CS7DS3 Main Assignment

NAMAN ARORA
TCD ID: 19323369
MSc Computer Science - Future Networked Systems

I. Executive Summary/Objective Statement

In order to determine the variables that affect wine evaluations and categorise wines into superior and non-superior groups, I examined a dataset comprising reviews of 2,500 French wines for this research. My analysis seeks to offer both technical and non-technical readers insightful information. This paper highlights the varieties that are reasonably priced and the price ranges that are most likely to produce wines of higher quality for non-technical stakeholders, such as wine dealers. I make sure that the statistical models and methods are accurate and rigorous for technical stakeholders, such as data analysts and team leaders.

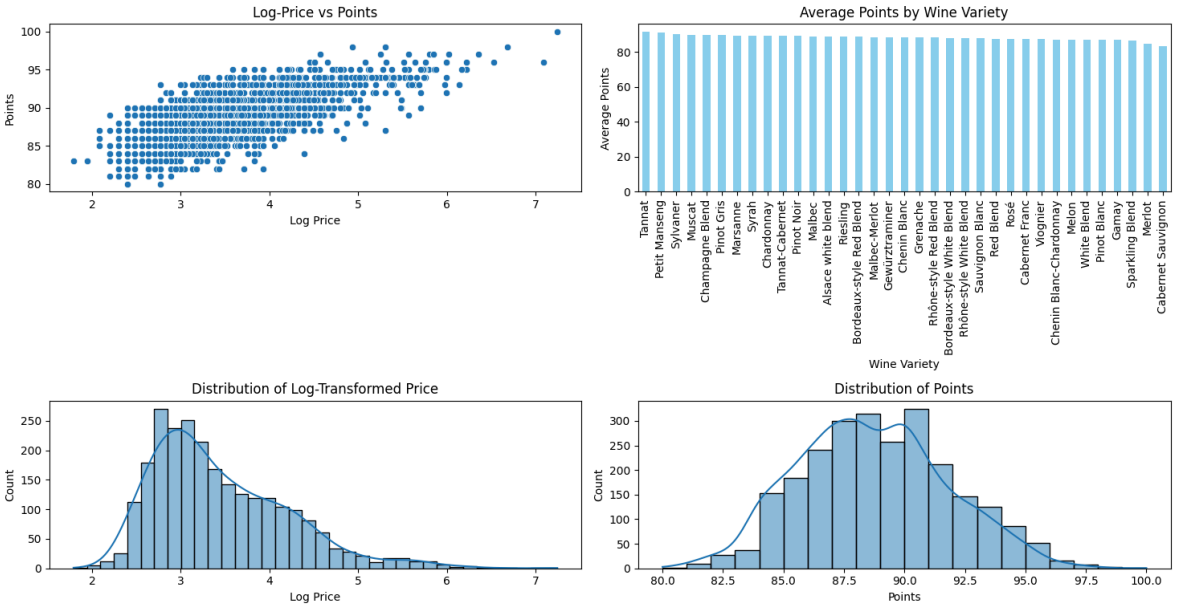
To accomplish the goals, I used a mix of statistical models and data visualisation methods:

- In order to pinpoint the precise wine scores and identify the major factors affecting the ratings, regression models were employed.
- Classification models were created to set better wines apart from the rest according to their characteristics.
- In order to guarantee convergence and estimate parameters with uncertainty, Bayesian inference was utilised.
- Violin plots, ROC curves, and confusion matrices are examples of statistical visualisations that were used to compare scores between varieties and show categorization performance.

Key Findings:

1. Impact of pricing and Variety: There is a positive correlation between wine ratings and log-transformed pricing, and certain varietals regularly score higher than others.
2. Model Accuracy: My regression models successfully predict accurate scores with a mean squared error of **4.33,** while my classification models identify excellent wines with good accuracy (**~78%**).
3. Interpretation of the Data: The Bayesian model produces trustworthy posterior estimates with well-converged chains, confirming the links discovered in the regression and classification models.

The major picture, which is presented after this overview, effectively illustrates the linkages and model performances by consolidating these findings through graphical representations. More details about my data, methods, and analysis are provided in the following sections, which also include suggestions for more research.



