**CSIS 3290-001 Term Project – Kunal Ajaykumar Jeshang (300328339)**

GitHub Repository = <https://github.com/kjeshang/NespressoMetropolisTrainingApp>

Direct Link to Web Application = <https://nespresso-training-app.onrender.com/>

**Introduction and discovery**

Nespresso is an operating unit under the Nestle family of companies. It was founded in 1986 in Lausanne, Switzerland, and has since expanded globally; especially in recent years within the modern-developed world. It is a luxury retail brand that specializes in selling single-use coffee capsules and automated capsule-based coffee machines (*Nespresso* 2022). I have been working as a Coffee Specialist at the Nespresso Metrotown boutique since June 2021. It is my duty to educate customers regarding our product line-up and facilitate transactions with customers regarding coffee and machine purchases. The essence of my job is to frequently make recommendations to customers regarding both current and new coffee flavours. Majority of the time it is the customer that asks the coffee specialist for recommendation based on a flavour that they are familiar with or based on other taste preferences. Thus, the coffee specialist must make a recommendation based on their understanding of the coffee menu and their personal experience tasting the coffee. This leads to a certain degree of variability in the quality of the recommendation as the customer may or may not enjoy the recommended coffee. Therefore, this project was completed to explore possibility of standardizing the recommendation process using machine learning and natural language processing (NLP) using textual data. In addition, this project also attempts to prototype a recommendation engine for the coffee flavours but in the guise of an educational training platform, as a Plotly Dash web application, for both new & existing coffee specialists.

**Data Preparation**

The dataset used for this project was created manually using the Nespresso Canada, UK, and Australia websites, and it consists of general and taste related information regarding the coffee flavours. The dataset is initially in the form of an Excel workbook with two sheets, and each sheet consists of coffee data for the respective Nespresso machine lines; Vertuo and Original. Each machine line has its own respective type of capsules. For some context, a Vertuo coffee capsule is only compatible with an Vertuo line machine, and an Original coffee capsule is only compatible with an Original line machine. The Original line was the initially the only line of machine sold by Nespresso, and it brews the typical European serving sizes; Espresso (40ml) and Lungo (110ml). The Vertuo line was later introduced in 2013 to accommodate the North American consumer whereby larger serving sizes are preferred; this machine line brews Espresso (40ml), Double Espresso (80ml), Gran Lungo (150ml), and Coffee (230ml). After the ‘raw’ dataset was created, I imported it into the Jupyter notebook and concatenated the data from the Vertuo and Original line sheets together into a single Pandas dataframe.

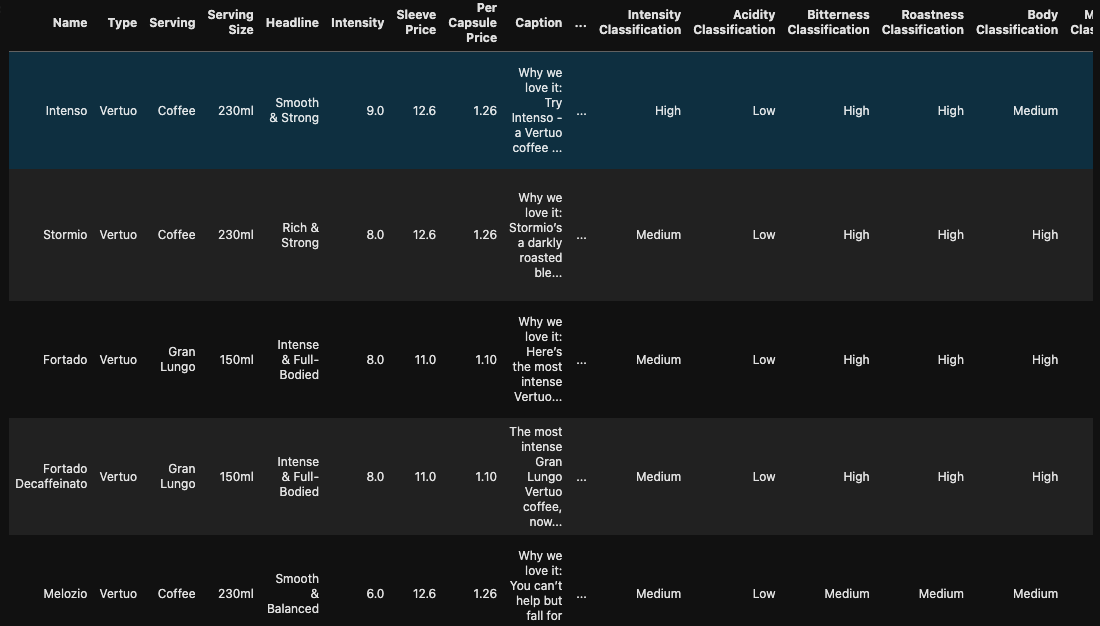
The dataset retrieved from the website is raw as there are certain bits of information (i.e., features) that are not provided for some coffees due to internal privacy reasons or discrepancy. In the context of a data analyst, this yields to null values in the dataset. Features such as intensity and taste profile levels are not provided at all or partially by the Nespresso website. To accommodate for this, I had to apply my own judgement as well as reach out for expertise from the Nespresso Metrotown team leaders for assistance on what values would be logical to include. As my project dealt with textual data for the main purpose of creating a recommendation engine, significant numerical features, such as intensity and taste profile levels, are used to create new classification features via binning ranges. For example, a coffee with an intensity of “2” would have a roast type of “blonde”, whereas a coffee with an intensity of “9” would have a roast type of “dark”.

