VinoMetrics Project Notes

💡 **This document is designed to show our decision-making process through the project. It is not an entirely coherent body of text but should offer insight into our working through the project**.

Table of Contents

Useful Links

UX/UI Presentation

Project Repository

Week plan

Pre-work

Dataset selection

- Choose a compelling dataset

Dataset Search & Evaluation

- Verify the dataset size, quality, and relevance
- Ensure it's suitable for linear regression analysis

Monday

Overview Presentation

- Present project idea, covering the chosen topic, dataset, and what we intend to explore or predict with our linear regression model
- Gather feedback to refine our approach if needed

Dataset Search & Evaluation

- Verify the dataset size, quality, and relevance

- Ensure dataset suitability for linear regression analysis

Initial Data Exploration

- Conduct a quick exploratory data analysis (EDA) to understand the dataset's structure, variables, and any apparent trends or issues like missing values

Detailed EDA

- Dive deeper into the data to uncover patterns, correlations, and distributions
- Identify potential features for our linear regression model

Data Cleaning

- Address missing values, outliers, and incorrect data types
- Ensure the dataset is clean and ready for further analysis

Tuesday

Data Preprocessing

- Based on our EDA, prepare the data for modelling. This may include feature engineering, normalisation or standardisation of variables, and encoding categorical features

Model Development

- Begin with simple linear regression models to establish a baseline
- Experiment with different feature combinations and model parameters

Wednesday

Model Refinement

- Evaluate model performance using appropriate metrics (e.g., R^2, RMSE)
- Adjust our model by adding, removing, or transforming features as necessary

Validation

- Perform cross-validation to ensure your model's reliability and generalizability to unseen data

Thursday

Final Analysis

- Interpret the model results to draw meaningful conclusions
- Identify key predictors and their impact on the target variable

Presentation Drafting

- Start compiling our results, insights, and methodology into a coherent presentation
- Include visualisations that clearly communicate our findings

Finalisation Phase

- Finalise our presentation and rehearse for the final presentation

Friday

Deliver our Presentation

- Confidently present our project, articulating the problem, the approach, key findings, and the implications of our results. Engage with our audience during the Q&A session to address their questions and feedback

Checklist

- ☑ **~~Choose dataset~~**
  - ☑ ~~Verify dataset size, quality and relevance~~
  - ☑ ~~Ensure suitability for linear regression analysis~~
- ☑ **~~Build presentation deck to pitch concept to UX/UI peers~~**
- ☑ **~~Conduct data cleansing~~**
  - ☑ ~~Check for null values~~
    - ☑ ~~Handle null values~~
  - ☑ ~~Check for duplicates~~
    - ☑ ~~Handle duplicates~~
  - ☑ ~~Check formatting~~
    - ☑ ~~Handle formatting issues~~
  - ☑ ~~Format column names~~
  - ☑ ~~Check for missing values~~
    - ☑ ~~Handle missing values~~
  - ☑ ~~Export clean dataframe~~
- ☑ **~~Conduct EDA~~**
  - ☑ ~~Chart overview of distribution of variables~~
  - ☑ ~~Describe each variable~~
  - ☑ ~~Build correlation heatmap~~
    - ☑ ~~Compare correlation heat maps for red and white datasets~~
  - ☑ ~~Identify outliers~~
    - ☑ ~~Handle outliers~~
- ☑ **~~Conduct data preprocessing~~**
  - ☑ ~~Take decision on whether to merge our 2 datasets for analysis~~
  - ☑ ~~Take decision on which scaler/transformer to apply to our variables~~
    - ☑ ~~Apply different scalers to different variables~~
  - ☑ ~~Encode categorical variables (as appropriate)~~
- ☐ **Develop model**
  - ☑ ~~Build simple linear regression models to establish a baseline~~
  - ☑ ~~Perform cross-validation to ensure reliability of model~~

- ☑ ~~Evaluate model performance using appropriate metrics (e.g., R², RMSE)~~
- ☑ ~~Adjust the model by adding, removing, or transforming features as necessary~~
- ☑ ~~Using R2, check whether our model is overfitted or under fitted~~
- □ (Optional) Predict score of ‘new’ (as in, hitherto untested) wines
  - ☑ ~~Apply white wine data to model~~
    - □ If model has limited explanatory power on white wine data, build new model for white wine
- □ (Optional) Look for fresh data set to test model

☑ **~~Test hypotheses~~**

- ☑ ~~Write hypotheses~~
  - ☑ ~~Hypotheses per input variable or grouped? (Answered)~~
  - ☑ ~~Reject or fail to reject null hypothesis~~

□ **Conduct final analysis**

- ☑ ~~Interpret results to draw meaningful insights~~
- □ Identify key predictors of high-scoring wine
- ☑ ~~Document any other insights that emerge through our investigation~~

□ **(Optional) broaden the scope of the investigation**

- □ Can we bring in additional data to enhance our model?
- □ Can our input variables be converted into more commercial descriptors
- □ Can we apply our model to other drinks?

□ **Ensure thorough and coherent documentation**

- □ Write presentation and build compelling slide deck
- □ Ensure cleanliness of all artefacts:
  - □ Github repository
    - □ Google Doc of summary notes (ensure high readability)
    - □ Ipynb Notebooks for code (including consistent naming convention)
      - □ Separate Notebooks per step of analysis
    - □ CSVs (including consistent naming convention)
    - □ Slide decks (including consistent naming convention)
    - □ Tableau visuals

☑ **~~Enhance presentation through creation of interface for wine producers to predict score~~**

Meeting minutes

Key:

- Decision to be taken
- Decision taken

📅Monday 05.02.24

1. Separate analysis for red and wine datasets or one combined analysis?

Among our data portfolio, we had 2 datasets: one for red wine, one for white wine.

The dataset for red wine contained 1599 instances.
The dataset for white wine contained 4898 instances.

As a rough guide: our white wine data outweighs our red wine data by a ratio of approximately 3:1. Further research reinforces that white *vinho verde* is far more common than reds.

Based on further examination of the 2 datasets, we found that we were able to generate the model with the strongest explanatory power when using the red wine dataset.

As such we took the decision to separate datasets, and to focus our efforts on building a robust model for predicting red wine quality.

The decision to separate red and white wines for purposes of building a predictive model is further justified when considering the divergence of the mean averages of the input variables.

Characteristic	Mean red	Mean white
Fixed acidity	8.3	6.8
Volatile acidity	0.5	0.3
Citric Acid	0.3	0.3
Residual Sugar	2.5	5.9
Chlorides	0.1	0.0
Free sulfur dioxide	15.9	34.9
Total sulfur dioxide	46.8	137.2
Density	1.0	1.0
pH	3.3	3.2
Sulphates	0.7	0.5
Alcohol	10.4	10.6

2. Delete or retain duplicates?

There is no unique identifier in our datasets but there are some duplicate rows.