Summary of Findings:
1. Wines with higher log-transformed prices tend to receive higher ratings, as shown in Plot 1.
2. The average points by wine variety (Plot 2) indicate that certain varieties consistently receive higher ratings.
3. The distribution of log-transformed prices (Plot 3) is approximately normal, indicating the effectiveness of the transformation.
4. The distribution of points (Plot 4) shows a slight skew toward higher ratings, with many wines clustered between 85 and 92 points.

Central Figure

II. Data Description

Two thousand five hundred wine evaluations from Wine Enthusiast make up the dataset used for this analysis. Every review offers details on the qualities of a certain wine, such as the price, variety, rating score (points), and an evocative text. This dataset's primary variables are:

- Points: A rating system consisting of a number between 80 and 100, where wines with a score of 90 or higher are deemed "superior."
- Superior Rating: A binary variable that designates as "superior" any wine that receives a score of 90 or higher (zero otherwise).
- Cost: The significantly distorted retail price of each wine in USD. This was normalised using a log-transformation that I did.
- Variety: Statistical information pertaining to the sort of wine or grape variety. The dataset contains more than twenty different kinds.
- Wine descriptors are categorical variables that indicate the particular characteristics of each wine, such as "Crisp," "Fruit," "Rich," and so on.
- Description: A free-text area with the reviewer's qualitative observations about the qualities of the wine. An assessment of sentiment polarity was determined using this text.

Data Transformations and Preprocessing

- Log-Transformed Price: I used a log-transformation to normalise the price variable's distribution because of its skewness.
- Target Encoding of Variety: In order to capture the average influence of each variety on the points rating, the categorical variety variable was encoded using target encoding.
- Sentiment Analysis: To determine how positive or negative each review was, sentiment polarity was taken out of the free-text description.