Now that the data is cleaned, the most significant textual features will undergo pre-processing in preparation for NLP whereby the output would be a new feature, called “Textual Info”, that will be used to perform further data analysis. The following are the features that are used for NLP pre-processing: Type, Serving, Serving Size, Headline, Caption, Taste, Best Served As, Notes, Category, Roast Type, Intensity Classification, Acidity Classification, Bitterness Classification, Roastness Classification, Body Classification, Milky Taste Classification, Bitterness with Milk Classification, Roastiness with Milk Classification, and Creamy Texture Classification. The aforementioned features are combined together into a single variable and lower-cased in the process. Then tokenization takes place where each word is an element in a list. After that, lemmatization is performed so that extended words are reduced to their base words. Certain words are then removed from consideration if they are pronouns or adverbs using Part-of-Speech (POS) tagging. The remaining words are combined together and output as the new “Textual Info” feature.

***Summary Statistics of Numerical Features***

******

***Brief Peek at the Data*** (Note – Not all 41 columns are visible)



***Summary of Features***

|  |  |  |
| --- | --- | --- |
| *Feature* | *Explanation* | *Data Type* |
| ID | Abbreviation of the machine type and integer number to act as a unique identifier to serve as a potential creation of a database in the future. | object |
| Name | Name of the coffee. | object |
| Type | The machine where the coffee flavour capsule is compatible with. | object |
| Serving | The type of coffee drink (i.e., espresso, 'full-cup' coffee, etc). | object |
| Serving Size | The size of coffee drink in milliliters. | object |
| Headline | The introductory phrase that distinguishes the coffee. | object |
| Intensity | The primary indicator of strength of coffee strength. | float64 |
| Sleeve Price | The price of the coffee, which comes in packages of 10 capsules (i.e., cup of coffee) of a respective flavour, in Canadian Dollars. | float64 |
| Per Capsule Price | The price an individual coffee capsule in Canadian Dollars; note that the coffees are sold in packs of ten capsules and NOT on a per-capsule basis. | float64 |
| Caption | A brief description about the coffee and why Nespresso (& it's customers) enjoy the flavour of coffee. | object |
| Taste | A detailed description explaining the coffee's taste profile, coffee bean origin, and other key bits of information. | object |
| Best Served As | Recommended coffee drink and serving size for the respective coffee flavour. | object |
| Notes | Aromatic profile and flavour of the coffee. | object |
| Acidity | Numerical value describing the coffee's taste profile in terms of acidity; range = 1 to 5. | float64 |
| Bitterness | Numerical value describing the coffee's taste profile in terms of bitterness; range = 1 to 5. | float64 |
| Roastness | Numerical value describing the coffee's taste profile in terms of roastness; range = 1 to 5. | float64 |
| Body | Numerical value describing the coffee's taste profile in terms of body; range = 1 to 5. | float64 |
| Milky Taste | Numerical value describing the coffee's taste profile in terms of milky taste; range = 1 to 5. | float64 |
