### Requirements

- Gather and prepare your data using the `requests` library.

- \*\*Create and compare at least two models\*\*. These can be any classifier of your choosing: logistic regression, Naive Bayes, KNN, SVM, Random Forest Classifier, etc.

- \*\*Bonus\*\*: use a Naive Bayes classifier

- A Jupyter Notebook with your analysis for a peer audience of data scientists.

- An executive summary of your results.

- A short presentation outlining your process and findings for a semi-technical audience.

\*\*Pro Tip:\*\* You can find a good example executive summary [here](https://www.proposify.biz/blog/executive-summary).

---

### Necessary Deliverables / Submission

- Code and executive summary must be in a clearly commented Jupyter Notebook.

- You must submit your slide deck.

- Materials must be submitted by \*\*11:59 PM PST on Friday, January 8th 2021\*\*.

- Presentation must be ready by \*\*09:00 AM PST on Friday, January 8th 2021\*\*.

---

For Project 3 the evaluation categories are as follows:<br>

\*\*The Data Science Process\*\*

- Problem Statement

- Data Collection

- Data Cleaning & EDA

- Preprocessing & Modeling

- Evaluation and Conceptual Understanding

- Conclusion and Recommendations

\*\*Organization and Professionalism\*\*

- Organization

- Visualizations

- Python Syntax and Control Flow

- Presentation

### The Data Science Process

\*\*Problem Statement\*\*

- Is it clear what the goal of the project is?

- What type of model will be developed?

- How will success be evaluated?

- Is the scope of the project appropriate?

- Is it clear who cares about this or why this is important to investigate?

- Does the student consider the audience and the primary and secondary stakeholders?

\*\*Data Collection\*\*

- Was enough data gathered to generate a significant result?

- Was data collected that was useful and relevant to the project?

- Was data collection and storage optimized through custom functions, pipelines, and/or automation?

- Was thought given to the server receiving the requests such as considering number of requests per second?

\*\*Data Cleaning and EDA\*\*

- Are missing values imputed/handled appropriately?

\* missing values were not much of an issue in this data set and were handled swiftly. In the initial cleaning phase Null Values were checked using MissingNo’s library. I created a bar chart highlighting null values. Fortunately, after handling duplicates and other unorthodox values in previous steps only 8 null values existed in the Homebrewing dataset and only 1 null value existed in the Winemaking dataset. I dropped all 9 rows.

- Are distributions examined and described?

- Are outliers identified and addressed?

- Are appropriate summary statistics provided?

- Are steps taken during data cleaning and EDA framed appropriately?

- Does the student address whether or not they are likely to be able to answer their problem statement with the provided data given what they've discovered during EDA?

\*\*Preprocessing and Modeling\*\*

**- Is text data successfully converted to a matrix representation?**

Text Data is converted to matrix representation using Count Vectorizer and Tf-idf. In Tf-idf,

‘Smooth\_idf’ hyperparameter is set to the default of ‘True’ to prevent zero division.

**- Are methods such as stop words, stemming, and lemmatization explored?**

\* Stop words are were initially removed prior to EDA by using CountVectorizers pre-made ‘english’ list of stop words combined with NLTK’s pre-made list of ‘english’ stop words. After performing EDA I found more words I wanted to pull out of the data set based on frequency counts so I appended the previous stop word list with my own additions.

**- Does the student properly split and/or sample the data for validation/training purposes?**

\* A training a testing set was created prior to every model using train\_test\_split function in SKLearn.

**- Does the student test and evaluate a variety of models to identify a production algorithm (\*\*AT MINIMUM:\*\* two classification models, \*\*BONUS:\*\* try a Naive Bayes)?**

**- Does the student defend their choice of production model relevant to the data at hand and the problem?**

**- Does the student explain how the model works and evaluate its performance successes/downfalls?**

\*\*Evaluation and Conceptual Understanding\*\*

- Does the student accurately identify and explain the baseline score?

- Does the student select and use metrics relevant to the problem objective?

- Does the student interpret the results of their model for purposes of inference?

- Is domain knowledge demonstrated when interpreting results?

- Does the student provide appropriate interpretation with regards to descriptive and inferential statistics?

\*\*Conclusion and Recommendations\*\*

- Does the student provide appropriate context to connect individual steps back to the overall project?

- Is it clear how the final recommendations were reached?

- Are the conclusions/recommendations clearly stated?

- Does the conclusion answer the original problem statement?

- Does the student address how findings of this research can be applied for the benefit of stakeholders?

- Are future steps to move the project forward identified?

### Organization and Professionalism

\*\*Project Organization\*\*

- Are modules imported correctly (using appropriate aliases)?

- Are data imported/saved using relative paths?

- Does the README provide a good executive summary of the project?

- Is markdown formatting used appropriately to structure notebooks?

- Are there an appropriate amount of comments to support the code?

- Are files & directories organized correctly?

- Are there unnecessary files included?

- Do files and directories have well-structured, appropriate, consistent names?

\*\*Visualizations\*\*

- Are sufficient visualizations provided?

- Do plots accurately demonstrate valid relationships?

- Are plots labeled properly?

- Are plots interpreted appropriately?

- Are plots formatted and scaled appropriately for inclusion in a notebook-based technical report?

\*\*Python Syntax and Control Flow\*\*

- Is care taken to write human readable code?

- Is the code syntactically correct (no runtime errors)?

- Does the code generate desired results (logically correct)?

- Does the code follows general best practices and style guidelines?

- Are Pandas functions used appropriately?

- Are `sklearn` and `NLTK` methods used appropriately?

\*\*Presentation\*\*

- Is the problem statement clearly presented?

- Does a strong narrative run through the presentation building toward a final conclusion?

- Are the conclusions/recommendations clearly stated?

- Is the level of technicality appropriate for the intended audience?

- Is the student substantially over or under time?

- Does the student appropriately pace their presentation?

- Does the student deliver their message with clarity and volume?

- Are appropriate visualizations generated for the intended audience?

- Are visualizations necessary and useful for supporting conclusions/explaining findings?

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Part 1 of the project focuses on \*\*Data wrangling/gathering/acquisition\*\*. This is a very important skill as not all the data you will need will be in clean CSVs or a single table in SQL. There is a good chance that wherever you land you will have to gather some data from some unstructured/semi-structured sources; when possible, requesting information from an API, but often scraping it because they don't have an API (or it's terribly documented).

Part 2 of the project focuses on \*\*Natural Language Processing\*\* and converting standard text data (like Titles and Comments) into a format that allows us to analyze it and use it in modeling.

Part 3 of the project focuses on \*\*Classification Modeling\*\*. Given that project 2 was a regression focused problem, we needed to give you a classification focused problem to practice the various models, means of assessment and preprocessing associated with classification.