***Name:*** Gayathri Girish Nair | ***Student ID:*** 23340334 | ***Course Code:*** CS7DS3 | ***Stream:*** Data Science

# WIND DATA ANALYSIS

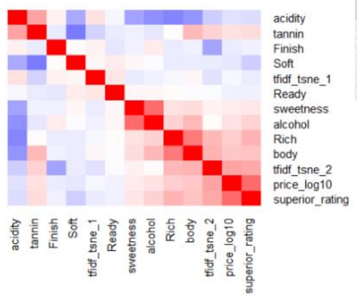
**Note:** Please find codebase related to this project on GitHub here. Also, other than the central figure, many others (around 33 figures were generated and examined in total) are available as links (stored no google drive) in order to fit content within the page limit. Please find the high quality pdf of the central figure here. Also, kindly note that this submission is only 8 pages long. The last page is references.

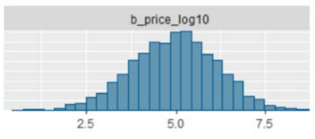
## 1. Central Figure

### Let them drink Wine

### GOOD NEWS! DECEPTIVE WINE PRICES

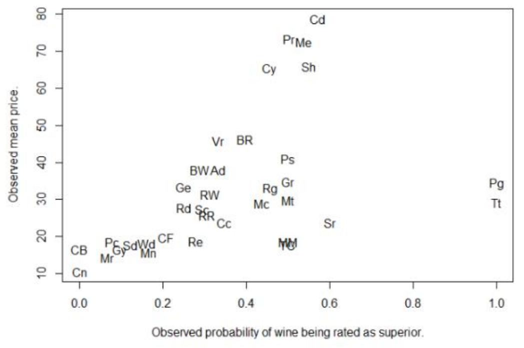
*Analysis of wine reviews that uses Bayesian hierarchical logistic regression to model variety wise effects in addition to population level effects of wine characteristics on likelihood of it being rated as superior.*





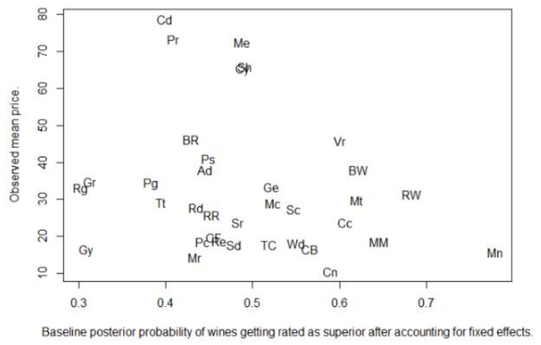
*Prices were most positively correlated with wine ratings and price had greatest effect on logs odds of wine being rated as superior. effect on logs odds of wine being rated as superior.*

**Plot 1:** Unadjusted relationship between price (y axis) and the observed probability of superior rating for each wine variety (x axis)

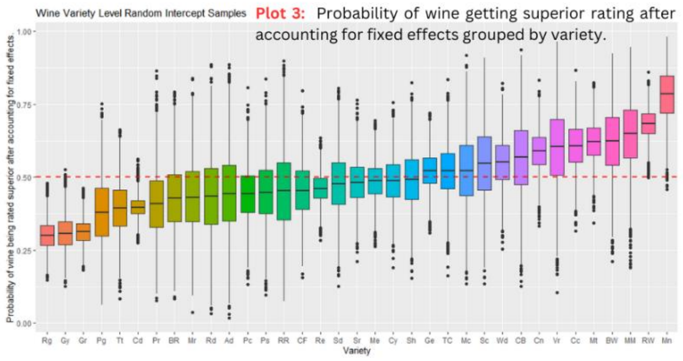


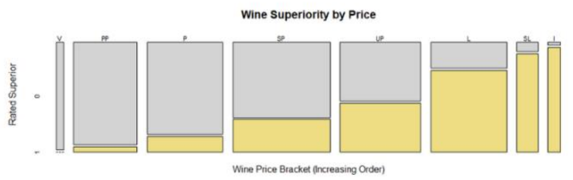
*From plot 1 to plot 2, the probability of high-priced wines like Cd (Champagne Blend) dropped while that of cheaper ones like Mn (Melon) increased. Even varieties like Cn (Cabernet Sauvignon) and CB (Chenin Blanc-Chardonnay) with very low odds of being ranked as superior in plot1 were assigned higher probabilities (55-60%) by the model in plot2 likely because they have characteristics beyond price that are similar to that of superior wines. Thus, price is an important factor influencing observed probabilities. It's possible that some varieties like Cd may be overpriced. Long story short, you don't have to break the bank to have great wine. There are plenty of affordable options. Here, the model recommends Melon (Mn), Malbec-Merlot (MM) and Rhône-style White Blend (RW) wine varieties as great affordable options that may even be better at times, than high-priced alternatives.*

**Plot 2:** Posterior probability of wines getting rated as superior per variety after accounting for fixed effects v/s observed prices.

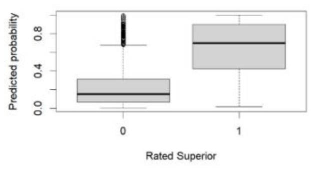


**Plot 3:** Probability of wine getting superior rating after accounting for fixed effects grouped by variety.

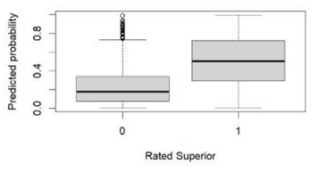




**TRAIN SET:** Accuracy = 0.8128, F1 Score = 0.7437



**TEST SET:** Accuracy = 0.7325, F1 Score = 0.5664



### BACKGROUND & METHOD

- *Dataset of 2500 wine reviews, assigned ratings, prices, and presence indicators for flavor/appearance/mouthfeel characteristics.*
- *NLP methods employed to get scores for key wine judging characteristics (acidity, sweetness, tannin, alcohol, body). Topic modelling done to explore. Feature engineering added new features like price bracket, alcohol, tanning, body, acidity, sweetness, tfidf_tsne_1. Price log scaled to reduce skewness. Min max normalization before model fitting.*
- *Likelihood of superiority of wine ratings modeled using the Bayesian Hierarchical Logistic Regression approach. Grouping variable = variety. Predictors = price_log10 + tannin + alcohol + Rich + price_log10 * body + price_log10 * tfidf_tsne_1_norm. Response variable = superior_rating. Data distribution: Binomial (success = wine superior), Priors: Student t distribution for random effects term and Normal distribution for population parameters.*
- *Model was tested on new test data found online. Some overfitting was observed.*

### CONCLUSION

- *Of all the fixed effects in the model,  has the highest average effect (comparatively highest coefficient of 4.96).*
- *Derived features had larger effect compared to the single indicators like "Rich" reinforcing value of having derived key wine judgement criteria incorporating more than one indicator from text.*
- *The baseline tendency is for wines to not get rated as being "superior" (population fixed effect intercept estimate = -10.55).*
- *Barring interaction terms, fair evidence against zero effect for all other predictors (posteriors away from 0).*
- *Some wine varieties get ranked high more often than others (plot 3) but large variance + outliers suggest high subjectivity.*
- *Removing effect of price revealed that some cheaper wines may be just as good as more expensive ones.*

## 2. Objective Statement

The main objective of this project was to examine effects of various indicators such as wine characteristics on the probability that the wine gets rated points $\geq 90/100$ and thus superior. It was also of interest to examine if some varieties of wines are more or less likely to be rated highly. Prices of wines were also considered to perhaps investigate if it really is worth it to go for expensive wines.