| Bitterness with Milk | Numerical value describing the coffee's taste profile in terms of bitterness with milk; range = 1 to 5. | float64 |
| Roastiness with Milk | Numerical value describing the coffee's taste profile in terms of roastiness with milk; range = 1 to 5. | float64 |
| Creamy Texture | Numerical value describing the coffee's taste profile in terms of creamy texture; range = 1 to 5. | float64 |
| Ingredients & Allergens |  | object |
| Number of Capsules per Sleeve | Number of capsules per pack of coffee (i.e., sleeve). | int64 |
| Net Weight per Total Number of Capsules | Total weight of capsules in coffee sleeve in grams. | object |
| Capsule Image Link | Image of coffee capsule. | object |
| Capsule & Sleeve Image Link | Image of coffee capsule and sleeve. | object |
| Decaf Coffee? | Whether the coffee is caffeinated or decaffeinated. | object |
| Category | Menu category of the coffee (i.e., Inspirazione Italiana, Signature Coffee, Espresso, etc). | object |
| Other Information | Additional information on whether the coffee's intensity was estimated, as well as other noteworthy information. | object |
| Status | Whether the coffee is a past or current fixture of the Nespresso menu. | object |
| Roast Type | Classification of coffee roast; classes = blonde, medium, dark. | object |
| Intensity Classification | Classification of intensity; classes = low, medium, high. | object |
| Acidity Classification | Classification of acidity taste profile; classes = low, medium, high. | object |
| Bitterness Classification | Classification of bitterness taste profile; classes = low, medium, high. | object |
| Roastness Classification | Classification of roastness taste profile; classes = low, medium, high. | object |
| Body Classification | Classification of body taste profile; classes = low, medium, high. | object |
| Milky Taste Classification | Classification of milky taste profile; classes = low, medium, high. | object |
| Bitterness with Milk Classification | Classification of bitterness with milk taste profile; classes = low, medium, high. | object |
| Roastiness with Milk Classification | Classification of roastiness with milk taste profile; classes = low, medium, high. | object |
| Textual Info | Pre-processed & combined textual features. | object |

After the NLP pre-processing portion of the analysis is completed, the coffee data is prepared/transformed. As aforementioned, the coffee data in this form is used for further data analysis & modelling, in addition, serves as the backend to the web application. For further explanation regarding data preparation, please refer to the “1\_DataPreparation” and “3\_NLPPreProcessing” Jupyter Notebooks. To better understand the coffee dataset in greater detail, please refer to “2\_DataExploration” Jupyter Notebook.

**Model Planning & Implementation**

The first part of the data analysis phase is defining the selected coffee. For the purpose of familiarity, the Signature Coffee “Intenso” is selected as it is the coffee that is closest to a good old-fashioned roasted full cup of coffee. This is the coffee that would be utilized perform the following feature engineering techniques:

* Term Frequency Inverse Frequency (TF-IDF)
* Bag of Words (BoW)
* Average Word2Vec
* Average Word2Vec x TF-IDF

