification

Project Topic: Machine Learning for Text Classification
Case Study Audience: 2nd-year UVA Data Science students
Project Source: Group Project – Panic at the Deadline

Why am I doing this? The purpose of this case study is to guide you through building, evaluating, and interpreting a text classification model that predicts which South Park character delivered a given line of dialogue. You will work with a clean and balanced dataset, use TF-IDF feature extraction, train a logistic regression classifier, and analyze model performance and misclassifications. The goal is to help you practice preprocessing, classification modeling, and evaluation. You will follow this rubric to ensure that you meet expectations for each component of the case study.

What am I going to do?

- Final product: Recreate GitHub repository.
- Understand the classification problem.
- Load and explore the dataset, prepare the data for modeling.
- Build model features and train logistic regression classifier.
- Evaluate the model. View outputs.

Rubric Metrics: meets expectations when

Formatting & Repository Structure	• GitHub repository is clean, well-organized, and easy to navigate.
	• Top-level items include:
	  ○ README.md
	  ○ Hook document (PDF)
	  ○ Rubric (PDF)
	  ○ Data folder containing the cleaned dataset or link to source
	  ○ Scripts folder with all code necessary to run the case study
	  ○ Supplemental Materials folder with two required sources
	• README clearly explains how to run code and what the case study is about.
	• File names are clear and consistent.

Data Preparation & Exploration	• Student loads the cleaned dataset successfully.
	• Performs the following text preprocessing steps:
	  ○ Lowercasing
	  ○ stripping whitespace

	○ removing punctuation ○ handling missing values • Correctly explains the purpose of under sampling and oversampling. • Verifies that the final dataset is balanced across all 10 characters. • Includes a short explanation of why balancing matters.
Feature Engineering	• Justifies settings such as: ○ max_features ○ min_df, max_df ○ n-gram range • Includes numerical metadata features (Season, Episode) and scales them. • Correctly combines sparse TF-IDF matrix with numeric features.
Modeling	• Student trains a multinomial logistic regression classifier. • Uses a proper train/validation/test split with stratification. • Provides evidence that the model successfully trains without errors. • Explains what logistic regression is doing in a multi-class setting.
Evaluation & Visualization	• Student generates and includes: ○ classification report (precision, recall, F1) ○ confusion matrix heatmap ○ per-class accuracy plot ○ misclassification analysis (table or plot) • Provides written interpretation of key metrics: ○ Which characters are easiest to predict? ○ Which characters are confused with one another? ○ What patterns appear in the confusion matrix? • Identifies at least one meaningful insight into model performance.