General questions that this project tried to answer were as follows.

- Do points assigned to wine depend on the variety of the wine? By how much?
- What sort of relationship exists between the price of wine and the probability of it being rated high?
- Can a hierarchical logistic regression model capture patters well enough to make fairly (≥70%) accurate predictions?
- How can review text be better leveraged to create more informative input features?

## 3. Introduction

This project investigated a "wine review" dataset to understand patterns, if any, that link wine varieties, their prices, and the no. of points they receive upon being rated (superior = points $\geq 90$ or not) on multiple characteristics (taste, smell, appearance, mouthfeel, etc).

The dataset is fairly big with 2500 data points, but the distribution of those data points among the various wine varieties is highly uneven. For instance, there are only 3 data points for the "Petit Manseng" variety of wine while there are 417 data points corresponding to the "Chardonnay" variety. In such a case wherein multiple groups of interest are underrepresented in the data set; frequentist methods may be unreliable with the model being prone to overfitting. Overfitting here, refers to the model fitting the data corresponding to varieties with highest no. of data points very well but failing to fit underrepresented varieties well enough due to a lack of data points, to ultimately result in high variance between estimates w.r.t inputs belonging to different groups. Thus, adopting the Bayesian framework was deemed a good idea because by incorporating prior information and considering hierarchical structures, it would provide more robust estimates through partial pooling that leverages the idea of borrowing strength across varieties to better estimate parameters.

The decision to use a Hierarchical Logistic Regression model to fit the data was made because such a model would be able to capture the discrete binary response variable $superior\_rating$ in light of a Bernoulli distribution (if wine received superior rating then 1 = success and 0 = failure otherwise) through a linear combination of predictor variables, while also taking into account the grouping structure (wine $varieties$) present in the data such that group parameters are pooled together to share information (partial pooling), thereby allowing for underrepresented groups to draw support from others and also adding a regularization effect which should reduce over-fitting.

Subscripts

Give below is some important notation to bear in mind w.r.t subscripts in order to understand the model definition that follows.

- $k = 1, \dots, K$ refer to groups that are present in the data set and are being considered in the model. Here, because the ***grouping variable*** is wine $variety$ and there are 35 different varieties of wine in the data set, $K = 35$.
- $i = 1, \dots, n$ refer to data points that are present within a group. For example, for the "Chardonnay" variety of wine in the data set, there are 417 data points (wines). Thus, w.r.t the "Chardonnay" group, $n = 417$.
- $p = 1, \dots, P$ refer to predictor variables used to estimate each instance of the response variable. In the model considered here, the 12 ***predictor variables*** are $Rich$, $price$, $alcohol$, $tannin$, the interaction between $price \times tsne(tfidf)\ component\ 1$, and between $price \times body$. Thus, $P = 6$.

The Model

The ***response variable*** $y_{ik}$ (the variable we want to model) is $superior\_rating$. This is a binary variable such that when it is 1, this means that the corresponding wine was assigned a rating of $\geq 90$ points. Consequently, $superior\_rating = 0$ implies that the wine received a rating $< 90$ points.

Since here, the response variable $y_{ik} \in \{0, 1\}$, meaning that outcomes are discrete and binary, logistic regression would be a better choice for a data model than linear regression because the latter predicts continuous values while the former deals specifically with binary outcomes. Moreover, a linear regression model would try to fit a straight line to the data, which likely wouldn't accurately represent the binary nature of the response variable (superior rating or not). In contrast, a logistic regression model uses a sigmoid function to transform the linear combination of predictors into a probability between 0 and 1 which can then be interpreted as the chance of observing a superior rating ($superior\_rating = 1$) for a particular wine given its characteristics $x$.

The data model is assumed to be $y_{ik} \sim Bernoulli(\theta_{ik})$ wherein, a wine's rating being superior ($superior\_rating = 1$) is a success, and it receiving a lower rating $\left(superior_{rating} = 0\right)$ is seen as failure.

The log-odds of success (a wine being deemed superior), is modelled as a linear function of the predictor variables and the grouping term to obtain a logistic regression model as follows wherein $\beta_1, \dots, \beta_p$ are the coefficients associated with each predictor variable that explains the corresponding response variable instance.

$$\text{Log}\left(\frac{\theta_{ik}}{1-\theta_{ik}}\right) = \beta_{0k} + \beta_1 x_{ik1} + \cdots + \beta_p x_{ikp}$$

The Random Effects (RE) intercept $\beta_{0k}$ is special because it represents how, assuming a certain population baseline log odds of wines being rated as superior, this varies from group to group. It is the inclusion of this RE term, that makes this a special kind of hierarchical model called a Random Effects Hierarchical Model. W.r.t this dataset, $\beta_{0k}$ represents difference between the population baseline log-odds of wine being ranked as superior and the log-odds for the same per group (wine variety). [1]

The link function here, is $\phi(x_{ik}) = \log\left(\frac{\theta_{ik}}{1-\theta_{ik}}\right)$. This link function computes $\log(.)$ so that the resulting values are valid probabilities and do not fall outside the [0, 1] range [2]. Thus, given covariates $x_{ik}$ and an associated vector of coefficients $\beta$, $y_{ik}$ can be obtained as $\mathbb{E}[y_{ik}] = \theta_{ik} = \phi^{-1}(x_{ik}) = \frac{1}{1+e^{-\beta^T x_{ik}}}$, meaning that $\mathbb{E}[y_{ik}]$ can be used to predict the value of $y_{ik}$.

