Project Rubric - NLP								
Project Specifications	**Metric for success**	**Developing - 0**		**Accomplished - 1**		**Exemplary - 2**		**Notes**
README.md	Student has a clear readme, highlighting important aspects of the project.	Student does not have a readme, or has one that is just a copy of the notebook.	☐	Student has a readme that is clear and well organized. It outlines their data (sources, anything quirky or hard to understand, etc.), their process, and their recommendations based on results.	☐	Student has a readme with a clear and well organized outlines, conclusion, and recommendation section. Visualizations are present. Language and markdown lend themselves to succinctness.	☑	Great job! Love the directory. Could provide a bit more explanation of the word frequency graphs.
Data Collection	Chosen dataset was relevant to classification.	Dataset was not relevant to classification.	☐	Dataset was simple (such as a Kaggle dataset not supplemented with any other data), but relevant to classification. At least 1000 rows and at least 10 features.	☐	Dataset was originally constructed (webscraping/api calls made) and relevant to classification. At least 1000 rows.	☑	Great job!
Exploratory Data Analysis	Student explores data using visualizations and descriptive statistics.	Inadequate visualizations and descriptive statistics, OR, did not interpret these appropriately.	☐	At least 3 well-constructed visualizations presented along with descriptive statistics; these are interpreted correctly. Student(s) can speak knowledgably about their data.	☐	Student created at least 3 well-constructed visualizations, discussed descriptive statistics, and is able to interpret and compare or link them together to give a cohesive introduction to the data.	☑	Great research and domain knowledge!
Preprocessing/Feature Engineering	Student processed data and created features to increase the predictive capability of the models.	Very little to no preprocessing, no feature engineering	☐	Student preprocessed data and developed new features.	☑	Student preprocessed data and developed domain-motivated features based on their EDA that contributed positively to their model's performance.	☐	Great feature engineering. Think about how you could address class imbalance in the future.
Model Fitting	Student fit at least two models - a baseline and a refined model. Summarize model impact and meaning.	Attempted basic model fitting (or forgot to model fit). Incorrect application. Misinterpreted results.	☐	Correctly fit a baseline and a single model. Correctly interpreted model results. Summarized model meaning & impact.	☐	Compared multiple models, including a discussion of their iterative process and pros/cons of different models. Detailed numerical and visual analysis of models.	☑	Thank you for including a baseline! Think about how your class imbalance could have affected your results!
Model Evaluation	Student evaluated models using appropriate methods and interpreted the results in the context of their data.	Inappropriate methods selected, or results misinterpreted.	☐	Student evaluated their models using at least one metric and interpreted results correctly.	☑	Sudent evaluated their models using more than 1 metric, and provided reasoning as to why they prioritized certain metrics. Student interpreted results correctly.	☐	It would be great for you to talk about the type I vs type II error tradeoff. Again, think about how class imbalance impacts this. Dig a bit deeper into netural tweets and why they were classified this way.
Business Case & Conclusions	Student makes a business recommendation driven by data analysis.	No clear business case or no conclusion is present.	☐	Appropriately challenging business case with conclusion present and includes a business recommendation motivated by data analysis.	☐	Appropriately challenging business case with conclusion present, includes a data-driven business recommendation, and future steps for further analysis/recommendations.	☑	Really unique! Great use of your domain knowledge and insightful future steps
Quality of Code, Github	Student creates a notebook (or several) that is well documented, clean, easy to read and understand, and has a neat, organized GitHub repo.	Code is not in Github, or repo is used improperly (only one or two commits to add completed notebook). Code may not run, or is hard to read, or is not commented.	☐	Code is on Github with frequent commits (and relevant commit messages). Code is clean, organized, and well commented.	☑	Code is on Github with frequent commits (and relevant commit messages). Code is clean, organized, and well commented. Markdown cells add clarity and outline the process. Functions are utilized where appropriate to increase readability and reduce repetition.	☐	Work on commenting and annotating your code a bit more so that anyone could follow along. If you're collaborating on GitHub, please work on branches and then merge to master!
Presentation	Student delivers a clear, organized, well-thought out presentation that can be understood by a non-technical audience.	Slides are unclear, disorganized. Visuals are not legible to audience. Slides may be too text-heavy. Student goes over time.	☐	Student delivers a presentation that is mostly organized and clear in the allotted time. Overall, presentation is understood by audience.	☐	Student is engaging and presents a well-organized, clear, legible presentation in the allotted time. Visuals add to the presentation and are well explained. The presentation tells a cohesive story.	☑	Very strong presentation!
Score (1-5)	5		0		3		12	