Red: 240 duplicates across 1599 rows of data (15%)
White: 937 duplicates across 4898 rows of data (19%)

Assumptions made in deciding how to handle duplicate rows:

- low likelihood that entirely duplicate rows could be coincidence
- after resolving duplication, we retain >1000 rows per wine colour

✅ Decision taken: drop duplicates

3. Which scalers should we apply to input variables?

General Considerations:

- **MinMaxScaler** is good when your data doesn't have outliers and you want to scale it to a specific range
- **StandardScaler** is a good default choice for many scenarios, especially when working with algorithms that assume a normal distribution
- **PowerTransformer** is useful when your data has a non-Gaussian distribution and you want to make it more Gaussian-like

Source: ChatGPT

We have taken the approach of exhaustively checking the efficiency of all XY scaler combinations, excluding combinations where no scaler would be applied to the X but a scaler would be applied to the Y. We will choose the combination that delivers the most powerful model.

Y ⬇️	X ➡️	**No scaler**	**MinMax**	**Power Transformer**	**Standard**	**Robust**
No scaler		Test	Test	Test	Test	Test
MinMax		No Test	Test	Test	Test	Test
Power Transformer		No Test	Test	Test	Test	Test
Standard		No Test	Test	Test	Test	Test
Robust		No Test	Test	Test	Test	Test

In order to speed this work up, we created the following linear regression definition ⬇️

```
def linear_regression(X_train, X_test, y_train, y_test):

    # Linear regression
    lm = LinearRegression()
    model = lm.fit(X_train, y_train)
    print(f'model coefficients:\n {model.coef_}\n')
    print(f'model intercept:\n {model.intercept_}\n')
    # Applying model to X test
    y_pred = model.predict(X_test)
    # Ensure y_test is in the correct format (pandas Series or 1D numpy array)
    if isinstance(y_test, pd.Series):
        y_test_reset = y_test.reset_index(drop=True)
    else:
        y_test_reset = y_test  # Assuming y_test is already a numpy array
    # Creating combined table with y_test and y_pred
    # Check if y_test_reset is a pandas Series and convert y_pred to a similar type
    if isinstance(y_test_reset, pd.Series):
        y_pred_series = pd.Series(y_pred, index=y_test_reset.index, name='y_pred')
        residuals_df = pd.concat([y_test_reset, y_pred_series], axis=1)
    else:
        # If inputs are numpy arrays, stack them horizontally
        residuals_df = np.column_stack((y_test_reset, y_pred))
        # Convert to DataFrame for easier manipulation later on
        residuals_df = pd.DataFrame(residuals_df, columns=["y_test", "y_pred"])
    # Calculating residuals
    residuals_df["residual"] = residuals_df["y_test"] - residuals_df["y_pred"]
    print(f'Residuals:\n {residuals_df}\n')
    # Root mean squared error
    rmse = mse(y_test_reset, residuals_df["y_pred"], squared=False)
    print(f'Root mean squared error: {rmse} \n')
    # R^2
    r2 = r2_score(y_test_reset, residuals_df["y_pred"])
    print(f'R2: {r2} \n')
    # Calculating adjusted R^2
    n = X_train.shape[0]  # Number of observations in the training set
    p = X_train.shape[1]  # Number of features used for training
    adjusted_r2 = 1 - (1 - r2) * (n - 1) / (n - p - 1)
    print(f'Adjusted R2: {adjusted_r2} \n')
    return model.coef_
```

Results of our scaler/transformer tests are documented here ⬇️

⚠️*Note: RMSE values may not be accurate due to an error in the function used to calculate the metrics*

Worked on	Features	Preprocessing X	Preprocessing y	RMSE	R²	Adjusted R²
JK	All except wine_type_red	None	None	0.723	0.315	0.313
MB	All except wine_type_red	Standard Scaler	None	0.723	0.315	0.313
MB	All except wine_type_red	Standard Scaler	Standard Scaler	0.820	0.315	0.313
MB	All except wine_type_red	Standard Scaler	Power Transform	0.823	0.312	0.309
MB	All except wine_type_red	Standard Scaler	MinMax Scaler	0.121	0.315	0.313
JK	All except wine_type_red	Power Transform	None	0.729	0.304	0.302
JK	All except wine_type_red	Power Transform	Standard Scaler	0.826	0.304	0.302

JK	All except wine_type_red	Power Transform	Power Transform	0.723	0.305	0.303
JK	All except wine_type_red	Power Transform	MinMax Scaler	0.121	0.304	0.302
MB	All except wine_type_red	MinMaxScaler	None	0.723	0.315	0.313
JK	All except wine_type_red	MinMaxScaler	Standard Scaler	0.820	0.315	0.313
JK	All except wine_type_red	MinMaxScaler	Power Transform	0.823	0.312	0.309
JK	All except wine_type_red	MinMaxScaler	MinMax Scaler	0.120	0.315	0.313
MB	All except wine_type_red	RobustScaler	None	0.723	0.315	0.313
MB	All except wine_type_red	RobustScaler	Standard Scaler	0.820	0.315	0.313
MB	All except wine_type_red	RobustScaler	Power Transform	0.823	0.312	0.309
MB	All except wine_type_red	RobustScaler	MinMax Scaler	0.121	0.315	0.313

✅ Decision taken: we opted for Standard Scaler on the X and no scaler on the y.

4. Can we link attributes of wine to more tangible descriptors?

Characteristic	Taste note
Fixed acidity	The wine's backbone - it contributes to the overall structure and firmness. Higher levels may make the wine taste more robust.
Volatile acidity	Think of it as the tanginess or sharpness in the wine. Too much can make it taste vinegary or unpleasant.
Citric Acid	Adds a refreshing, citrusy flavor to the wine. You might notice hints of lemon or lime.
Residual Sugar	This represents the sweetness in the wine. Wines with higher residual sugar will taste sweeter.
Chlorides	The saltiness or salinity in the wine. Too much might make it taste salty.
Free sulfur dioxide	Acts as a preservative. It helps to keep the wine fresh and prevents unwanted spoilage.

Total sulfur dioxide	The total amount of sulfur dioxide, which contributes to the wine's overall stability and longevity.
Density	Refers to the weight or thickness of the wine. Higher density can give the wine a richer, more full-bodied feel
pH	Measures the acidity level. Lower pH wines are more acidic, while higher pH wines are less acidic.
Sulphates	Acts as a preservative. Sulphates help to maintain the wine's freshness and prevent oxidation.
Alcohol	The alcohol content in the wine. Higher alcohol wines might feel warmer and more full-bodied.