By default, the $brms$ package that was used to build the model, assumes that the RE intercept follows a Student T distribution as follows wherein the mean is 0 (to reflect no prior belief about the average value of the intercept across varieties), standard deviation is 2.5 (to imply a weak prior that allows for some variability without asserting a strong influence) and the degrees of freedom is set to 3.

$$\beta_{0k} \sim t(0, 2.5, 3)$$

Generally, the bigger the degrees of freedom, the closer to a normal distribution, the t distribution is [3]. So, with low degrees of freedom here, the tails of the T distribution will be fat, meaning that more extreme values are allowed and thus a weaker prior. The package $brms$ packages has set this as the default to keep the prior weakly informative while also “providing at least some regularization to considerably improve convergence and sampling efficiency” [4]. These conditions are favorable for our use case here as well. Hence, we stick to this default prior definition.

Since population level parameter coefficients $\beta_0, \beta_1, \dots, \beta_p$ are unknown, priors must be chosen for them as well, which, in accordance with prior conjugacy, can be assumed to be a normal distribution. This may be expressed as follows where subscript $j$ refers to each parameter with index value $\in \{0, 1, \dots, p\}$. Due to a lack of domain knowledge regarding wines and the absence of an acquaintance who is a domain expert, here the choice was made to assume a non-informative prior such that mean is $0$ and standard deviation is $10$.

$$\beta_j \sim N(0{,}10)$$

The $brm$ function that comes with the $brms$ R package was used to define the model as follows using the $formula$ syntax. Here, the term $(1|variety)$ indicates that a random hierarchical intercept term $\beta_{0k}$ should be included in the model and that the intercepts should grouped at the wine $variety$ level.

```
fit <- brm(
  superior_rating ~ 1 + price_log10 + tannin + alcohol + Rich + price_log10*body +
                    price_log10*tfidf_tsne_1_norm + (1|variety),
  family = bernoulli(logit), data = wine_reviews, prior = prior(normal(0, 10), class=b)
)
```

*Note: Please find reasons for choice of selected predictor variables in section 5 below.*

## 4. Data Description

Initial data processing was done using Python via a python notebook. All corresponding code is available in the $data\_processing.ipynb$ file which contains cells with run outputs and annotations. Python was chosen instead of R for this phase so as to leverage complementary skills and apply NLP methods. This section will describe briefly, all that was done in each section in this $data\_processing.ipynb$. Python libraries leveraged include: $nltk, spacy, pyLDAvis, numpy, pandas, TextBlob, scikit-learn, re, matplotlib$.

Load & Inspect: This step involved loading the dataset and taking a look at its contents. The raw dataset contained 2500 rows and 16 columns. Each row corresponds to characteristics of one wine and its associated review. Columns include: $points$ (points that the wine was assigned, which judging from the values, seems to be in the 100-point scale [5]), $superior\_rating$ (a binary variable that when 1 implies that the wine was rated $\geq 90$ points, thereby making it superior), $price$ (the price of the wine in dollars), $variety$ (this is a categorical field containing strings that indicate what kind of wine this is), $Crisp, Dry, Finish, Firm, Fresh, Fruit, Full, Rich, Round, Soft, Sweet$, (these are all binary variables that when 1, imply that the corresponding term appeared in the review description) and $description$ (this is free text that is the full review that the wine was given). The data set was examined to check for null values. There were no null values and hence, no missing value handling was required. A main purpose of this file is to process the text available in the $description$ field. Thus, the next course of action was to save all text from this column into a text file called $review\_text.txt$ for a quick skim to get an idea of what wine reviews generally look like.

Cleaning: Next step involved cleaning the descriptions to remove unnecessary, meaningless (from a machine’s perspective) words and special characters. This involves the steps: 1. The description was converted to lowercase. 2. Any character other than numbers, letters and % symbol was dropped. The decision was made to retain numbers and the % symbol as percent of certain aspects present/absent in wine might be a useful indicator of its rating. 2. Next, contractions like “it’ll” was expanded to “it will”. A list of contractions and their expansions was obtained from here for the same. 3. Next, all instances of ’s used to indicate possession and extra spaces were dropped. 4. Then, stop words (words like and, an, a, etc.) that are often meaningless to machines were dropped. The $stopwords$ list that comes with the $nltk$ library was leveraged to do this. 5. The next step involves removing as many non-English words as possible. Once again, the list of English words that comes with $nltk.corpus$ was used to word match here.

Lemmatization: The next phase was lemmatization, which refers to converting words to their root words (e.g., "doing" → "do", "sweetened" → "sweet", etc.) [6]. This is done to preserve word meanings while reducing size of the corpus. The language model $en\_core\_web\_lg$ offered by the $spacy$ python package was used to do this.

Feature Engineering: This step involved generating new useful features and adding it to the dataset. Five feature extraction processes were implemented as follows.

(i) ***Sentiment Detection:*** The $textblob$ python package makes available, a pretrained model that, upon being fed in text, outputs the sentiment of the text in the range $[-1, 1]$ such that $-1$ indicates extremely negative, 0 implies neutral and 1 indicates extremely positive. A new feature that captured the sentiment score as output by this library was added to the dataset as a new column $sentiment$.

(ii) ***Variety Name Encoding:*** A dictionary was manually created that matched all wine variety names to short, two-letter codes (e.g., Champagne Blend → Cd, Bordeaux-style Red Blend → BR, etc.). This is so that the names may be displayed in R with less clutter. Thus, shortened version of each variety was added to the database as a new column called $variety\_c$.

(iii) ***Price Bracket:*** Next, a new column called $price\_bracket$ was added. This column contains categories associated with the expensiveness of each wine. Wines are generally grouped into 9 price brackets (Extreme Value (EV), Value (V), Popular Premium (PP), Premium (P), Super Premium (SP), Ultra Premium (UP), Luxury (L), Super Luxury (SL), Icon (I)) in increasing order of their price from categories. [7] This field was added in hopes of discovering additional interesting patterns.

