# DIAML: Assignment 6

by Anthony Rizkallah

## Table of contents

**Python Packages**

Pandas

Matplotlib

Numpy

Sklearn.linear_model

Sklearn.metrics

Sklearn.tree

Sklearn.model_selection

# 1 Nonlinearity

## 1.1 Definition

A non-linear model involves two variables whose relationship lacks linearity (or a straightforward relationship) and is expressed in an equation where its terms are not to the first degree. They are important models for various reasons– three of them: They can *capture complex relationships* in data that linear models can't capture. By the same token, non-linear models *offer flexibility* and can adapt to a variety of patterns in data. Finally, data can sometimes seem linear over sub-sections, but when looking on the wholistic trend, the relationship would be *more accurately described* as a non-linear equation.

## 1.2 Mathematical Equation

$$y(x) = 1 - \frac{x^2}{2!} + \frac{x^4}{4!}$$

This equation is a simplistic model of the sun's irradiance at a certain point on earth. This equation is a taylor series expansion of $\cos(x)$ which is a simple model for the sunlight at a certain point on earth– the solar irradiance equation is $S = S_0 \cos(\theta)$. Approximating $\theta$ for small angles does not fully encapsulate the real relationship, but it almost follows the same path.

The application of this equation can be used by a solar farm to forecast how much energy it is expected to generate over a day where the sunlight is at its full capacity.

## 1.3 Parsimony of nonlinear and linear models

When it comes to parsimony, linear models are the winners over non-linear ones because they assume straightforward relationships. On the other hand, non-linear models are less parsimonious because they favor complexity over flexibility to find complex patterns in data where relationships are not as straightforward, or if the goal is to achieve more precision in the short term rather than find the general trend over elongated periods. For example, if the general trend is growth, one can use linear regression to model the trend, but if the goal of the model is to find short-term patterns, then non-linear models should be used, sacrificing interpretability for flexibility.

Linear:

$$y_{\text{linear}} = \beta_0 + x_1\beta_1 + x_2\beta_2 + \cdots + x_n\beta_n + \epsilon$$

Non-linear:

$$\begin{aligned} y_{\text{non-linear}} &= f(x_1, x_2, x_3, \ldots x_n) + \epsilon \\ &\phantom{=} \text{Examples} \\ y_{\text{non-linear}_1} &= \beta_0 + x_1\beta_1 + x_2^2\beta_2 + \cdots + x_n^n\beta_n + \epsilon \\ y_{\text{non-linear}_2} &= \beta_0 + x_1\beta_1 + x_1x_2^2\beta_2 + \cdots + x_1x_n\beta_n + \epsilon \end{aligned}$$

where $\beta_0$ is the y-intercept, $x_n$ is the predictor, and $\beta_n$ is the weight assigned to each predictor, and $\epsilon$ is the residual or error term.

## 1.4 Characteristics of surrogates

### 1.4.1 Amplitude Distribution and Linear Correlations

The preserved characteristics in surrogates are *Amplitude Distribution* and *Linear Correlations*. Amplitude distribution keeps the same distribution of values as the original time series, ensuring that the range and spread of the values remain consistent. Additionally, linear correlations preserve the linear correlation structure of the original data, so that time-dependent relationships are maintained, and non-linear structures are removed.

### 1.4.2 Two Surrogate Techniques

**Random Shuffle**: Randomly shuffles original data points and destroys linear relationships, but preserves the amplitude distribution.

- → Approach: Randomly shuffle the original data points
- → Usage: Used as a test to eliminate time-related dependencies

Random Shuffle is a technique that is used to test for linear effects. By randomizing the order of data points, it removes both linear and non-linear dependencies, providing a baseline to test whether observed relationships are due to structure in the sequence or simply the distribution of values.

**Random Phases (Fourier Transform)**:

- → Approach: Take the fourier transform of the original data, randomize the phases of the Fourier coefficients, and then apply the inverse Fourier transform
- → Usage: Used when linear relationships and distributions must be preserved

Random Phases is a technique that is used to isolate non-linear effects. It helps in determining whether the non-linearity in the data is caused by random noise or whether there is an inherent non-linear relationship within the data.

## 1.5 Entropy, Information, and Mutual Information

### 1.5.1 Entropy

Entropy is the measurement of the average uncertainty in the value of the discrete-valued probability density. In other words, entropy measures the randomness, disorder, or expected surprise.

$$H(x) = -\sum_{i=1}^{n} p(x_i)\log(p(x_i))$$

where $H(x)$ is the entropy and $p(x_i)$ is the probability of the outcome of variable $x_i$. Higher entropy means higher unpredictability and vice versa.