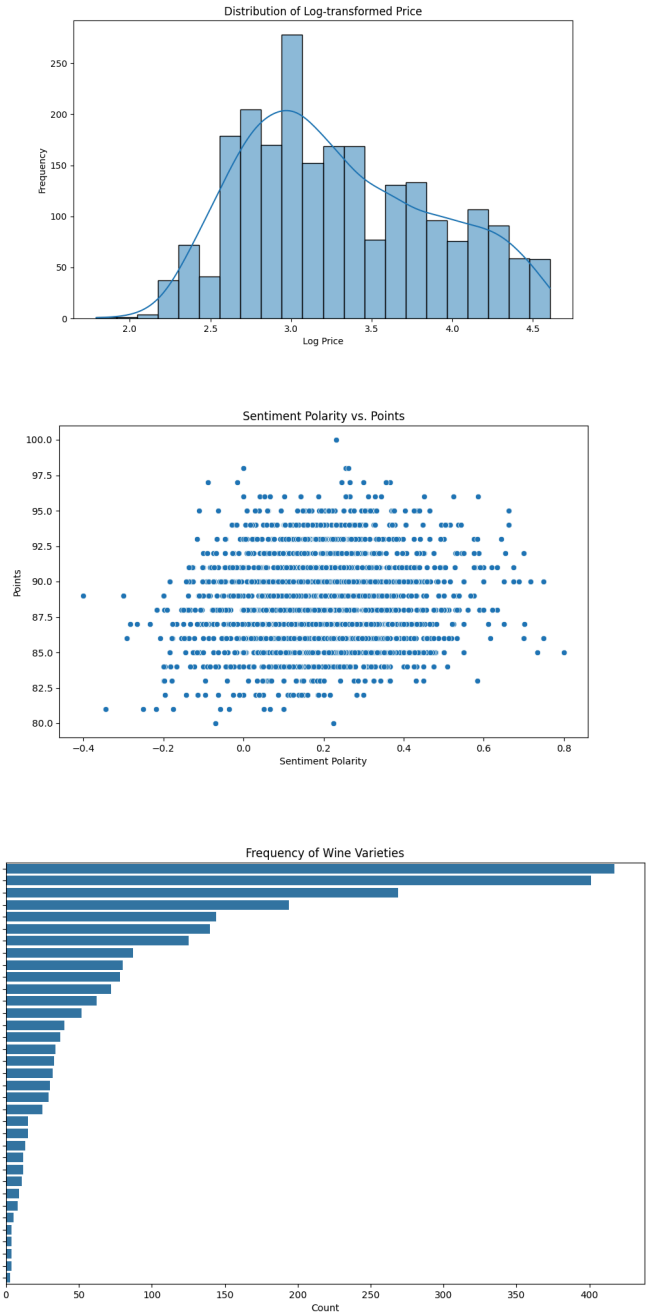


Fig: PLots showing different log price frequency, sentiment polarity, and wine variety distribution

Dataset Challenges and Assumptions

1. Missing Values: Preprocessing was made easier because the dataset included no missing values.
2. Class Imbalance: In the binary classification task, there is a little class imbalance because wines categorised as "**superior**" only make up about **39%** of the data.
3. Feature Correlations: Preliminary data analysis revealed correlations between a few descriptors and points, suggesting that particular flavour/aroma attributes are typically linked to higher ratings.
4. Text Data: Although more preprocessing was needed for feature engineering, free-text descriptions were helpful for extracting sentiment.

III. Analysis: Models and Methods

Several models and strategies were employed to analyse the wine reviews dataset. These are organised into the following sub-sections:

Analysing exploratory data (EDA)

- Univariate Analysis: To comprehend the distributions of each variable, I looked at them separately. For example, the points variable had an approximately normal distribution, whereas the price variable was substantially skewed, resulting in a log-transformation.
- Bivariate Analysis: Log-transformed pricing and point ratings showed a positive association when relationships between the variables were visualised using scatter plots and bar plots.
- Multivariate Analysis: The point distributions of different wine varietals were compared using box and violin plots, which showed that some varietals routinely get higher ratings than others.

Models of regression

- The Random Forest Regressor To determine each wine's precise point score, the ensemble approach was used. The target-encoded variety, sentiment polarity scores, and log-transformed pricing were among the features of the model. The mean squared error of the Random Forest model was about 4.33.
- Bayesian Regression: To estimate parameter distributions, a Bayesian model utilising MCMC sampling was constructed. Credible intervals for each feature were provided by the posterior distributions of the coefficients, indicating that **greater costs had a positive effect on ratings**.

Models of Classification

- Gradient Boosting Classifier: The objective of this ensemble model was to categorise wines into two groups: superior (ranked 1) and non-superior (0). With features borrowed from regression models, the model's overall classification accuracy was about **78%**. Plotting a confusion matrix and ROC curve allowed us to see how well the classifier performed.
- Visualisation of the Decision Boundary: In order to display the decision regions of the model, I used a mesh grid to illustrate the classification decision boundary. This plot

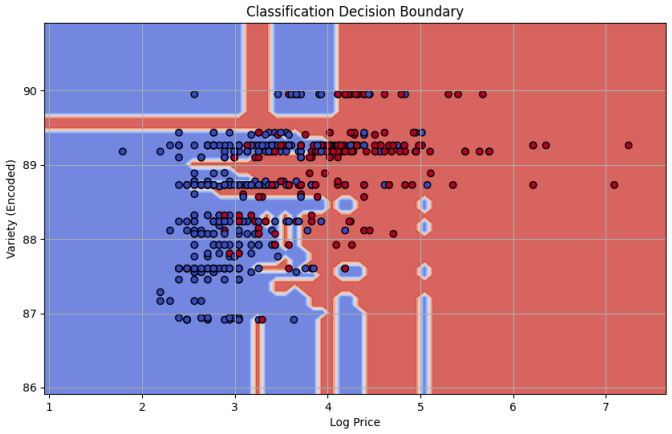
shed light on how the classifier distinguishes between wines that are superior and those that are not.

Bayesian Deduction and MCMC Examination

- Parameter Estimation: MCMC sampling was used to derive posterior distributions for each coefficient (beta parameters) and standard deviation (sigma), which resulted in uncertainty estimations.
- Diagnostics for Convergence: R-hat statistics and trace plots were employed to verify chain convergence, suggesting that the MCMC sampling attained a stable posterior distribution.