5. **Which hypotheses would we like to test?**

✅ Decision taken:

H0 : β 1 = β 2 = β 3 = β 4 = β 5 = β 6 = β 7 = β 8 = β 9 = β 10 = β 11 = 0

Null hypothesis: The coefficients of all input variables are equal to 0.
None of our wine properties predict wine quality.

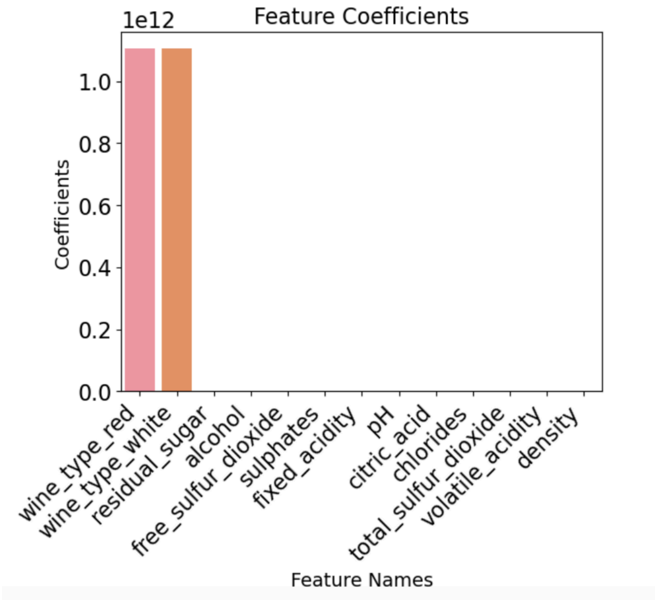
H1 : β 1 = β 2 = β 3 = β 4 = β 5 = β 6 = β 7 = β 8 = β 9 = β 10 = β 11 != 0

Significant linear relationship between at least one of the 11 input variables and the explained variable.
At least one of our wine properties predicts wine quality.

📅Tuesday 06.02.24

1. **How do we handle wine colour as an input variable?**

When we added wine colour as an input variable, we observed that it had an overwhelming and unhelpful effect on our feature coefficients, as shown below.



We believe that this happened for the following reason: linear regression models rely on independence of input variables. In the case of wine colour, red or white, the two variables are 100% dependent: if it is not red, it is white and vice versa.

✅ Decision taken:

As such, we decided that our options were either:

- drop wine colour entirely
- include just one wine colour in our variables

In this case we opted to include just one wine colour (red) as this enabled us to build a more powerful predictive model.

2. **How might we identify outliers with a view to improving our model's explanatory power?**

After our first iteration of linear regression, we observed that certain scaler combinations improved our RMSE but failed to improve our R2. Our hypothesis was that the presence of outliers was impacting our R2.

Decision taken:

We would chart a box plot for each input variable to map potential outliers.

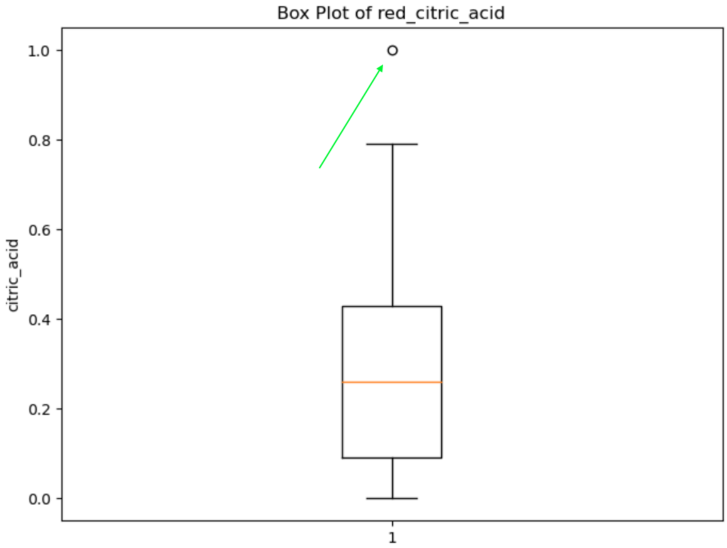
This analysis can be seen in Notebook: 'Wine Data Outlier Identification'

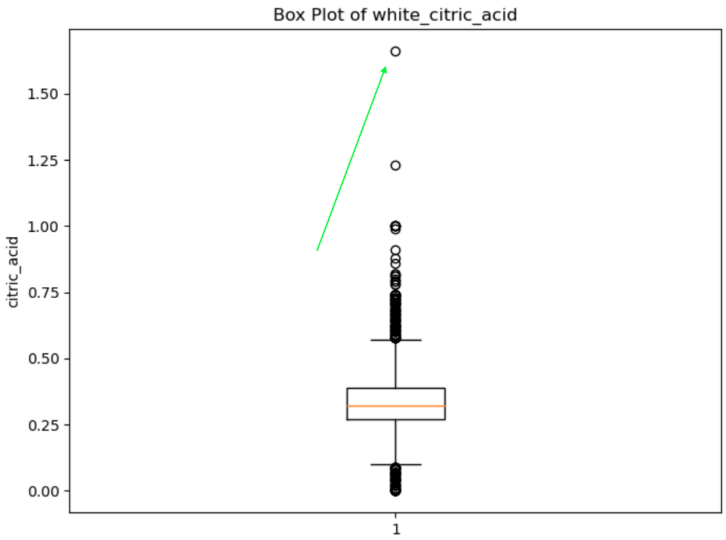
We have 11 input variables across 2 data frames i.e. 22 input variable sets.
Of the 22 input variable sets, outliers were detected (by naked eye) in the following categories – indicated by X.