(iv) ***Wine Characteristics Score:*** Five key characteristics based on which wine is judged, include sweetness (how sweet/not), acidity (how tart/not), tannin (how astringent/bitter), alcohol (how warm/spicy), body (how heavy, mouthfeel, fullness, etc.). [8] It was observed that reviews contained words true and opposite to these characteristics. Thus, it would be possible to split sentences into 3-grams (read 3 words at a time – words 1, 2, 3, then 2, 3, 4, then 3, 4, 5, etc.) and then look for presence of words true to and opposite to each of the 5 aforementioned characteristics and add +1 when a true word was found and a -1 when an opposite word was found, before multiplying the result with the sentiment of the sentence segment and then aggregating it over every segment to obtain the final characteristic score for all five characteristics for each review. Dividing the final score for each review by the no. of word segments traversed results in a well-behaved value between 0 and 1. The original idea involved multiplying by sentiment so that "not sweet" would not result in $sweet = 1$. But, upon trying, it was found that sentiment scores received from TextBlob were erroneous in many cases and hence the "multiply by sentiment" step was dropped. Final scores for each characteristic for each review was added to the database as 5 new columns ($acidity, sweetness, body, tannnin, alcohol$) which would look as shown in figure 1. For each characteristic, a list of around 10 positive (+1) and negative (-1) words in accordance with wine-tasting lingo, was obtained both based on the raw dataset, online articles and using GenAI (ChatGPT 3.5). These words were stemmed (end shortened) wherever possible so as to allow for more matches with variations of words with similar meaning (e.g. acidic was shortened to acid which would allow it to match with acidity, acidic, as well as acid). Given below, is the dictionary that was used.
***Note:*** It was observed that some of the indicator variables in the raw dataset like (round, full, rich, soft, dry, etc.) were in fact, only some of the words associated with at least one of these characteristics. Thus, these characteristic scores also encompass information in many of those indicator variables.

```
character_points_map = {
    "sweetness": {1: ["sweet", "sugar", "dessert", "fruit", "honey", "sacchar", "lush", "ripe"], -1: ["dry", "crisp", "tart",
"austere", "sharp", "astringent", "tangy", "brac", "acid", "sour"]},
    "acidity": {1: ["acidic", "tart", "tang", "zest", "sharp", "crisp", "brac", "bright", "zing", "live"], -1: ["sweet", "mellow",
"smooth", "velvet", "round", "soft", "lush", "creamy", "full", "rich"]},
    "tannin": {1: ["tannin", "astringent", "grip", "structure", "firm", "texture"], -1: ["smooth", "soft", "silk", "velvet", "mellow",
"round"]},
    "alcohol": {1: ["alcohol", "drunk", "spirit", "booze", "bubbl", "liquor", "hooch", "fort", "hot", "warm", "spic", "full"], -1:
["dry", "crisp", "tart", "austere", "sharp", "acid", "bitter", "astringent", "tangy", "tannic"]},
    "body": {1: ["bod", "struct", "full", "weight", "mouthfeel", "texture", "rich", "depth", "deep", "substance", "visc"], -1:
["light", "thin", "water", "delicate", "weak", "insubstantial", "flimsy", "dilute", "lack", "air"]}
}
```

(v) ***TF-IDF:*** Term Frequency, Inverse Document Frequency (TF-IDF) is a way to measure importance of words in a piece of text w.r.t the full collection of documents. For example, in this wine data set, the word "wine" might be frequently occurring in a given review, but this does not make it important here as it is very likely for this word to appear in most other reviews as well. Thus, this word is less informative about the uniqueness of a particular review. TF-IDF overcomes this issue surrounding computing only $TF_{ij} = frequency\ of\ word\ i\ occurring\ in\ document\ j$, by also computing inverse document frequency $IDF_{ij} = \log\left(\frac{total\ no.\ of\ words}{frequency\ of\ word\ i\ occuring\ in\ all\ documents}\right)$ which will result in a final score such that effect of globally common words are suppressed in order to highlight that of common review-specific words. In our case here, document = review [9] [10]. This was implemented using the $TfidfVectorizer$ offered by the $scikit - learn$ package. [11]. The result is a vector of length = corpus length ($= 2796$

here) associated with each review. These high dimensional vectors were reduced to 1, 2 and 3 components using the t-SNE algorithm [12] as available through the $scikit - learn$ package. The $matplotlib$ package was then used to plot these components from a 1D (figure 2), 2D (figure 3) and 3D (figure 4) perspective. There was no clean clustering of superior and non-superior rated data points in all cases, but some visual distinction between the two could be made in figure 3 associated with the 2D extracted components. Hence, these components were added to the database as 2 new columns $tfidf\_tsne\_1$ and $tfidf\_tsne\_2$.

Topic Modelling: Topic modelling (identified 3 latent topics in superior as well as non-superior wines and 2 topics among all wines to try and obtain superior v/s non-superior clusters + visually examine difference of themes among the two) the leverages Latent Dirichlet Allocation (LDA) was done using the $LatentDirichletAllocation$ by $scikit - learn$. [13] Resulting clusters were examined using python's $pyLDAvis$ package in an interactive way to reveal that some indicators in the raw dataset is unlikely to be useful towards predicting $superior\_rating$ as associated words (like Fresh, Fruit, Sweet) appears in both topics associated with superior wines and others in large numbers alike. Also, other important themes like acidity, ripeness, and so one, were not considered in the indicator variables. Inadequacy of indicator variables so discovered, is what inspired feature engineering process (iv) above.

## 5. Analysis

Exploratory Data Analysis (code in the $eda.R$ file).