Visualisations of Statistics

- Plots of violins and swarms were utilised to compare wine scores among the best kinds and show variations in score distributions.
- Probability Plots: The effectiveness of classification models was evaluated using ROC curves, confusion matrices, and classification reports.



<u>Explanation of this above figure:</u>

Box Plot of Wine Scores by Variety:

- Shows how different wine varietals' ratings for wine points are distributed. Box plots show the outliers (dots), quartiles (box), and median (horizontal line). **Champagne Blend and Rosé**

are two varieties with higher median ratings; mixes in the **Bordeaux style** are more variable.

Violin Plot of Wine Scores by Variety:

- Shows the distribution shape and median of scores for each wine type by combining the features of box plots and density plots. Variations in score distributions are indicated by different density forms, with **Champagne Blend displaying a more condensed range**.

Swarm Plot of Wine Scores by Variety:

- The whole distribution of scores is displayed by grouping individual wine scores into dots. demonstrates how scores from different kinds overlap and form clusters, **suggesting that certain wines are rated higher than others.**

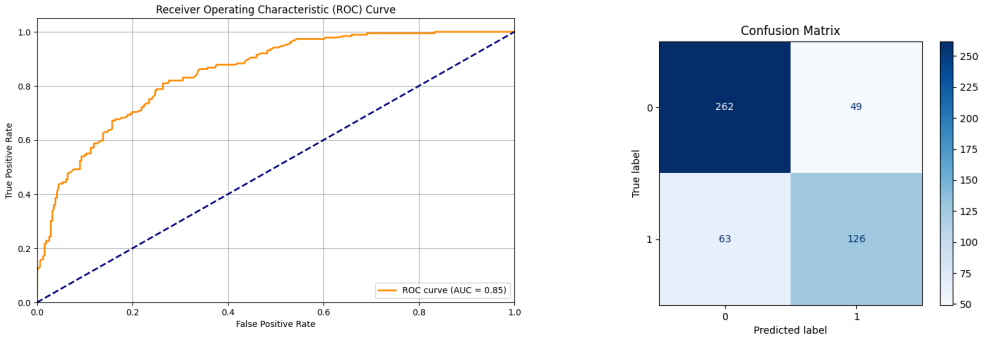


Fig: Model output parameters, confusion matrix (FP, FN etc) and ROC curve showing accuracy area

Formal Model Specification:

1. Random Forest Regressor:

- Let $X = (x_1, x_2, \ldots, x_p)$ represent the feature vector of a given wine.
- Each decision tree T_k in the forest provides a prediction $f_k(X)$.
- The overall model output $\hat{y}$ is given by:
  $\hat{y} = \frac{1}{K} \sum_{k=1}^{K} f_k(X)$
  where K is the total number of decision trees.