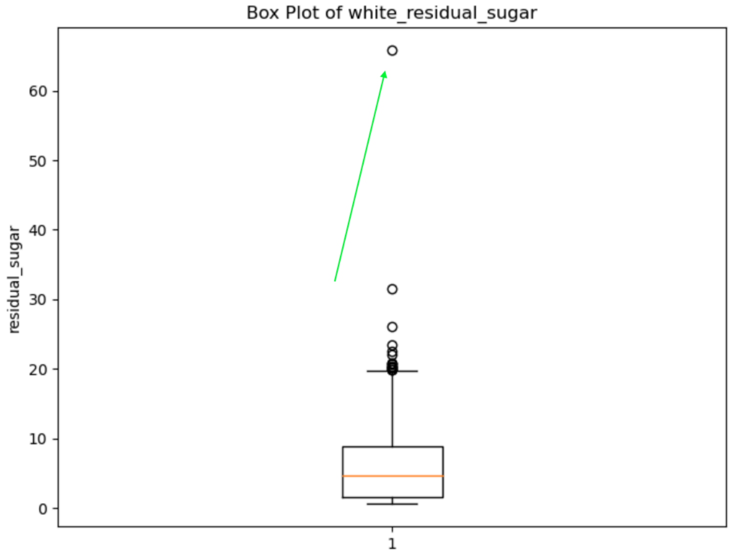
	Feature	Red_df	White_df
1	volatile_acidity		
2	fixed_acidity		
3	citric_acid	X	X
4	residual_sugar		X
5	chlorides	X	
6	free_sulfur_dioxide	X	X
7	total_sulfur_dioxide	X	X
8	density		X
9	pH	X	
10	sulphates	X	
11	alcohol	X	

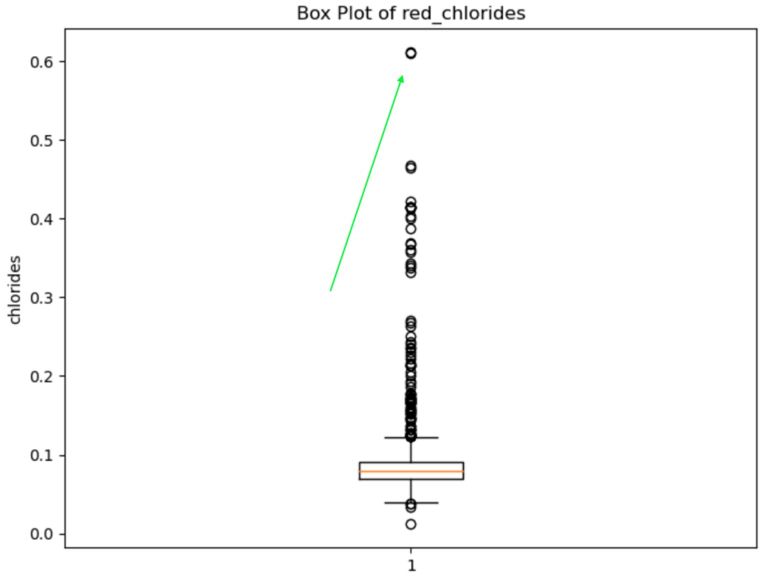
Features with outlier (identified by chatgpt)

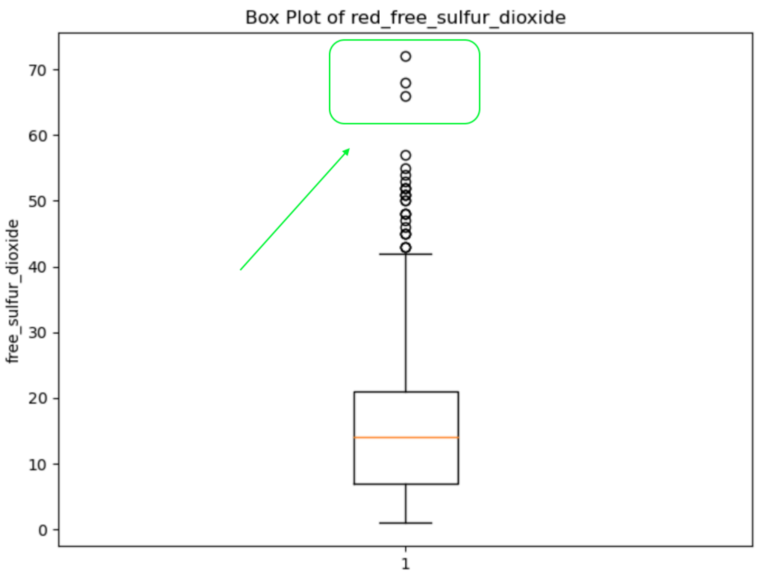
- Fixed_acidity,
- Volatile_acidity
- Citric_acid
- Residual_sugar
- Chlorides
- Free_sulfur_dioxide
- Total_sulfur_dioxide
- sulphates