This step involved simply loading the data set and examining it. First, a pie chart (figure 5) was plotted to view the distribution of the superior (~ 40%) to non-superior (~ 60%) ratings. This was found to be fairly balanced. Thus, that model is less likely to severely overfit either class. Plotting a histogram of prices revealed that it was heavily skewed right (very few large prices) (figure 6). This can negatively impact performance of linear models due to the deviation from the multivariate normality assumption [14]. The $log_{10}$ transformation was applied to correct this (figure 7) (new column $price\_log10$). Next, the distribution of wines per price bracket was plotted and this seemed normal (figure 8) and raised an interesting question: *"Why does the distribution of prices appear skewed at the population level, but normal when grouped by price bracket?"* Some investigation revealed that the reason for this visual skewness is because of the difference between how the prices are segmented in either case. The histogram associated with population level prices was using a fixed bin size of 100 and thus the first bin engulfed 7/9 lowest price brackets as in the real wine pricing system ([\$0- \$4] = EV, [\$4-\$10] = V, [\$10-\$15] = P, [\$15-\$20] = PP, [\$20-\$30] = P, [\$30-\$50] = UP, [\$50-\$100] = L, [\$100-\$200] = SL, [>\$200] = I). When following segmentation true to the underlying pricing process, wine prices are distributed normally. Plotting proportion of superior wines per price bracket as a mosaic plot (figure 9) revealed the first hint as to there being a positive linear relationship between wine prices and the odds of them being ranked as superior. Next, plotting wine count v/s proportion of wines rated superior per variety (figure 10) showed that for most wine varieties in this data set, there is a 20% to 60% chance of wines being rated as being superior. Further, mosaic plots were generated for all 12 of the indicator variables ("Crisp", "Dry", "Finish", "Firm", "Fresh", "Fruit", "Rich", "Ready", "Round", "Soft", "Sweet", "Full" - includes few new ones added in feature engineering phase after LDA revealed them to be relevant) (figure 11). This mosaic plot with 12 subplots showed that while "Fruity" was the most commonly occurring, it along with most others except for "Rich" and "Soft", really convey anything about the superiority of the wine because in most cases, proportion of superior wines when indicator $= 1$ was very similar to when indicator $= 0$. Thus, from this, it was clear that "Rich" and "Soft" are potential predictor variables. Subsequently, distribution of the engineered wine characteristic scores were visualized using a plot (figure 12) containing multiple histograms which were all found to be generally normal looking and thus likely apt for use as a predictor with a linear model as is. Further, to view joint effects, a pair plot was generated using the 5 characteristics score features ($acidity, body, tannin, sweetness, alcohol$) and $price\_log10$ such that markers were colored yellow if wine was superior and blue otherwise (figure 13). Feature $price_log10$plotted against most other features (especially $body$) revealed visual linear separability. Thus, it was deemed that at least one of these interactions should be a part of the model. Another similar pair plot (figure 14) was generated using features $body, tfidf\_tsne\_1, tfidf\_tsne\_2$ and $price\_log10$ which revealed linear separability when $tfidf\_tsne\_1$ plotted against $price\_log10$. Thus, this too would likely be part of the model. The next visualization step involved generating a mosaic plot to view proportions of wines rated superior/not against sentiment score assigned rounded to a single decimal point to produce 14 factors (groups) (figure 15). This plot showed that ratings were predominantly neutral to positive, and that this sentiment rating is likely not a good predictor because w.r.t most data points, the differences in sentiment do not significantly affect the proportion of superior to non-superior ratings. Moreover, there were some mismatches between the sentiment expressed in the review and the score assigned because the most positive sentiments (>= 0.8) were associated with non-superior ratings. This is expected to be either because of inaccuracy of pre-trained model used to obtain the scores or due to the human (judge) tendency to speak nicely of something (wine) so as not to sadden/offend even if they don't truly appreciate it. Nevertheless, here, this feature is too unstable and not useful enough to be a predictor. Finally, a correlation heatmap (figure 16) was plotted among all columns that could possibly be predictors. When choosing variables to include in a linear model, it's important to pick ones such that there is minimum multicollinearity as input parameters being correlated to each other instead of/in addition to the response variable can lead to estimates of the logistic regression model being less precise and less stable (violates assumption of independence). While this may not always be a problem if aim is prediction alone since some of the input features may still be good, truly independent, predictors; because here, we'd like to interpret the general effects of the variables, it's best to keep multicollinearity at a minimum. [15] The correlation matrix here confirmed our hunch from figure 9 before, that price maybe positively correlated with superiority as here $price\_log10$ was the feature with highest positive correlation with $superioir\_rating$. Overall good candidates for predictors based on correlation with the target variable were $price\_log10$, $tannin$, $alcohol$, $acidity$, $body$, $Soft$, $Rich$ and $tfidf\_tsne\_2$. However, to minimize correlation between input features, some of these were avoided and the final choice for predictor variables, while also considering past observations were: $price\_log10$, $tannin$, $alcohol$, $body$, $Rich$, $tfidf\_tsne\_1$ and interactions $price\_log10 * body$ as well as $price\_log10 * tfidf\_tsne\_1$.

MCMC Performance: The $bmrs$ package by default uses the No-U-Turn Sampler (NUTS) sampler, which is a variant of the Hamiltonian Monte Carlo (HMC) sampler for greater efficiency and speed. [16] The command $summary(fit)$ reported $Rhat$ values associated with every parameter to be 1.0. This is an indicator that MCMC chains have converged to a stationary distribution. ESS values, all being much $> 100$ suggest that autocorrelation was low. Also, viewing the posterior distributions and MCMC chains using the $plot(fit, variable = c(...))$ command (figure 17), due to all trace plots appearing akin to random noise, revealed good mixing and random MCMC sampling with low bias covering the search space evenly. Furthermore, all 4 chains arrived at the same random pattern indicating reliable convergence. Thus, we may safely conclude here, that the parameter space has been sufficiently explored and that posterior distributions represent reliable estimates as well as uncertainty (more theory around assessing MCMC performance was discussed at length in assignment 2).

Model Output (code in the $model.R$ file).

Before fitting the model, some predictors like $tfidf\_tnse\_1\_norm$ needed to be normalized due to difference in scale compared to other predictors (others need not be normalized because $price\_log10$ already in log scale, $Rich$ is 0/1 and characteristic scores are [0, 1]). Min max normalization was done. If not normalized, the model can get biased towards higher valued features for no good reason. Log 10 transformation was applied to price to reduce skewness.

**Fit Summary:** Once fitted, the summary was output as follows.

```
Multilevel Hyperparameters:
~variety (Number of levels: 35)
              Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
sd(Intercept)     0.65      0.15     0.41     0.98 1.00     1035     2102
Regression Coefficients:
                              Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
Intercept                       -10.55      1.84   -14.08    -6.88 1.00     2175     2400
price_log10                       4.96      1.19     2.57     7.27 1.00     2153     2356
tannin                            2.79      0.72     1.37     4.21 1.00     4438     2946
alcohol                           2.72      0.60     1.49     3.92 1.00     4403     3301
Rich                              1.02      0.14     0.75     1.30 1.00     4389     2703
body                              0.97      2.83    -4.75     6.51 1.00     2312     2370
tfidf_tsne_1_norm                -4.40      1.77    -7.95    -0.98 1.00     2738     2637
price_log10:body                 -0.56      1.87    -4.19     3.23 1.00     2266     2652
price_log10:tfidf_tsne_1_norm     2.67      1.15     0.47     4.96 1.00     2802     2656
```

*Of all the fixed effects in the model,* $price\_log10$ *has the highest average effect* (comparatively highest coefficient of 4.96) on the log-odds of a wine being rated as superior. This was expected given high correlation with response variable from the correlation matrix. What this means is that a one-unit increase in price (log transformed) is associated with an increase in the log-odds of wine being "superior" by around 5 units on average (estimate = 4.96) give or take around 1 unit ($\sigma = 1.19$). The variable with the next most noticeable effect is $tfidf\_tsne\_1\_norm$ with a negative estimate suggesting a drop in log-odds of success (wine rated superior) with increase in predictor value. Next greatest effect (positive) is asserted by derived features, alcohol, and tannin. The model estimates that a one-unit increase in alcohol/tannin content is associated with an increase of approximately 3 units on average, in the log-odds of wine being rated "superior" (with comparatively least uncertainty $sigma \approx 0.7$). The fact that these derived features have a larger effect size compared to the single indicator "Rich" reinforces the value of having derived them to represent key wine judgement criteria while incorporating more than one indicator. The interaction term of $price\_log10$ and $body$ however, had least effect (smallest coefficient estimate). The model predicts a very low log-odds (Intercept estimate = -10.55) of a wine being rated as "superior" when all other factors are zero. *This hints at a baseline tendency for wines to not get rated as being "superior".* That said, the presence of the random intercept for "variety" with a quite high standard deviation (1.84) gives clues about how some variety-specific effects might be at play.