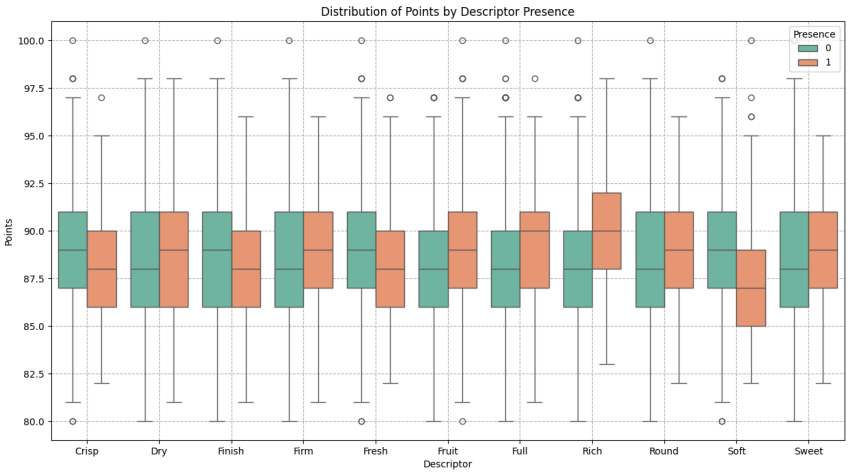
2. Gradient Boost Classifier

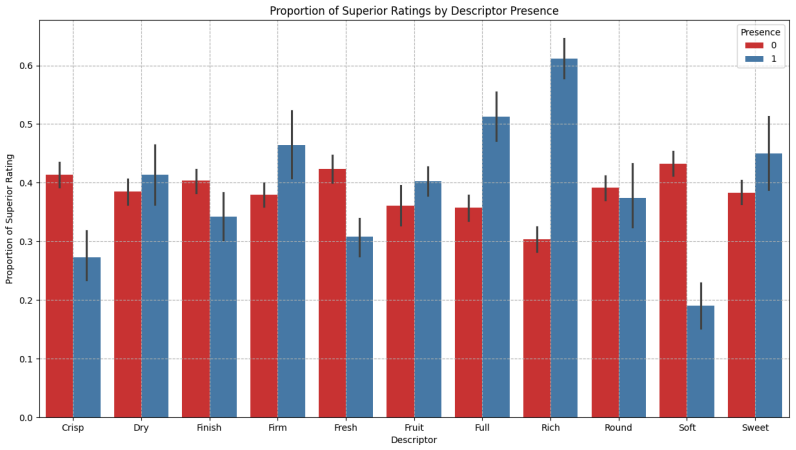
- Let X be the feature vector and $F_0(X)$ represent the initial model.
- For each subsequent model m, the residual errors of the previous model are minimized:
  $r_i^m = y_i - F_{m-1}(X_i)$
  where r_i^m represents the residuals of sample i at iteration m.
- The final model prediction is:
  $F_M(X) = F_0(X) + \sum_{m=1}^{M} \gamma_m h_m(X)$
  where $h_m(X)$ is the weak learner at iteration m and γ_m is a learning rate.

3. Bayesian Linear Regression Model

- Let $X_i = (x_{i1}, x_{i2}, \ldots, x_{ip})$ represent the feature vector of the i-th wine.
- The response variable y_i (wine points rating) is modeled as:
  $y_i \sim \mathcal{N}(\mu_i, \sigma)$
  where:
  $\mu_i = \beta_0 + \sum_{j=1}^{p} \beta_j x_{ij}$
  and σ is the residual standard deviation.
- Priors are assigned to each parameter:
  $\beta_0, \beta_1, \ldots, \beta_p \sim \mathcal{N}(0, 10)$
  $\sigma \sim \text{HalfNormal}(1)$

Analysis of Binary Features (Dry, Crisp, Sweet):





Distribution of Points by Descriptor Presence:

- Examines the distribution of points for different wines first according to whether certain characteristics (such "dry," "crisp," etc.) were used in the review.
- **"Full" and "Rich" wines typically get higher median scores than wines without these designations.**

Proportion of Superior Ratings by Descriptor Presence:

- Based on the presence or absence of each descriptor, the fraction of wines obtaining a superior rating (90+ points) is displayed in a bar plot.
- **It is more likely that a wine with characteristics like "Rich," "Full," and "Sweet" will be evaluated as outstanding.**

IV. Conclusions

Summary of Results

In order to categorise wines into superior and non-superior groups and to comprehend the major elements impacting wine ratings, I investigated a variety of statistical models and visualisation techniques in this investigation. Important outcomes consist of:

1. Log-Transformed Price: Both the Random Forest and Bayesian regression models demonstrate that higher log-transformed prices are positively correlated with higher point ratings.
2. **Influence of Wine Variety: Violin and swarm plots show that certain wine varietals, such Champagne Blend and Rosé, routinely score higher than others.**
3. Model Execution: Mean Squared Error (MSE) of 4.33 indicates that wine scores were accurately predicted by Random Forest Regression.
4. Gradient Boosting Classification: 78% classification accuracy was achieved by classifying wines into superior and non-superior categories.
5. Bayesian Inference: The most significant predictor was the log-transformed price, as seen by the reliable intervals that Bayesian regression produced around coefficient estimations.