Important to note is that the entropy formula in this case is not bounded between 0 and 1 and might be difficult to normalize it. If the goal is to normalize entropy so that it can be standardized in a certain analysis, then

$$H_{\text{norm}} = \frac{H(x)}{H_{\text{max}}}$$

where $H_{\max} = \log(n)$. In other words, maximum entropy is achieved by assuming that all outcomes are equally probable with a probability of $\frac{1}{n}$.

### 1.5.2 Information

Information is also a measurement of surprise or unpredictability, but instead of measuring the unpredictability of a dataset, it measures the uncertainty of a single variable.

$$I(p(x)) = \log\left(\frac{1}{p(x)}\right) = -\log(p(x))$$[1]

where $p(x)$ is the probability of getting variable $x$ in the dataset, $I(p(x)) \geq 0$ –denoting that information is a non-negative quantity, and $I(1) = 0$ –denoting that recurring events have no uncertainty. Moreover, information of independent events is additive:

$$I(p(x_1)p(x_2)) = I(p(x_1)) + I(p(x_2))$$

Interestingly, entropy can be rewritten as a function of information as follows

$$H(x) = \sum_{i=1}^{n} p(x_i) I(p(x_i))$$

This represents how Entropy and Information are related. In fact, the former represents the average uncertainty in the dataset and the latter represents the uncertainty of one variable in the dataset.

### 1.5.3 Mutual Information

Mutual Information measures the amount of information that two random variables share. In other words, it quantifies how much knowing one of these variables shares about the other.

$$I(x, y) = \sum_{x} \sum_{y} p(x, y) \log\left(\frac{p(x, y)}{p(x)p(y)}\right)$$

where $p(x, y)$ is the joint probability of $x$ and $y$, and $p(x)$ and $p(y)$ are the marginal (independent) probabilites of $x$ and $y$.

If $p(x, y) = p(x)p(y)$, then the probabilities of the two events are independent and the mutual information $I(x, y)$ would be 0 because $\log(1) = 0$, representing no relationship between the two.

On the other hand, if $p(x, y) \neq p(x)p(y)$, then there is some relationship that exists between the two variables such that the higher $I(x, y)$, the stronger the relationship.

### 1.5.4 Using Entropy to Measure Regularity

Entropy measures the uncertainty or disorder in a dataset. Low entropy indicates high regularity (more predictable patterns), while high entropy suggests low regularity (more randomness). This can be particularly useful in time series analysis, natural language processing, or other domains where pattern detection is important.

For example, in health monitoring, entropy can be used to analyze heart rate variability. A regular heart rhythm would have a moderate level of entropy which would show natural fluctuation. On the other hand, very low entropy may indicate abnormal regularity and high entropy could suggest irregularities. By calculating the entropy of the time series, a feature that represents the regularity of a person's heart rhythms can be created, which could be used for diagnostic purposes or to monitor patient health.

1. DIAML Lecture Slides

### 1.5.5 Using Mutual Information in Feature Selection

As mentioned in *Section 1.5.3*, mututal information can be used for feature selection because it quantifies the relationship between a feature and the dependent variable– the higher the $I(x, y)$, the stronger the relationship. Mututal Information could be better compared to linear correlation, becuase it can capture both linear and non-linear relationships, whereas linear correlation might miss the more complex relationships that cannot be straightforwardly expressed.

# 2 Classification using trees

## 2.1 Decision trees

### 2.1.1 Components

- Nodes: That's where decisions are made. Each node uses a *feature* of the data to split it into different branches
- Branches: Represent the different outcomes of each decision made at each node
- Leaves: Represent the final decisions or classifications

### 2.1.2 Pruning

It is the process of removing sections of the tree that provide little predictive power. This is beneficial because it helps simplify the model and avoiding overfitting. It's necessary because trees that are too complex or have many branches can overfit the training data and may not generalize well to new data.

### 2.1.3 Attractiveness

Decision trees are often chosen for classification tasks because they are easy to interpret, can handle both numerical and categorical data, and work well with large datasets. They also provide a **clear path** for decision-making. This makes them useful for applications where explainability and the logical steps are important.

## 2.2 Improving decision trees

### 2.2.1 Steps to building decision trees

1. **Data Collection:** Collect a large and representative dataset for training and testing.
2. **Feature Selection:** Identify relevant features that might enhance the model's performance.
3. **Data Preprocessing:** Clean the data, handle missing values, and encode categorical variables.
4. **Model Selection:** Choose a suitable data-driven model (e.g., Random Forest).
5. **Model Training:** Train the selected model on the training dataset.
6. **Parameter Tuning:** Optimize the model by tuning parameters.
7. **Model Evaluation:** Assess the model's performance on a separate test dataset.

