Project Name: Natural Language Processing Project
Project Type: Required partner project

Project Goals:
Can the programming language of a GitHub repository be predicted from the text of the README document? Create a dataset of GitHub readme documents, Explore using NLP techniques, Build a model that predicts the programming language using NLP techniques to extract features and classification algorithms for prediction. Deliverables: Well documented Jupyter Notebook, slide deck, and 5 min presentation

Stage	Tools	Brief Description of Process	Challenge Resolution
Plan	● Visual Studio	● Obtained requirements from curriculum ● Decided on subject for dataset gathering ● With partner determined MVP questions to answer for general audience ● Established workload timeline and split up tasks for each of us to address	● No unusual challenges in this section
Acquire	● Visual Studio ● .py script	● Built function to get urls from 1st search page and then get the readme text and language (if present) and return dataframe. ● Broadened function to get urls from multiple search pages ● Scraped 100 search pages with 10 urls each with filtering for only those with readme and language returned 800 observations	● First time using web scraping to obtain data ● Error getting data from more than 10 pages - used sleep function - could have/should use GitHub API for this
Prepare	● Visual Studio ● .py script ● Jupyter Notebook	● Identified the 4 most common languages and filter data to include only those observations (reduced dataset to approx 400 rows) ● Cleaned data (lowercase, convert to only	● First time using regex

		ascii characters and recode to utf-8), use regex to remove anything not a-z, 0-9, or a space ● Tokenized using NLTK Toktoktokenizer, removed stopwords, then created column of stemmed vs lemmatized words ● Split dataset into train, validate, test, and returned additional train_explore dataset	
Explore	● Jupyter Notebook ● Visual Studio	● Checked for imbalance in the dataset language distribution ● Found top 5 words for each language type and discovered 3 words were common to all, returned to prepare and added 'file', 'data', and 'environmental' to the stopwords list ● Created visualization of distribution of top 20 words by by language ● Performed hypothesis test to determine if the length of the readme text was significantly different from the overall mean length. It was not for 3 of the languages, only HTML showed significance ● Functions are found in explore.py	● Did not use word cloud with question mark mask in final notebook, but included the visualization in the slide deck
Model	● Jupyter Notebook ● Visual Studio	● Partner extracted features using both Bag of Words (BOW) and TF-IDF (Term Frequency-Inverse Document Frequency). ● Established a baseline of 35% by predicting the most frequent language (Python) for all observations ● Then used multiple classification models to predict the language including Logistic	● Model results accuracy were disappointing and under other class teams. Scheduled office time with instructor to review process to ensure I implemented everything correctly

		Regression, Decision Tree, Random Forest, K Nearest Neighbors (KNN), and both multinomial and complement Naive Bayes ● Functions are found in model.py	
Evaluate	● Jupyter Notebook ● Visual Studio	● Top 3 models based on accuracy were run on the validate data and the best performing, the Logistic Regression using TF-IDF was run on the test set. The final result was an average of 47% accuracy on unseen data. ● Functions are found in model.py	● Partner created the functions to return evaluations
Model Explanation	**How does your algorithm work?**	● *From curriculum* ● Technically a regression algorithm (goal is to find the values for the coefficients that weight each input variable) ● Used for predicting discrete outcomes (binomial and multinomial) ● Because the prediction for the output is transformed using the logistic function, a non-linear function, it is a classification algorithm. ● The output is a value between 0 and 1 that represents the probability of one class over the other. ● Like linear regression, logistic regression works better when you remove attributes that are either unrelated to the output variable or correlated to other attributes.	

Delivery	● Jupyter Notebook ● Canva	● Partner create slide deck using Canva ● I added Summary, Conclusions, and Next Steps to the Final Jupyter Notebook	● No unusual challenges in this section