Overall Evaluation

The dataset's important patterns and linkages were successfully found by the models and techniques used:

- EDA and Transformations: By normalising the skewed distribution, the price log-transformation helped produce forecasts that were more accurate.
- Regression Models: Accurate wine evaluations were robustly predicted by the Random Forest Regressor and Bayesian models.
- Models of Classification: Good accuracy and dependable classification boundaries were attained by the Gradient Boosting Classifier.

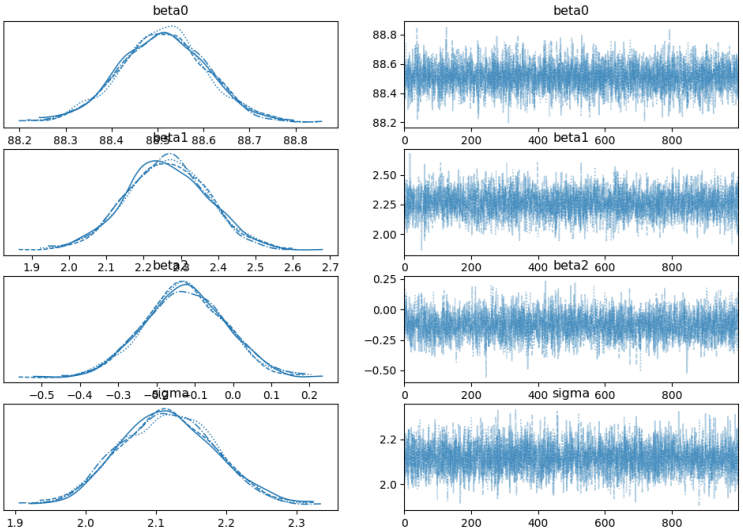
- Statistical Visualisation: Differences in rating distributions between kinds were made clearer by using violin plots and decision bounds.

Both expert and non-technical audiences were able to get practical insights from the mix of models and visualisation tools.

Further Recommendations

- Alternative Classification Models: To compare predicted accuracy and feature relevance, use alternative models like neural networks or Support Vector Machines (SVMs).
- Textual Analysis: To identify important features that affect ratings, make even more use of the description field by utilising NLP techniques like word embeddings or topic modelling.
- Expanded Dataset:
    - For further generalizability, look into other areas or bigger datasets.
    - Incorporate with market sales information to forecast customer inclinations.
- Grouping and Clustering:
    - To find high-performing segments, group different wine varietals into larger clusters.
    - To uncover hidden patterns, apply clustering techniques like as hierarchical clustering or K-means clustering.

These suggestions, together with the knowledge gathered from this analysis, will aid in the improvement of upcoming wine review studies to assist well-informed choices made by the wine business.



MCMC TRACE PLOTS

APPENDIX

Description of **Python** Methods and Packages Used

1. Pandas and Numpy:

- The dataset was loaded and modified into a DataFrame format using the Pandas package. Its techniques aided in the aggregation, encoding, and cleansing of data.
- Numpy: Numpy was a crucial tool for numerical calculations, particularly for transforming data and creating mesh grids for decision boundaries.

2. Seaborn and Matplotlib:

- Violin plots, swarm plots, and scatter plots were produced using the statistical visualisation toolkit Seaborn. The association between points and characteristics like price and diversity was revealed by these charts.
- Matplotlib: To graphically summarise and interpret the data, ROC curves, confusion matrices, and custom subplots were created using this core visualisation package.

3. Scikit-Learn:

- Creating Models To categorise wines as superior or non-superior and to precisely estimate their point values, the Random Forest Regressor and Gradient Boosting Classifier models were used. Splitting the data was made easier using the train_test_split function.
- Metrics: The performance of the model was assessed using functions such as roc_curve, AUC, and classification_report.

4. PyMC3: Bayesian Inference: Using Markov Chain Monte Carlo (MCMC) sampling, PyMC3 offered the means to build and sample from a Bayesian linear regression model. To estimate uncertainty, credible intervals and trace plots were employed.

5. Category Encoders: Target Encoding: I was able to use target encoding with this package to encode the wine variety feature that is categorical. Using the mean points score as a basis, this method assists in converting the variation into a numerical representation.