The Word2Vec related techniques use a smaller pre-trained model (glove-wiki-gigaword-50) that 65MB in size and contains 400,000 vectors (RaRe-Technologies). The first two feature engineering techniques are performed throughout the entirety of the analysis. In regards to extracting important features and performing validation by classification using the “Roast Type” as the target feature, only TF-IDF and BoW is utilized. Based on my research, there is no conclusive way to validate a content-based recommendation engine, in turn, I am utilizing classification to provide some indication of accuracy. The Roast Type is used as the target feature because it is the predominant determinant of a customer opting to purchase one coffee from another. A pipeline is applied to TF-IDF/BoW vectorizer and Multinomial Naive Bayes. In the pipeline, the coffee data is split into training (80%) & test (20%) datasets; the random state is 42. The aforementioned vectorizers are typically used in NLP and is not expensive on computing resources. Multinomial Naive Bayes is considered a strong tool for analyzing text and classification (*Multinomial naive Bayes explained: Function, Advantages &amp; disadvantages, applications in 2023* 2022). Grid Search Cross-Validation is utilized after implementing the pipeline for to fine tune the parameter of alpha for Multinomial Naive Bayes. In regards to retrieving recommended coffees, all of the above techniques are utilized with “Cosine Similarity” as the constant measure of distance. In the latter parts of the analysis, an experiment is conducted to retrieve recommendations using a combination of the above feature engineering techniques along with the various pair-wise measures of distance: Cosine Similarity, Linear Kernel, Euclidean Distance, and Manhattan Distance. The first two measures of distance are classical in the context of NLP, but for the purpose of exploration, I was curious to see what recommendations and similarity scores would I receive using the latter two measures of distance. Below is a table with additional fixed parameters used in majority of the analysis phase.

**Results Interpretation and Implications**

The top ten recommendations were retrieved using each technique but with a constant measure of distance. Below is a breakdown of the 1st and 10th recommended coffee given that selected coffee was Intenso, along with Cosine Similarity scores.

***Alternate Recommendations for Intenso using all Feature Engineering Techniques and Cosine Similarity***

|  |  |  |
| --- | --- | --- |
| *Technique* | *1st Recommendation & Similarity Score* | *10th Recommendation & Similarity Score* |
| TF-IDF | Diavolitto (0.7289) | Melozio (0.5252) |
| BoW | Stormio (0.7790) | Inspirazione Venezia (0.6827) |
| Average Word2Vec | Stormio (0.9857) | Buenos Aires Lungo (0.9695) |
| Average Word2Vec x TF-IDF | Stormio (0.9760) | Miami Espresso (0.9532) |

The first recommendations for Intenso using all of the techniques are logical. TF-IDF yielded Diavolitto, which is a Vertuo Espresso (40ml) that has an intensity level of 11; note – 11 is the maximum intensity for Vertuo coffees. The other techniques yielded Stormio, which is a Vertuo Coffee (230ml) that has an intensity level of 8. The tenth recommendations for Intenso using all techniques are all unique. The distinction between techniques utilizing Average Word2Vec have very high similarity scores for both the first and tenth recommendations. It is does not seem logical as despite there being a total 70 coffees on the standard Nespresso menu, the similarity scores are all above 0.9 for the first & tenth recommended coffees that utilize Average Word2Vec; there is significant variability even though the recommendation itself is valid. In the case of TF-IDF and BoW, the recommendations are logical, and similarity scores display distinguishable variability so one can differentiate between a closely similar to distantly similar recommended coffee. Therefore, based on the results and interpretation, TF-IDF and BoW are the most ideal to consider for model validation and prediction, as well as use in the Plotly Dash web application.

As it has been deemed that TF-IDF and BoW were the most logical techniques, so the most important features are extracted. Specifically, the features that have a frequency score greater than 0 using the aforementioned techniques for the Intenso coffee.

***Important Features using TF-IDF and BoW***

|  |  |
| --- | --- |
| *TF-IDF Important Features* | *BoW Important Features* |
|  |  |

There is a clear difference between the important features between TF-IDF and BoW. There is an incremental distribution between the lesser important to the more important feature for TF-IDF. In the BoW feature chart, the “word” coffee is the most important feature compared to the others by a large margin. This is likely because that Intenso is a 230ml full cup of coffee. Another distinction is that the word “coffee” is the fifth important feature under TF-IDF but the most important for BoW. Interestingly, the word “coffee” refers to the serving of the coffee and “230ml” refers to the serving size of the respective serving. The quality of the feature extraction can be questioned because despite performing lemmatization in the NLP pre-processing step of the project, the word “arabicas” was not reduced to “arabica”. In turn, both words exist are a part of the important features for both TF-IDF and BoW. The general difference between both of these techniques is that TF-IDF considers the both the frequency and importance of the words, whereas BoW only considers the frequency of the words. The similarity is that a good proportion of the important features exist for both of the techniques, albeit with different frequency score. An interesting observation is that the word “dark” is the second most important feature for both TF-IDF and BoW.