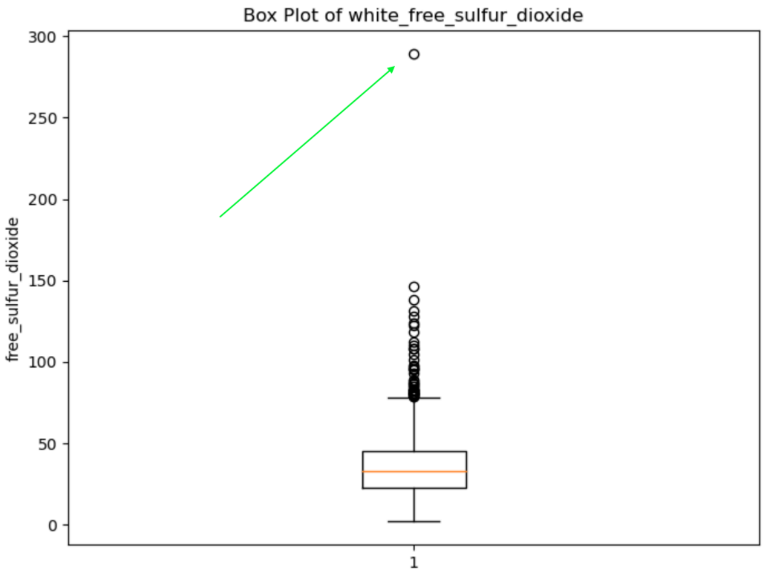


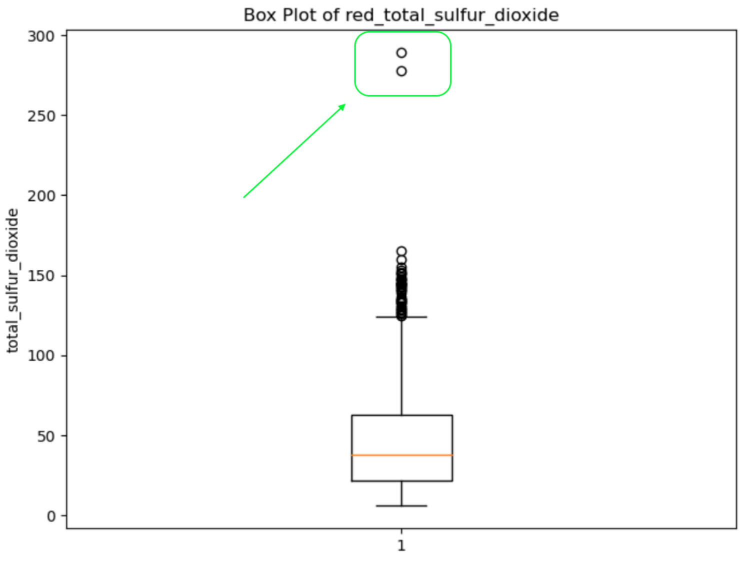

Box Plot of white_citric_acid
citric_acid
1.50
1.25
1.00
0.75
0.50
0.25
0.00
1

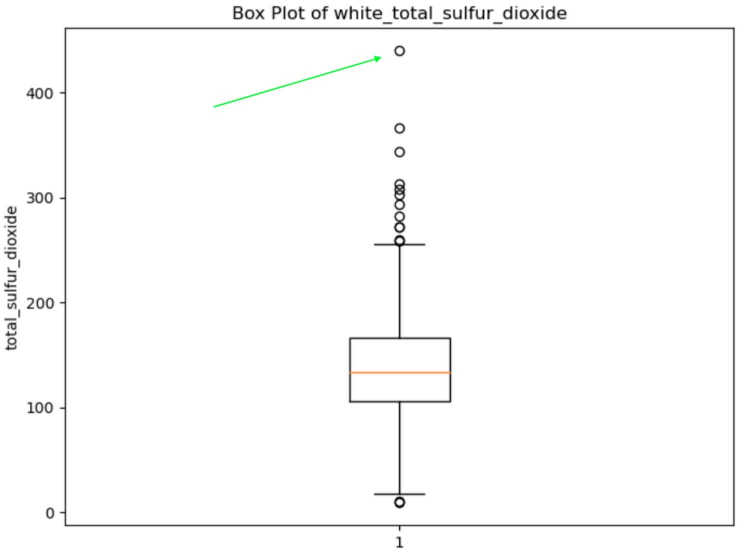

Box Plot of white_residual_sugar
residual_sugar
60
50
40
30
20
10
0
1

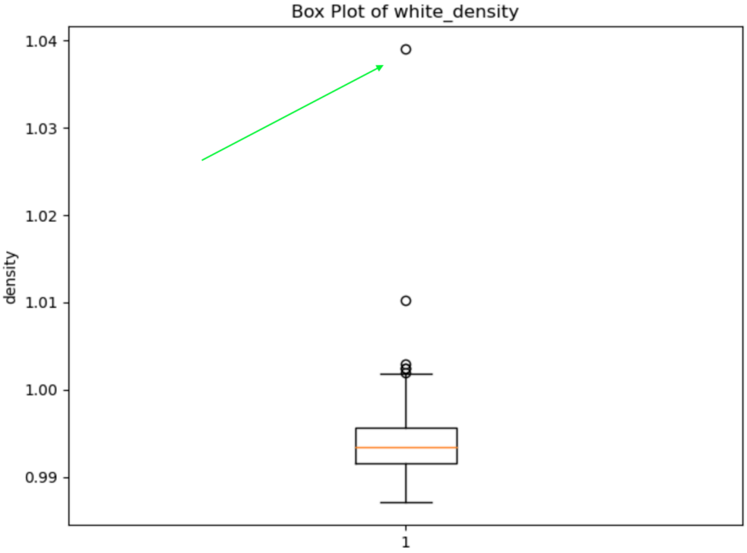

Box Plot of red_chlorides
chlorides
0.6
0.5
0.4
0.3
0.2
0.1
0.0
1

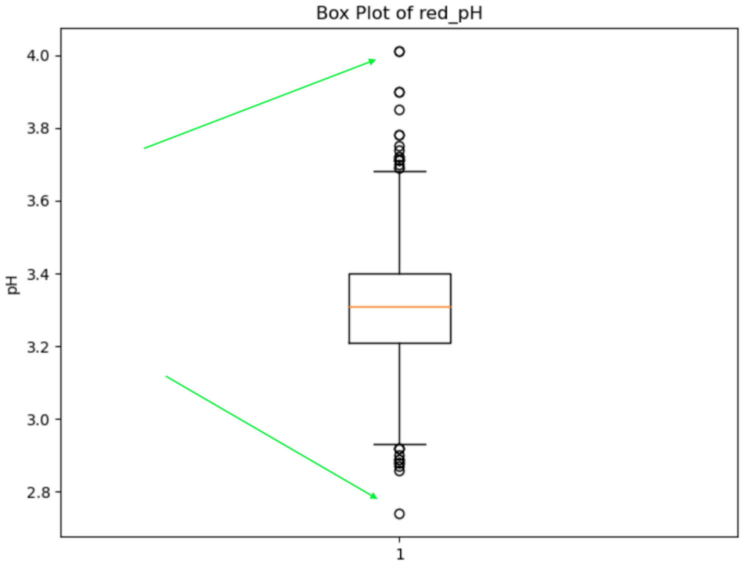

Box Plot of red_free_sulfur_dioxide
free_sulfur_dioxide
0
10
20
30
40
50
60
70
1

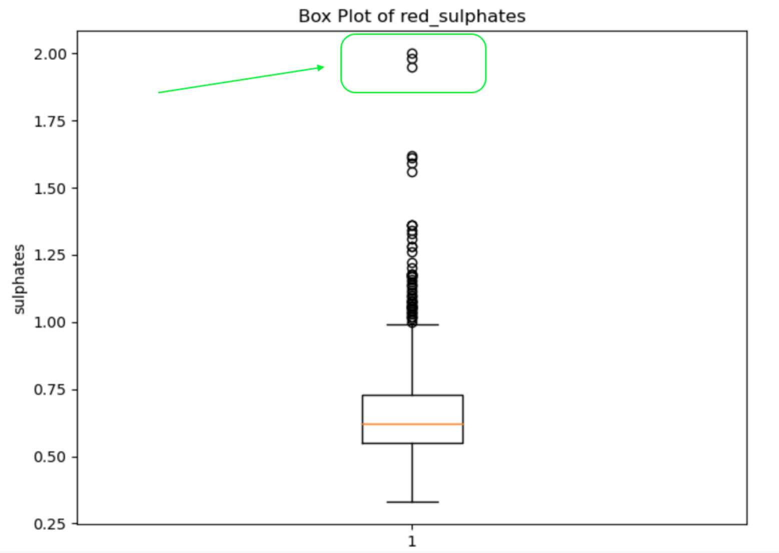

Box Plot of white_free_sulfur_dioxide
free_sulfur_dioxide
0
50
100
150
200
250
300
1