### 2.2.2 Evaluating decision trees

1. **Cross-Validate Model:** Use k-fold cross-validation to ensure the model's generalization across different subsets of the data.[2]

2. https://community.st.com/t5/mems-and-sensors/how-to-evaluate-the-performance-of-a-decision-tree/ta-p/49612

2. **Use Performance Metrics:** Evaluate the model using appropriate metrics such as accuracy, precision, recall, and F1-score.

3. **Compare with Rule-based Classifier:** Compare the performance of the data-driven model with the existing rule-based classifier.

4. **A/B Testing:** Deploy both models simultaneously and measure their effectiveness in real-world scenarios.

## 2.3 Titanic decision tree

### 2.3.1 Steps

To build the Titanic decision tree, the dataset must first be prepared. First, the features that were selected for the model were "Sex", "Age", and "Passenger Class". In the "Age" column, there are 263 NaN values that were imputed by the average age in the dataset, and no other NaN values. Then, transform categorical values to numerical in the dataset. Now, the dataset is ready to be used to build the model.

Then, to evaluate model performance, calculating the misclassification errors in the dataset:

$$\text{MCE} = 1 - \text{Accuracy Score}$$

where MCE is misclassification error.

The cross-validation average across all different k-fold iterations is another model evaluation technique that trains the model on different samples of the training dataset.

## 2.4 Performance of Titanic decision tree

### 2.4.1 Before Pruning

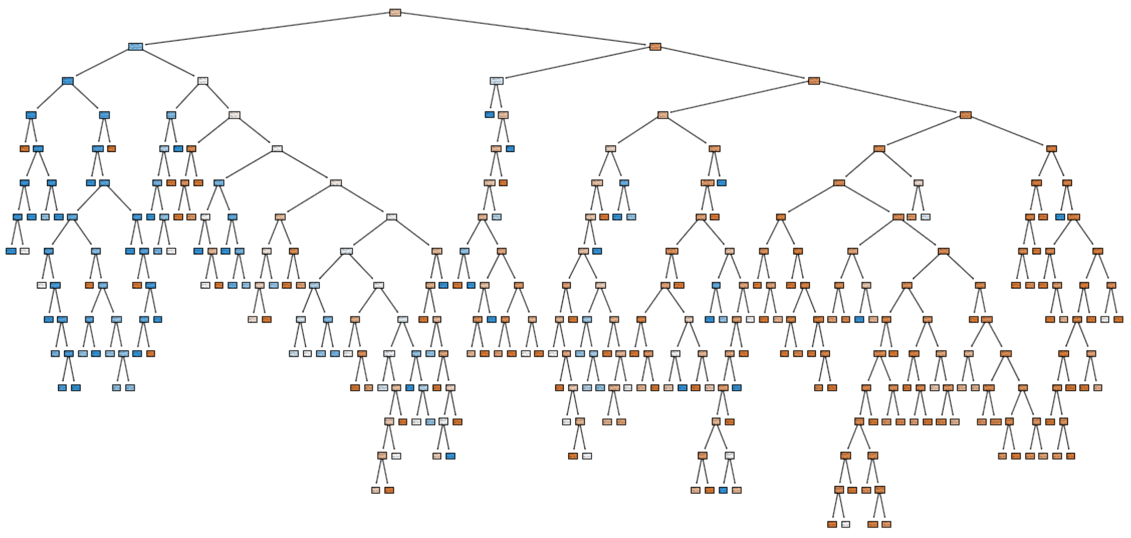

**Figure 1.** Decision Tree Before Pruning

Misclassification Error: 20.23%[3]

3. https://machinelearningmastery.com/repeated-k-fold-cross-validation-with-python/