The fit was also plotted to obtain histograms of posterior distributions associated with each population parameter's coefficient as shown in figure 17. Converting from log scale to probability scale, here, a one unit increase in price in the log scale would lead to an increase in the probability of wine being rated "superior" by a factor of anywhere between $e^{2.57} \approx 13$ and $e^{7.27} \approx 1436$ times. At first glance, these numbers seem super large. But upon deeper analysis, this is plausible because an increase of 13 – 1436 is relative to some baseline and not absolute change. So, if the baseline probability of a wine being rated "superior" was low, say 0.01, even a 1500-fold increase would only be an absolute increase in probability =0.01×1500=15%. Because the population parameter intercept was very negative (around -10) in the summary, it might be that baseline probability really is low. Nevertheless, the wide gap (1436 – 13 =1423) indicates low precision. Generally, if a distribution is centered away from 0 and the credible interval does not have 0, it suggests evidence for a non-zero effect of that variable. That is, the further the distribution is from 0 and the narrower the credible interval, the stronger the evidence. [1] [2] Bearing this in mind, it can be seen here that barring interaction terms, all others seem to be having fair evidence against zero effect. That is, the chance that relationships (positive/negative) observed here between wine characteristics (alcohol, richness, tanning, price) and its likelihood of being ranked as superior, is less likely to be due to random chance.

So far, fixed effects on the population level was being investigated. Next, random effects on the variety level was explored. Samples were obtained using the $brms::as\_draws\_matrix$ function and then a figure with box plots showing the probability of wine being rated superior after accounting for fixed effects per variety was plotted (figure 18 below).

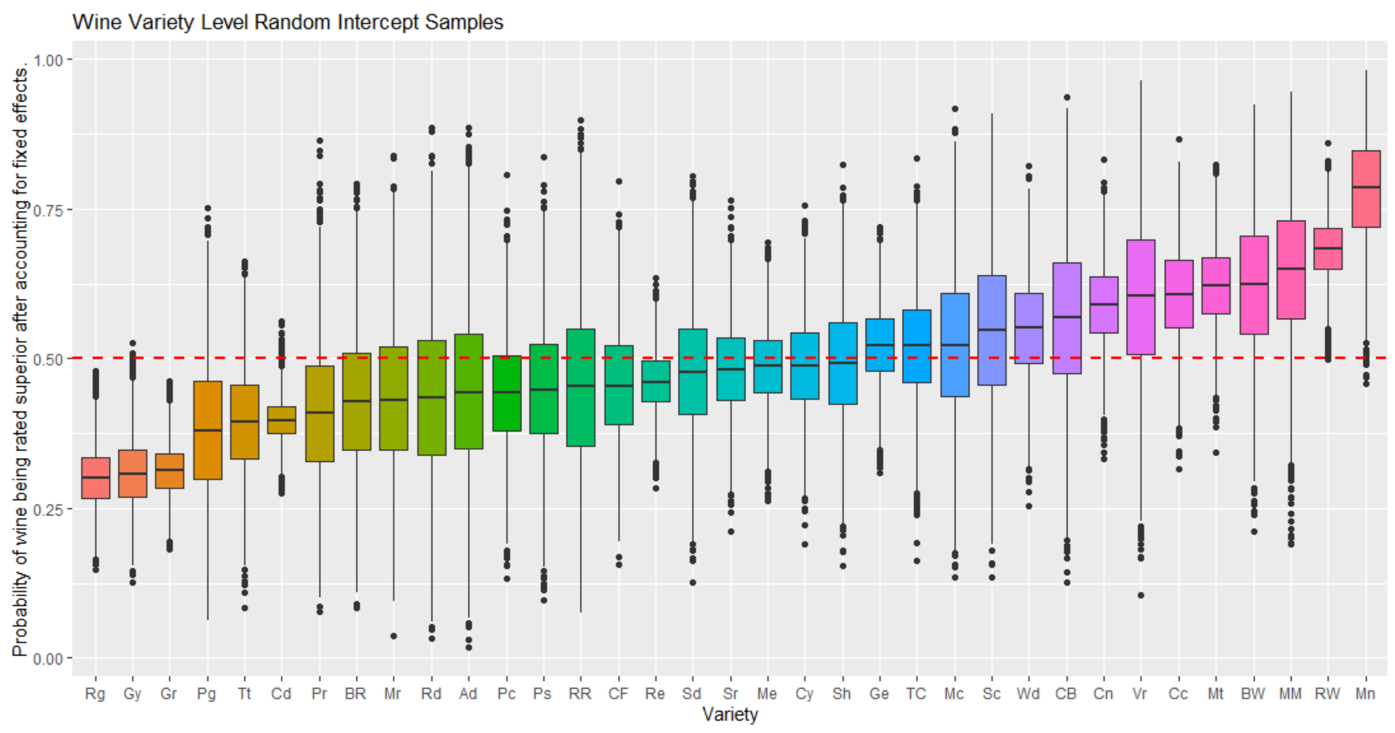


*Figure 18. Distribution of the probability of a wine getting rated as superior per variety after accounting for fixed effects.*

Each box in the plot represents the distribution of posterior draws for a specific wine variety. Had the intercepts been directly plotted on the y axis, this would have indicated the per-variety baseline log-odds of a wine being rated "superior" after accounting for the fixed effects in the model due to the predictor variables. Since that might be harder to readily infer from, log-odds $(x)$ were converted to probabilities via $\frac{e^x}{1+e^x}$. Now, the y-axis indicates baseline probability that a wine of a particular variety gets rated as "superior" after accounting for fixed effects in the model due to predictor variables. The 3 wine varieties at the leftmost end of the figure) (Riesling, Gamay, Gewürztraminer) are the ones with the lowest probability of having wines be rated as superior with minimal variance (more certainty). Varieties before Cn (Cabernet Sauvignon), all have median intercept value below/very close to 0.5, meaning that consequently suggests that the odds of these varieties of wines being rated high enough to be ranked as superior is low. The wine variety with the highest odds of being rated as superior is "Melon" on the extreme right of the figure. It's median, as well as that of neighboring varieties ("Rhône-style White Blend", "Malbec-Merlot", "Bordeaux-style White Blend", …, "Cabernet Sauvignon") lie quite above the 50% probability and thus have discernably higher odds of being ranked as superior to the varieties associated with the boxplots at the extreme left end. Thus, one may conclude that if one was to try a vine of the "Melon" variety, it's possible that it has some of the same properties as highly rated wine. Overall, some wine varieties do indeed get ranked high more often than others with few, in this dataset, having lowest odds $(< 40)$ of being ranked as superior. That said, wine ranking is based on extremely subjective views. This likely explains the relatively wide spread of distributions and ample outliers in most cases.