6. TextBlob: Sentiment Analysis: The wine description text was subjected to sentiment analysis using the TextBlob library, which produced a sentiment polarity score that indicated how positive or negative the evaluations were.

CODE

A. model.py

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from textblob import TextBlob
import category_encoders as ce
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.model_selection import train_test_split, cross_val_score
from sklearn.linear_model import LogisticRegression
from sklearn.ensemble import RandomForestRegressor, GradientBoostingClassifier
from sklearn.metrics import classification_report, mean_squared_error
from sklearn.preprocessing import PolynomialFeatures


df = pd.read_csv('wine_review.csv')

print(df.head())
df.info()

df['sentiment'] = df['description'].apply(lambda x: TextBlob(x).sentiment.polarity)


plt.figure(figsize=(10, 6))
sns.scatterplot(x='sentiment', y='points', data=df)
plt.title('Sentiment Polarity vs. Points')
plt.xlabel('Sentiment Polarity')
plt.ylabel('Points')
plt.savefig('sentiment_vs_points_model_py.png')


encoder = ce.TargetEncoder(cols=['variety'])
df['variety_encoded'] = encoder.fit_transform(df['variety'], df['points'])


df['log_price'] = np.log1p(df['price'])
poly = PolynomialFeatures(degree=2, include_bias=False)
df_poly = poly.fit_transform(df[['log_price']])
df_poly_features = pd.DataFrame(df_poly,
columns=poly.get_feature_names_out(['log_price']))
df = pd.concat([df, df_poly_features], axis=1)
```

```
X = df[['log_price', 'variety_encoded', 'sentiment'] +
list(df_poly_features.columns)]
y = df['points']
y_class = df['superior_rating']


X_train, X_test, y_train, y_test, y_class_train, y_class_test = train_test_split(
   X, y, y_class, test_size=0.2, random_state=42)


regressor = RandomForestRegressor(n_estimators=100, random_state=42)
regressor.fit(X_train, y_train)
predictions_reg = regressor.predict(X_test)
mse = mean_squared_error(y_test, predictions_reg)
print(f"Mean Squared Error: {mse}")


classifier = GradientBoostingClassifier(n_estimators=100, learning_rate=1.0,
max_depth=1, random_state=42)
classifier.fit(X_train, y_class_train)
predictions_class = classifier.predict(X_test)
print(classification_report(y_class_test, predictions_class))


scores = cross_val_score(classifier, X_train, y_class_train, cv=5)
print(f"Average Accuracy: {scores.mean()}")
```

B. MCMC.py

```
import pymc3 as pm
import arviz as az
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt

df = pd.read_csv('wine_review.csv')

sample_size = 500
df_sample = df.sample(n=sample_size, random_state=42)

import category_encoders as ce
encoder = ce.TargetEncoder(cols=['variety'])
```

```
df_sample['variety_encoded'] = encoder.fit_transform(df_sample['variety'],
df_sample['points'])

df_sample['log_price'] = np.log1p(df_sample['price'])

X_bayesian = df_sample[['log_price', 'variety_encoded']].values
y_bayesian = df_sample['points'].values

X_mean = np.mean(X_bayesian, axis=0)
X_std = np.std(X_bayesian, axis=0)
X_bayesian_normalized = (X_bayesian - X_mean) / X_std

def run():
   with pm.Model() as model:
       beta0 = pm.Normal('beta0', mu=0, sigma=10)
       beta1 = pm.Normal('beta1', mu=0, sigma=10)
       beta2 = pm.Normal('beta2', mu=0, sigma=10)

       mu = beta0 + beta1 * X_bayesian_normalized[:, 0] + beta2 *
X_bayesian_normalized[:, 1]

       sigma = pm.HalfNormal('sigma', sigma=1)

       y_obs = pm.Normal('y_obs', mu=mu, sigma=sigma, observed=y_bayesian)

       trace = pm.sample(1000, tune=1000, return_inferencedata=True)

       return trace


if __name__ == '__main__':
   trace = run()
   az.plot_trace(trace)
   plt.savefig('trace_plot.png')

   summary = az.summary(trace)
   print(summary)
```