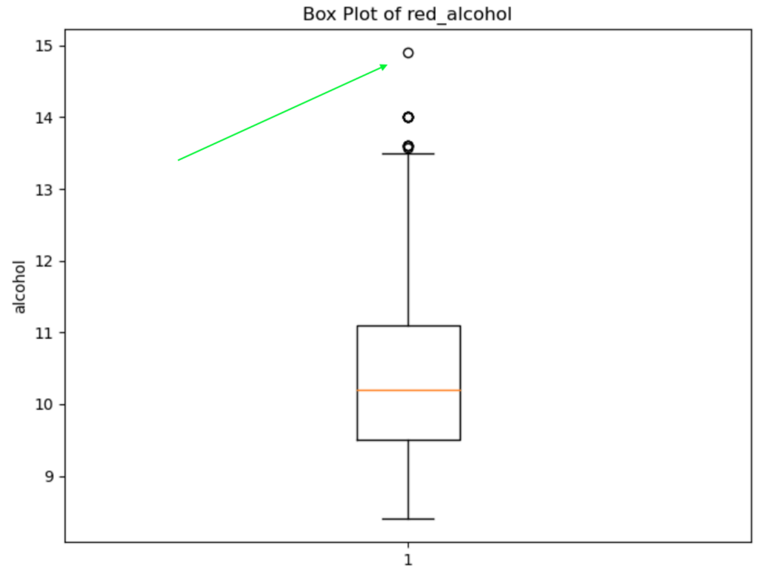

Box Plot of red_total_sulfur_dioxide
total_sulfur_dioxide
0
50
100
150
200
250
300
1


Box Plot of white_total_sulfur_dioxide
total_sulfur_dioxide
0
100
200
300
400
1


Box Plot of white_density
density
0.99
1.00
1.01
1.02
1.03
1.04
1


Box Plot of red_pH
pH
2.8
3.0
3.2
3.4
3.6
3.8
4.0
1





As we have outliers, we have the following options on how to handle them.

Options for how to handle outliers:

- **Do Nothing:** In some cases, outliers may represent genuine data points that are valid and meaningful. If you have reasons to believe that the outliers are not errors and are essential to your analysis, you may choose to leave them as they are.
- **Remove Outliers:** If the outliers are due to data entry errors or measurement errors, and they are likely to skew the analysis or modeling results, you may consider removing them from the dataset. This can be done by setting a threshold based on domain knowledge or using statistical methods like Z-score or IQR to identify and remove outliers.
- **Transform Data:** Transforming the data using mathematical functions (e.g., log transformation, square root transformation) can sometimes mitigate the impact of outliers on the analysis or modeling process. However, this approach may not always be appropriate, especially if the outliers are indicative of genuine phenomena in the data.
- **Winsorization:** Winsorization involves replacing extreme values with less extreme values. You can replace outliers with the nearest values within a certain percentile range (e.g., replacing values above the 95th percentile with the value at the 95th percentile).

- **Imputation:** If the outliers are due to missing data or measurement errors, you may impute them with a reasonable estimate based on neighboring data points or predictive modeling techniques.
- **Modeling Techniques:** Some machine learning algorithms are robust to outliers, while others are sensitive to them. Choosing appropriate modeling techniques that are less affected by outliers (e.g., robust regression, tree-based algorithms) can be an effective way to handle outliers.
- **Analyze Separately:** In some cases, it may be appropriate to analyze outliers separately to understand their impact on the analysis or modeling results. This can help identify patterns or insights that may not be apparent when analyzing the data as a whole.
- **Consult Domain Experts:** When in doubt, consult domain experts or stakeholders who have a deeper understanding of the data and its context. They can provide valuable insights into the nature of outliers and the best approach to handle them.

Source: ChatGPT

Having already applied Power Transformer to our model, we decided to test RobustScaler on our features to see whether this would improve our model.

RobustScaler

- RobustScaler scales features using statistics that are robust to outliers, such as the median and interquartile range (IQR).
- It subtracts the median from each feature and then scales it by the IQR. This makes it less sensitive to the presence of outliers compared to StandardScaler.
- RobustScaler is a good choice when you have features with outliers and want to preserve their relative differences.

Source: ChatGPT

✅Decision taken: In the end, we decided to apply IQR method to the variables identified above as consisting of outliers, capping thresholds at 0.75 and 0.25.

3. **Can we improve our model through removing any input variables?**

Using the red_df dataframe and Standard Scaler on X, MinMax scaler on Y, we ran our model deducting one input feature at a time.

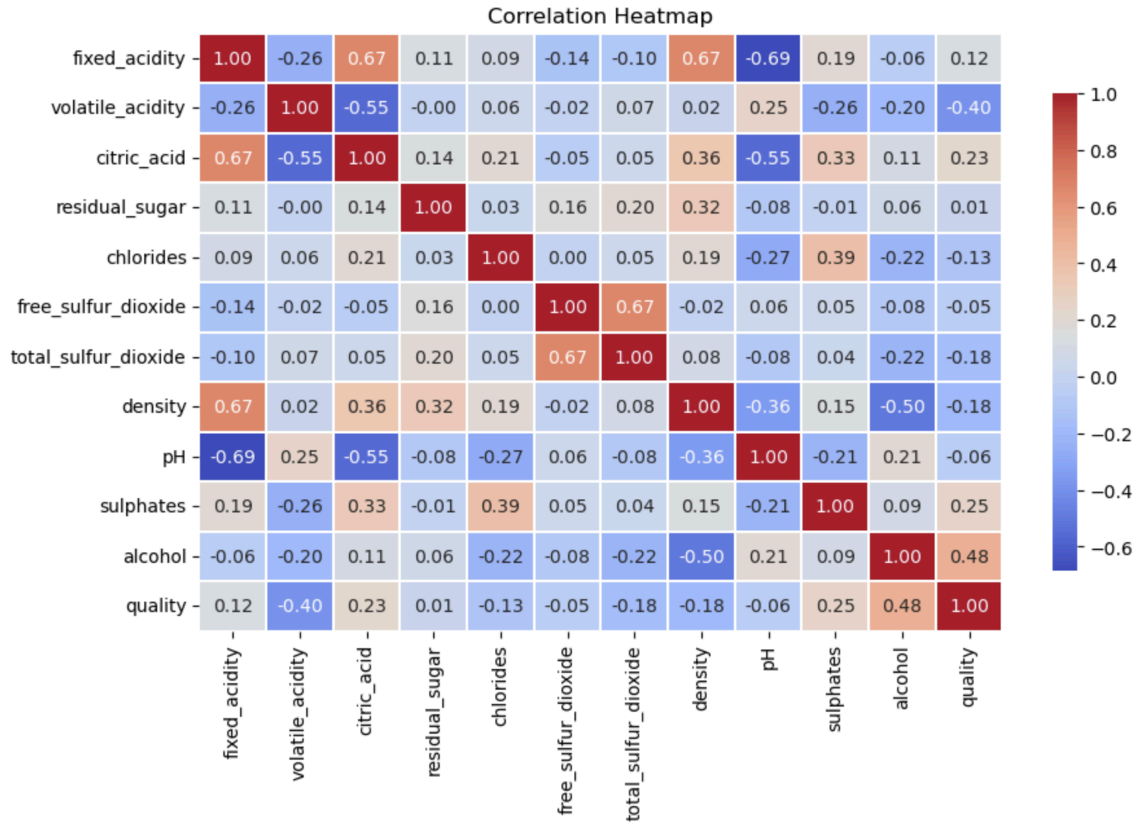
The results showed that we could get an ever so slight marginal gain by removing either free_sulfer_dioxide or density when modelling.