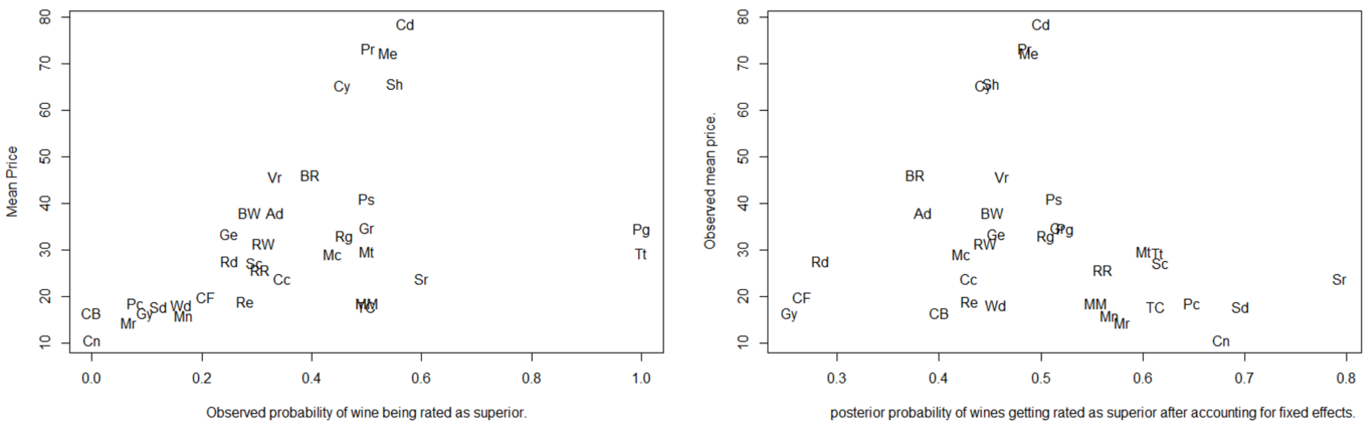


*Figure 19. Price v/s probability of wine being rated as superior before and after accounting for fixed population parameters.*

Next, original mean price was plotted against per variety probability of wine being ranked superior (figure 19 above). Here, subplot 1 shows the unadjusted relationship between price (y axis) and the observed probability of superior rating for each wine variety (x axis). Figure 19, right subplot (subplot 2) shows the observed mean price (y axis) v/s baseline posterior probability (log odds exponentiated as $\frac{e^x}{1+e^x}$ on x axis) of wines getting rated as superior per variety after accounting for fixed effects. Overall, it can be observed that probability of many wines, especially ones with a high price in figure 19 left plot (subplot 1), dropped in subplot 2. And many wines that had low probability and price in subplot 1, showed increase from that value in subplot 2. An interesting observation was that going from subplot 1 to subplot 2, probability of high-priced varieties (e.g. Cd = Champagne Blend) dropped while that of cheaper ones (e.g. Mn = Melon) increased. Varieties like "Cn" (Cabernet Sauvignon) and "CB" (Chenin

Blanc-Chardonnay) with very low observed superior wine probability (subplot 1) might have been assigned higher probabilities (55-60%) in subplot 2 because these varieties may have other characteristics than price, that are generally associated with superior wines. This suggests that price might be partially responsible for observed probabilities and it's possible that some varieties like Cd = Champagne Blend may be overpriced. So, *while it is true that price is positively correlated with wine rating, high price does not necessarily mean high quality wine. So, you don't have to break the bank to have great wine. There are plenty of affordable options***.** Here, *the model recommends wine varieties like Melon (Mn), Malbec-Merlot (MM) and Rhône-style White Blend (RW) as great affordable wines* that may be better overall than some of the high-priced wines based on properties like (alcohol, tannins, body). Other extreme changes can be observed in the probability of success associated with some varieties like Pg (Petit Manseng) and Tt (Tannat) which had 100% of observed wines rated superior in subplot 1, but in subplot 2 this drastically dropped to probability of a little less than 40%. This is expected to be because the initial high rating might have been partly due to a small sample size or specific preferences of the raters for those wines. By accounting for these factors, the model might have adjusted the probability downwards to a more general level for these varieties.

Furthermore, the 2.5%, 50% and 97.5% quantiles associated with each variety was compiled into a matrix and top few varieties with most negative and positive effect on the response variable was printed out. These were then plotted on a box plot (figure 20). This showed that the probability of wine being rated as superior is affected by the variety of wine. This chance is low (a little above 30%) if the wine variety was Gr = Grenache, but increases to around 80% if the variety is Melon. We can, however, see that there is significant uncertainty in all cases, although comparatively lower for varieties Re (Rosé) and Gr (Grenache).

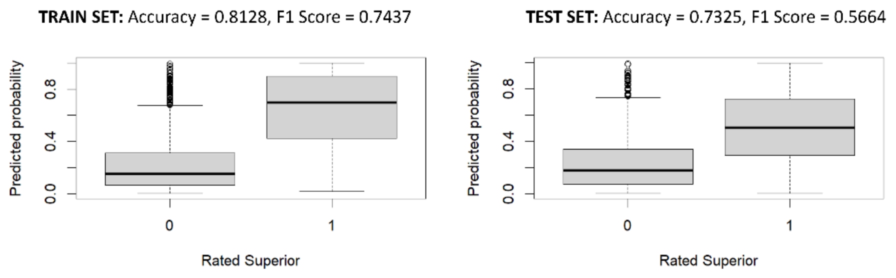


*Figure 21. Fitted model's predictions on both train and test set.*

Predictions

The fitted model was tested on both this dataset that it was fitted on (train set) as well as new data (test) obtained from here. Test data was subject to the same processing as was done on the train set like previously discussed. To judge the performance of the model, a confusion matrix was generated and both the accuracy and F1 score metrics were compute. F1 score (balance of Precision and Recall) was computed because it's value, unlike accuracy, is less prone to being misleading if datasets are highly imbalanced. [17] Results were as shown in figure 21. In both cases, the F1 score is lower than accuracy suggesting that there is an imbalance between precision and recall. There is also some overfitting given than accuracy and F1 scores on the testing set are lower than on the training set. Overall, the model is exhibiting some overfitting on training data as is evident due to the clear difference in F1 scores (there is likely an imbalance between false positives and negatives as the accuracy is still $\geq 70$ despite F1 score being quite a bit (0.18) lower). Given large spread, there is significant uncertainty as well. The model performs well on data it has already seen but faces difficulty regarding generalizing to unseen data in the test set. This can lead to unreliable predictions on new data.