In the results output below, the classification pipeline scores are retrieved after vectorization (TF-IDF/BoW) and applying multinomial naive bayes. No alpha is specified so the default alpha is 1 (*Sklearn.naive\_bayes.multinomialnb*). Hyperparameter tuning is performing using GridSearchCV to find out the most optimal value of alpha that can be used for multinomial naive bayes based on the feature engineering technique utilized (Team, 2022).

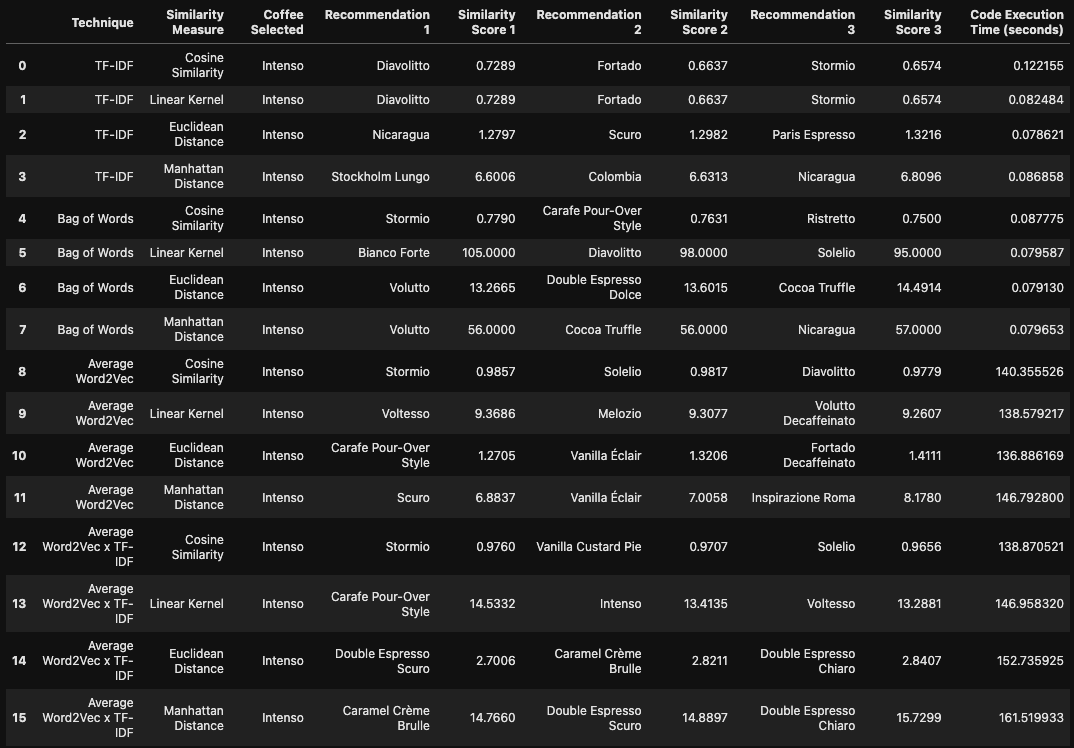
***Validation by Classification by Scoring Accuracy & Prediction of Roast Type***

|  |  |
| --- | --- |
| *TF-IDF* | *BoW* |
|  |  |

The classification results above indicate that prior to hyperparameter tuning, the accuracy of the classification model that utilizes BoW is superior, although the opposite is true for TF-IDF. The prediction models are overall not as optimal for either technique. This could be due to the fact that there are only 70 coffees on the standard Nespresso menu, in turn, the 70 rows in the dataset. From the confusion matrix, it can be deemed that the TF-IDF used with multinomial naive bayes is incapable successfully classifying Blonde roast coffee, however the prediction capability is somewhat agreeable when classifying Medium and Dark roast coffees. The BoW confusion matrix displays overall higher accuracy in classifying the roast of coffees.