red_df				
Standard Scaler X, Min Max Scaler Y				**Ordered by ⬇️ DESC**
	Feature dropped	**RMSE**	**R2**	**R2 Adjusted**
	free_sulfur_dioxide	0.125	0.372	0.367
	density	0.125	0.372	0.367

	Nothing dropped (control group)	***0.125***	***0.372***	***0.366***
	fixed_acidity	0.125	0.372	0.366
	residual_sugar	0.125	0.371	0.366
	total_sulfur_dioxide	0.125	0.371	0.366
	citric_acid	0.125	0.371	0.365
	pH	0.125	0.369	0.364
	chlorides	0.126	0.363	0.358
	alcohol	0.128	0.34	0.334
	sulphates	0.128	0.334	0.323
	volatile_acidity	0.129	0.325	0.319
	Feature dropped	**RMSE**	**R2**	**R2 Adjusted**
	free_sulfur_dioxide, density	0.125	0.371	0.366

✅Decision taken: drop free_sulfur_dioxide

4. Correlation Study for red wine



	fixed_acidity	volatile_acidity	citric_acid	residual_sugar	chlorides	free_sulfur_dioxide	total_sulfur_dioxide	density	pH	sulphates	alcohol	quality
fixed_acidity	1.00	-0.26	0.67	0.11	0.09	-0.14	-0.10	0.67	-0.69	0.19	-0.06	0.12
volatile_acidity	-0.26	1.00	-0.55	-0.00	0.06	-0.02	0.07	0.02	0.25	-0.26	-0.20	-0.40
citric_acid	0.67	-0.55	1.00	0.14	0.21	-0.05	0.05	0.36	-0.55	0.33	0.11	0.23
residual_sugar	0.11	-0.00	0.14	1.00	0.03	0.16	0.20	0.32	-0.08	-0.01	0.06	0.01
chlorides	0.09	0.06	0.21	0.03	1.00	0.00	0.05	0.19	-0.27	0.39	-0.22	-0.13
free_sulfur_dioxide	-0.14	-0.02	-0.05	0.16	0.00	1.00	0.67	-0.02	0.06	0.05	-0.08	-0.05
total_sulfur_dioxide	-0.10	0.07	0.05	0.20	0.05	0.67	1.00	0.08	-0.08	0.04	-0.22	-0.18
density	0.67	0.02	0.36	0.32	0.19	-0.02	0.08	1.00	-0.36	0.15	-0.50	-0.18
pH	-0.69	0.25	-0.55	-0.08	-0.27	0.06	-0.08	-0.36	1.00	-0.21	0.21	-0.06
sulphates	0.19	-0.26	0.33	-0.01	0.39	0.05	0.04	0.15	-0.21	1.00	0.09	0.25
alcohol	-0.06	-0.20	0.11	0.06	-0.22	-0.08	-0.22	-0.50	0.21	0.09	1.00	0.48
quality	0.12	-0.40	0.23	0.01	-0.13	-0.05	-0.18	-0.18	-0.06	0.25	0.48	1.00

Observations

Highly Correlated variables (>60)

- fixed acidity - citric acid, density, pH
- citric acid - fixed acidity
- free sulfur dioxide - total sulfur dioxide
- total sulfur dioxide - free sulfur dioxide
- density - fixed acidity
- pH - fixed acidity

Strongly independent variables

- residual sugar
- chlorides
- sulphates
- Alcohol

Best guess at optimal features for model based on correlations:

- citric acid
- total sulfur dioxide
- density
- pH
- residual sugar
- chlorides
- sulphates
- alcohol

Key learnings

Data selection: when searching for datasets suitable for linear regression, much of the challenge was in finding a dataset that fulfilled the following requirements:

- >1000 rows
- > 10 columns
- not time series data
- independence of variables

Pair programming: where one line of code has dependency on earlier lines of code, it is challenging to split the workload

Model testing

Combined Dataset (red and white)

Note: RMSE values may not be accurate due to an error in the function used to calculate the metrics

Worked on	Features	Preprocessing X	Preprocessing y	RMSE	R^2	Adjusted R^2
JK	All except wine_type_red	None	None	0.723	0.315	0.313

MB	All except wine_type_red	Standard Scaler	None	0.723	0.315	0.313
MB	All except wine_type_red	Standard Scaler	Standard Scaler	0.820	0.315	0.313
MB	All except wine_type_red	Standard Scaler	Power Transform	0.823	0.312	0.309
MB	All except wine_type_red	Standard Scaler	MinMax Scaler	0.121	0.315	0.313
JK	All except wine_type_red	Power Transform	None	0.729	0.304	0.302
JK	All except wine_type_red	Power Transform	Standard Scaler	0.826	0.304	0.302
JK	All except wine_type_red	Power Transform	Power Transform	0.723	0.305	0.303
JK	All except wine_type_red	Power Transform	MinMax Scaler	0.121	0.304	0.302
MB	All except wine_type_red	MinMaxScaler	None	0.723	0.315	0.313
JK	All except wine_type_red	MinMaxScaler	Standard Scaler	0.820	0.315	0.313
JK	All except wine_type_red	MinMaxScaler	Power Transform	0.823	0.312	0.309
JK	All except wine_type_red	MinMaxScaler	MinMax Scaler	0.120	0.315	0.313
MB	All except wine_type_red	RobustScaler	None	0.723	0.315	0.313
MB	All except wine_type_red	RobustScaler	Standard Scaler	0.820	0.315	0.313
MB	All except wine_type_red	RobustScaler	Power Transform	0.823	0.312	0.309
MB	All except wine_type_red	RobustScaler	MinMax Scaler	0.121	0.315	0.313

White Wine Dataset

Worked on	Features	Preprocessing X	Preprocessing y	RMSE	R^2	Adjusted R^2
JK	All	Standard Scaler	MinMax Scaler	0.127	0.268	0.265
JK	All	Power Transform	MinMax Scaler	0.127	0.263	0.260