Hence, this model must be further improved or alternative approaches should be considered before beginning to make predictions in the real world.

## 6. Conclusions

As for key observations, as given in the central figure, the first insight was that price was the best predictor (highest effect and correlation) among available ones w.r.t determining whether wine is rated as superior or not. Examining the random effects intercept also revealed that some wine varieties are indeed more likely to be ranked high compared to others, although all varieties did present significant variance which is assumed to be because of the subjective nature of wine judgement. Removing the effect of price revealed that some cheaper wines may be just as likely as the more expensive wines to be rates as superior. In a practical sense, this may be interpreted as there being wine varieties that are affordable, yet likely just as good as some of the more expensive ones.

W.r.t success of methods applied in this project, the fact that engineered features did prove to be good predictors suggests that the decision to distill reviews through NLP methods to obtain a more informative score comprising multiple key characteristics of wine quality, was a good idea. Also, the fact that the model overfit the training set with good accuracy and F1 scores suggest that the model is indeed capable of capturing the complexity inherent within the data and that input features which were carefully selected and provided were indeed useful ones (an initial version of the model included all engineered features as well as indicator variables, but it performed worse which is expected to be because of too much multicollinearity and added complexity; this experience inspired a more careful feature selection the next time round, that resulted in this model).

That said, there is much room for improvement as the resulting model did not generalize well to new data. One way to reduce overfitting may be to introduce regularization operations like L1 or L2 regularization. [18] Other possible future directions for this project includes trying a slightly different model with perhaps another level of modelling wherein the standard deviation and mean of the random effects intercept is also set as hyperparameters with their own priors. Further experimentation with more values (say some of the slope terms in addition to the intercept) varying across varieties may lead to a more robust model. [2] It may also be interesting to set the grouping parameter as price brackets and turn the problem on its head and try to predict prices instead. Also, a custom sentiment analysis model may be built and trained on wine reviews and scores so obtained can be another useful predictor.

## References

[1] M. Dietze, "Lesson 22b Hierarchical Bayes: Random Effects," YouTube, 24 March 2020. [Online]. Available: https://www.youtube.com/watch?v=_L-Lud336Ts. [Accessed 3 April 2024].

[2] LADAL, "Bayesian generalized linear mixed models with brms: Bodo Winter LADAL Webinar 2022," YouTube, 2 November 2022. [Online]. Available: https://www.youtube.com/watch?v=UwIgWD5pWzQ. [Accessed 4 April 2024].

[3] jbstatistics, "Introduction to the t Distribution (non-technical)," YouTube, 5 May 2013. [Online]. Available: https://www.youtube.com/watch?v=Uv6nGIgZMVw. [Accessed 5 April 2024].

[4] P.-C. Bürkner, "Prior Definitions for brms Models," brms, [Online]. Available: https://paul-buerkner.github.io/brms/reference/set_prior.html. [Accessed 5 April 2024].

[5] wine-searcher, "Wine Scores," wine-searcher, [Online]. Available: https://www.wine-searcher.com/wine-scores. [Accessed 5 April 2024].

[6] A. S. Gillis, "What is lemmatization?," Tech Target, March 2023. [Online]. Available: https://www.techtarget.com/searchenterpriseai/definition/lemmatization. [Accessed 4 April 2024].

[7] Wine Folly, "Reality of Wine Prices," [Online]. Available: https://winefolly.com/lifestyle/reality-of-wine-prices-what-you-get-for-what-you-spend/. [Accessed 3 April 2024].

[8] Wine Folly, "How basic wine characteristics help you find favorites.," Wine Folly, 2024. [Online]. Available: https://winefolly.com/deep-dive/wine-characteristics/. [Accessed 4 April 2024].

[9] B. Stecanella, "Understanding TF-ID: A Simple Introduction," MonkeyLearn, 10 May 2019. [Online]. Available: https://monkeylearn.com/blog/what-is-tf-idf/. [Accessed 5 April 2024].

[10] Unfold Data Science, "Feature Extraction techniques from text - BOW and TF IDF|What is TF-IDF and bag of words in NLP," YouTube, 24 April 2020. [Online]. Available: https://www.youtube.com/watch?v=76jmgV_ZPUs. [Accessed 4 April 2024].

[11] Unfold Data Science, "Countvectorizer and TF IDF in Python|Text feature extraction in Python," YouTube, 28 April 2020. [Online]. Available: https://www.youtube.com/watch?v=lBO1L8pgR9s&t=270s. [Accessed 5 April 2024].

[12] A. A. Awan, "Introduction to t-SNE," datacamp, March 2023 . [Online]. Available: https://www.datacamp.com/tutorial/introduction-t-sne. [Accessed 4 April 2024].

[13] A. K. Adesemowo, "#Tutorial: Topic Modelling visualisation (with SciKit [sklearn!]) using Gensim Lee Background Corpus," LinkedIn, 26 July 2022. [Online]. Available: https://www.linkedin.com/pulse/tutorialtopic-modelling-visualisation-scikit-sklearn-using-adesemowo/. [Accessed 3 April 2024].

[14] S. Poyrekar, "How does skewness impact regression model?," Quora, [Online]. Available: https://www.quora.com/How-does-skewness-impact-regression-model. [Accessed 5 April 2024].

[15] J. Frost, "Multicollinearity in Regression Analysis: Problems, Detection, and Solutions," Statistics By Jim, February 2017. [Online]. Available: https://statisticsbyjim.com/regression/multicollinearity-in-regression-analysis/. [Accessed 5 April 2024].

[16] P.-C. Bürkner, "brms: An R Package for Bayesian Multilevel Models," [Online]. Available: https://cran.r-project.org/web/packages/brms/vignettes/brms_overview.pdf. [Accessed 6 April 2024].

[17] Natasha Sharma, "Understanding and Applying F1 Score: AI Evaluation Essentials with Hands-On Coding Example," arize, 6 June 2023. [Online]. Available: https://arize.com/blog-course/f1-score/. [Accessed 6 April 2024].

[18] LinkedIn, "How do you explain the concept of regularization in logistic regression to a non-technical audience?," [Online]. Available: https://www.linkedin.com/advice/3/how-do-you-explain-concept-regularization-logistic. [Accessed April 2024].

[19] jpalm, "Exclusion of 0 in 95% Credible Interval," StackExchange, 2 July 2021. [Online]. Available: https://stats.stackexchange.com/questions/533023/exclusion-of-0-in-95-credible-interval. [Accessed 5 April 2024].

[20] jsocolar et al., "Bayesian credible intervals: Significant compared to what?," Stan, May 2021. [Online]. Available: https://discourse.mc-stan.org/t/bayesian-credible-intervals-significant-compared-to-what/22700. [Accessed 5 April 2024].