Below are the results of an experiment conducted that compares all feature engineering techniques with various measures of distance. The top three recommendations are shown, along with similarity score and code execution time.

***Compare Recommendations & Code Execution Time of various Techniques and Measures of Distance***



The results above indicate that Euclidean and Manhattan distances have variable degrees of success when it comes to yielding the mode accurate recommendations. There are few cases where the feature engineering technique is “Average Word2Vec” and “Average Word2Vec x TF-IDF” such that strong dark roast coffees are top recommended irregardless of size, although the subsequent recommendations are not entirely accurate at least when considering overall human taste and the roast. However, the aforementioned feature engineering techniques are very resource intensive as the code execution time regardless of the measure of distance is well over 120 seconds. Thus the cost of utilizing these techniques does not payoff in terms of accuracy. The classical techniques such as TF-IDF and BoW tend to yield more logical recommendations and are not resource intensive as code execution time is less than a second long. The Cosine Similarity and Linear Kernel are the best performing measures of distance across the board, and that definitely applies for TF-IDF and BoW. The Cosine Similarity distance measure was the most ideal as Linear Kernel yielded values greater than 1, which is not normalized, in the case of BoW.

For more detail regarding the data analysis portion of the project, please refer to “4\_DataAnalysis” Jupyter Notebook.

**Out-of-Sample Predictions**

To perform out-of-sample predictions, a new dataset was created using the Nespresso Canada, USA, UK, and Australia websites. The contents of this dataset consist of general and taste related information about seasonal and limited edition coffees. The data preparation (cleaning & transformation) process is similar to that of the main dataset, however there was a lot of standardization in terms of accommodating for null values as the Nespresso website provides less information regarding the seasonal and limited edition coffees on its website.

The out-of-sample selected coffee is Peppermint Pinwheel. It is seasonal Christmas special Vertuo coffee (230ml) and is a half-caffeinated roast. Using all feature engineering techniques and Cosine Similarity distance measure, recommendations are retrieved from the standard Nespresso menu. Below is a table with a breakdown of the top recommendations and similarity scores for Peppermint Pinwheel, along with brief comments on whether the recommendation is logical.

***Recommendations for Out-of-Sample Coffee using various Techniques and Cosine Similarity***

|  |  |  |
| --- | --- | --- |
| *Technique* | *Top Recommendation & Similarity Score* | *Comment* |
| TF-IDF | Half Caffeinato (0.5949) | Logical |
| BoW | Half Caffeinato (0.7163) | Logical |
| Average Word2Vec | Miami Espresso (0.9613) | Not Logical |
| Average Word2Vec x TF-IDF | Intenso (0.9567) | Not Logical |

TF-IDF and BoW both yielded Half Caffeinato which is also a half-caffeinated medium-to-light roast. The similarity score is reasonable but not very close to 1 which makes sense as the flavour profile for Half Caffeinato differs from Peppermint Pinwheel. The remaining techniques yielded both dark roast coffees that are full caffeinated roasts with similarity scores above 0.90 which does not make sense.

Below is a table showing output of predictions as well as accuracy scores whilst performing validation by means of classification, but making prediction using the out-of-sample data.

***Validation by Classification by Scoring Accuracy & Prediction of Roast Type using Out-of-Sample Data***

|  |  |
| --- | --- |
| *TF-IDF* | *BoW* |
| Pipe Score = 1.0  Grid Best Parameter = {'mulNB\_\_alpha': 0.1}  Grid Best Score = 0.7321428571428571  Grid Score = 1.0 | Pipe Score = 0.8  Grid Best Parameter = {'mulNB\_\_alpha': 0.1}  Grid Best Score = 0.7321428571428571  Grid Score = 0.8 |
|  |  |

Whilst performing validation by classification with the target feature being Roast Type, it was deemed that using TF-IDF for the out-of-sample pipeline yielded to an accuracy score of 100%. This differs from when assessing the predictive accuracy using the test set. When using BoW, the accuracy score is 80%.