Red Wine Dataset

Worked on	Features	Preprocessing X	Preprocessing y	RMSE	R²	Adjusted R²
JK	All	Power Transform	MinMax Scaler	0.128	0.396	0.389
JK	All	Power Transform	Power Transform	0.637	0.400	0.393
JK	All	IQR on chat gpt rec columns, Power Transform	Power Transform	0.637	0.400	0.393
JK	All	IQR on matthew rec columns, Power Transform	Power Transform	0.637	0.400	0.393
JK	All	IQR on on chat gpt rec columnsl, Standard Scaler	MinMax Scaler	0.643	0.389	0.382

Outlier Removal Testing

Tested on red wine data set using IQR method

Worked on	Columns with IQR applied	Max Thresh	Min Thresh	Preprocessing X	Preprocessing y	RMSE	R²	Adjusted R²
JK	Combined	0.5	0.0	Power Transform	Power Transform	0.635	0.403	0.397
JK	Combined	0.6	0.0	Power Transform	Power Transform	0.636	0.402	0.395
JK	Combined	0.7	0.0	Power Transform	Power Transform	0.636	0.401	0.394
JK	Combined	0.7	0.01	Power Transform	Power Transform	0.636	0.402	0.396
JK	Combined	0.75	0.25	No Transform	No Transform	0.631	0.410	0.404
JK	Combined	0.75	0.25	Standard Scaler	No Transform	0.634	0.405	0.400
JK	Combined	0.75	0.25	Power Transform	None	0.630	0.413	0.406
JK	Matthew recommended	0.75	0.25	Power Transform	None	0.629	0.415	0.408
JK	Combined	0.75	0.25	Power Transform	Power Transform	0.634	0.406	0.399
JK	Combined	0.8	0.0	Power	Power	0.637	0.400	0.394

				Transform	Transform			
JK	All	0.9	0.1	Power Transform	Power Transform	0.637	0.400	0.393
JK	Chatgpt recommended	0.9	0.1	Power Transform	Power Transform	0.637	0.400	0.393
JK	Matthew recommended	0.9	0.1	Power Transform	Power Transform	0.637	0.400	0.393
JK	Chatgpt recommended	0.9	0.1	Standard Scaler	MinMax Scaler	0.643	0.389	0.382

Refinement of current best model

Worked on	Features	Columns with IQR applied	Max Thresh	Min Thresh	Prepro X	Prepro y	RMSE	R^2	Adjusted R^2
JK	All except free_sulfur_dioxide	Matthew recommended	0.75	0.25	SS	None	0.632	0.410	0.404

Output from pipeline

Worked on	Features	Columns with IQR applied	Max Thresh	Min Thresh	Prepro X	Prepro y	RMSE	R^2	Adjusted R^2
JK	All except free_sulfur_dioxide	Matthew recommended	0.75	0.25	SS	None	0.631	0.409	0.404

Checking for underfitting/overfitting

In order to check for underfitting and overfitting in our model we compared the r2 and adjusted r2 values from our test set with the r2 and adjusted r2 values calculated for the train set.

Test set values:
R2: 0.419

Adjusted R2: 0.413

Train set values:
R2: 0.360
Adjusted R2: 0.354

Conclusion:
The R2 values for our train set are relatively close to the R2 values for our test set. This indicates that the model is neither overfitted or under fitted

🗒️**Note**

In some cases, we observe a significant improvement in the RMSE (usually where we apply MinMax scaler to the y-axis), however we do not see a corresponding improvement in R2.

We sourced the following insight from ChatGPT:

‘Improvements in the Root Mean Squared Error (RMSE) do not always guarantee improvements in the coefficient of determination.

Here are a few reasons why your RMSE may improve while your R2 does not:

- **Influence of Outliers:** RMSE is sensitive to outliers, meaning that even small improvements in predicting outliers can result in significant improvements in RMSE. However, outliers might not have as much influence on R2,which measures the proportion of variance in the dependent variable that is predictable from the independent variables.
- **Non-linearity:** If the relationship between your independent variables and the dependent variable is non-linear, improving the model's ability to predict the dependent variable may not result in a higher R2 value. In such cases, the model may reduce RMSE by better fitting the data points around the mean, but it may not capture the non-linear patterns effectively.
- **Multicollinearity:** If there are high correlations between independent variables (multicollinearity), adding more predictors or improving the prediction of one predictor might not significantly improve the overall explanatory power of the model (reflected in R2).
- **Overfitting:** Improvements in the training RMSE might indicate overfitting to the training data, where the model performs well on the training data but generalises poorly to new, unseen data. In such cases, the model may not capture the underlying relationships in the data, leading to little improvement in R2.
- **Data Variability:** If the variability of the dependent variable is large relative to the variability of the independent variables, improvements in predicting the dependent variable may not lead to substantial changes in R2.

Presentation structure plan

Duration: 7 mins/8 slides = 53 seconds/slide

Key topics to cover:

1. **Intro (1 slide)**
    a. name
    b. subject matter
    c. mission - one-liner
2. **Description of data (1 slide)**
    a. .describe()
        i. null values
        ii. duplicates
        iii. number of rows, number of instances
    b. explanation of quality metric
    c. limitations of data

    d. decision to separate red and white

3. **Input variables (1 slide)**
    a. brief description
    b. correlation heatmap
    c. outliers
        i. approach taken to outliers
4. **Linear regression (2 slides)**
    a. show purpose-built function to perform linear regression
    b. initial explanatory power of model
    c. steps taken to improve model
        i. feature importance
    d. revised explanatory power of model
5. **Hypothesis testing (1 slide)**
    a. show H0 and H1
    b. show F-statistic, prob(F-statistic)
    c. interpret prob(F-statistic)
    d. reject the null hypothesis
6. **Evaluation of model (1 slide)**
    a. interpret RMSE, R2, adjusted R2 (underfit/overfit?)
    b. demonstrate how model performs better for ‘average’ wine:
        i. Distribution of quality data
        ii. Mean of residuals grouped by quality score
    c. limitations of model (e.g. not consumer friendly)
    d. white wine (TBD)
7. **Exhibit interface (1 slide)**
    a. show interface that a wine producer might use

Narrative thread:

- wine quality hard to predict; no clear correlation with any given property
- human element to wine quality (potentially bias in the rating system)
    - Output 0-10, hesitance to rate at extremes (see min, max)
- marginal gains are acceptable in this context (within 0.6 of a rating)

How do experts rate wine? Is there a standard method?

Final day workload

☑ ~~**JK: Wine quality predictor**~~
☑ ~~**MB: Build skeleton of presentation**~~
☑ ~~**MB: Difference red wine, white wine**~~
☑ ~~**JK: Check describe of each of the quality date of each wine colour**~~
☑ ~~**MB: Map alcohol x quality**~~
☑ ~~**MB: Look up valid ranges for each variable**~~
☐ **Optional: Look up how experts taste wine**
☑ ~~**MB: Check whether we can do live tasting**~~
☐ **Together: Code review**