**Concluding Remarks**

The way this project has evolved over the course of the semester emulates the exploration of formulating a data driven solution through the act of performing data preparation, data exploration, pre-processing, data analysis, and out-of-sample prediction. It was discovered that TF-IDF and BoW was less resource intensive and yielded the most logical recommendations with a distinguishable variability in similarity scores. In a production environment the aforementioned discovery would be imperative to construct a concise & accurate recommendation engine for the Nespresso training platform web application. That being said, the Jupyter Notebook portions of this project were helpful to prototype some of the functions and visualizations that would become part of the interactive Plotly Dash web application. To a certain degree, the Jupyter Notebooks are fixed and static, whereas the Plotly Dash web application is dynamic as data filtration, parameter adjustment, and target feature changes can be made to suit the needs of whoever is using it. Therefore, a the final culmination of this project is not just the Jupyter Notebook analysis and Report, but also a prototype machine learning application which in some cases could be the final end-goal for a data driven organization in a production environment.

***Machine Learning Application***

The additional component of this project is the Plotly Dash web application that is meant to prototype a prospective Machine Learning Application that is in the guise of a training platform. It can be considered an extension of the work done in the Jupyter Notebooks. The web application was developed using Python 3.10.1, although compatibility with Conda Version of Python should not be a problem if all dependency requirements are met.

There are two versions. Before running any of the versions, please make sure that all dependencies are installed. Kindly refer to the "Additional Packages to Install" and "Requirements" text files.

Local Version: This was the version used in the development stage of the project. To run it locally, open “NespressoMetropolisTrainingApp” project directory in code editor/IDE, and then simply run the "index.py" file in the main directory.

Live Version: This is the live version of the application. It can be accessed directly online via a web browser by typing "https://nespresso-training-app.onrender.com/" in the URL search bar. Please note that the online web application runs a bit slow as it is deployed on Render (free tier), but functions reasonably. The main difference is that the structure of the application was altered to accommodate deployment to the Render platform, and GridSearchCV feature was removed from due to code execution timing out under the Render platform's free tier that only provides 512MB of RAM. If you wish to run this version locally, open “nespresso-training-app” project directory in code editor/IDE, and then run the "app.py" file in the main directory.

Below is a breakdown of the web application’s pages.

* Home: Landing page with the title of the project and my name.
* About: Brief description about the project.
* Explore: Exploration of the coffee dataset but in a fun-interactive & storytelling manner.
* NLP: Performs machine learning models to output recommendations, feature results, and validation based on selected coffee and specified parameters; additionally, informative coffee information is provided upon selection.

**References**

Note that Jupyter Notebook and Plotly Dash web application related references are included in an alternate document in the Report folder called “Programming References”.

Multinomial naive Bayes explained: Function, Advantages &amp; disadvantages, applications in 2023. upGrad blog. (2022, November 22). Retrieved December 2, 2022, from [https://www.upgrad.com/blog/multinomial-naive-bayes-explained/#:~:text=The%20Multinomial%20Naive%20Bayes%20algorithm%20is%20a%20Bayesian%20learning%20approach,tag%20with%20the%20greatest%20chance](https://www.upgrad.com/blog/multinomial-naive-bayes-explained/#:~:text=The Multinomial Naive Bayes algorithm is a Bayesian learning approach,tag with the greatest chance).

RaRe-Technologies. (n.d.). Rare-technologies/gensim-DATA: Data Repository for pretrained NLP models and NLP corpora. GitHub. Retrieved December 2, 2022, from <https://github.com/RaRe-Technologies/gensim-data>

Sklearn.naive\_bayes.multinomialnb. scikit. (n.d.). Retrieved December 2, 2022, from <https://scikit-learn.org/stable/modules/generated/sklearn.naive_bayes.MultinomialNB.html>

Team, G. L. (2022, June 13). Hyperparameter tuning with GRIDSEARCHCV. Great Learning Blog: Free Resources what Matters to shape your Career! Retrieved December 2, 2022, from <https://www.mygreatlearning.com/blog/gridsearchcv/>

Wikimedia Foundation. (2022, November 20). Nespresso. Wikipedia. Retrieved December 2, 2022, from <https://en.wikipedia.org/wiki/Nespresso>