In-Sample Cross-Validation Accuracy Average: 76%

### 2.4.2 After Pruning

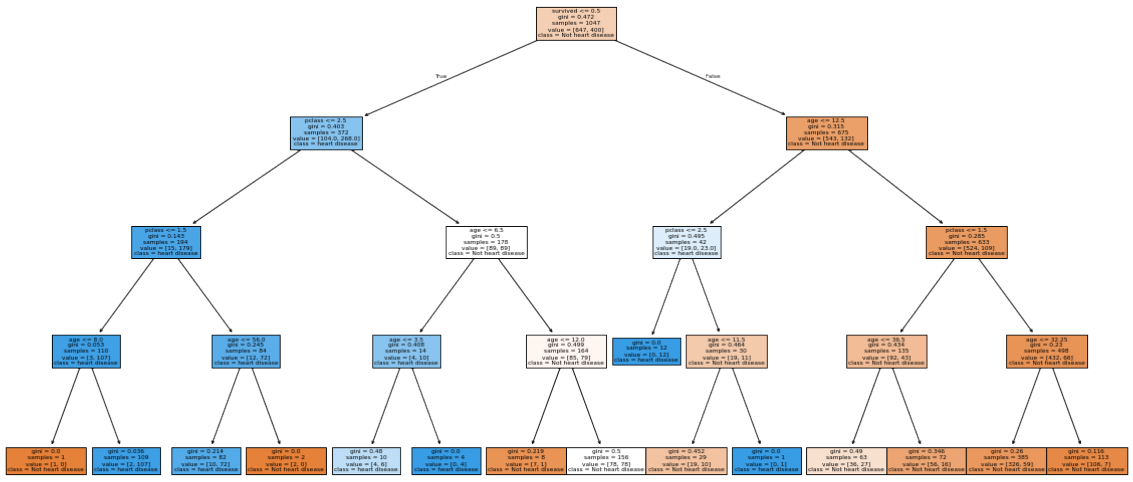

**Figure 2.** Decision Tree After Pruning

Misclassification Error: 17.56%[4]

In-Sample Cross-Validation Accuracy Average: 77.2%

### 2.4.3 Inference

After pruning the model, there was approx. a 3% improvement in misclassification compared to before pruning, and that's because the model is able to generalize better when the branches and nodes are limited to the most optimal model.

## 2.5 Logistic regression vs decision tree on Titanic dataset

### 2.5.1 Logistic Regression Setup

The logistic regression model is one that is used to determine the probability of a certain outcome, where in this case it is either survival or death. The logistic sigmoid function, also called the squashing function, maps out the whole real axis [-inf, +inf] into a finite interval [0, 1] and it is given by

$$y(x) = \frac{1}{1 + e^{-x}}$$

where $x$ is the dependent variable.

To train the logistic regression model, the goal is to use the sex, age, and pclass as the independent variables to predict the probability of survival on the titanic.

4. https://www.kaggle.com/code/arunmohan003/pruning-decision-trees-tutorial?scriptVersionId=66335932

After building the model, it is appropriate and necessary to evaluate whether the variables are statistically significant and have a p-value $< 0.05$. Since this was previously analyzed and all three variables “Sex”, “Age” and “Passenger Class” are all statistically significant.

### 2.5.2 Logistic Regression Model

Misclassification error: 18.32%

In-Sample Cross-Validation Accuracy Average: 77.6%

### 2.5.3 Best Model

The best model out of all– Decision Tree without pruning, Decision Tree with pruning, and Logistic Regression– is Logistic Regression. First, from an accuracy perspective, after cross-validating all models, the logistic regression model is slightly more accurate at 77.6% compared with the pruned decision tree at 77.2%. Second, the logistic regression model is more parsimonious, so from a simplicity point of view, the logistic regression is the better performer on both evaluative metrics. However, the decision tree had less misclassifications in training and prediction, but that is due to the nature of the model, where it adds more layers to the decision-making process, resulting in better classifications. The disadvantage is that the logistic regression model was better at generalizing the trend in the data, hence the higher cross-validation accuracy. All in all, the logistic regression model is better to use in the Kaggle competition as it is simple, more accurate, flexible, and explainable.

# 3 Classification using KNN

## 3.1 Concept and Approach

### 3.1.1 Concept

The concept of focusing on small neighborhoods of a state space is a concept that only considers the local information around each data point. Assuming that the relationships between information around each data point are locally simple, then the information can be considered. The approach is relevant when dealing with complex, non-linear relationships that can vary throughout the feature space. In other words, what is true at a certain point, might not be true at a different point. So, by assigning a location (say (1,1)) and limitting how far the relationship is true, meaning if the model were to evaluate the relationship at (5,5), it will use a different set of assumptions to make the prediction. This is generally ideal for non-linear relationships where information does not flow in a straightforward manner, providing local similarity, complexity management in regions where general trends fail to capture important relationships, and flexibility.

### 3.1.2 Steps for Implementation

1. **Data Collection:** Collect the data, impute missing values, and encode categorical variables

2. **Feature Selection:** Select relevant independent variables

3. **Scale features:** Since this is a distance-based method, the features must be scaled (standard scaling: subtracting the mean and dividing by the standard deviation)

4. **Choose neighborhood size:** Set the size of the neighborhood (k) which defines the distance at which each relationship that the model finds will be applicable

5. **Build Model**: Store the training dataset and use it to predict output for new data points by voting based on the nearest k training samples

6. **Cross-validate model:** Use k-fold cross validation to assess the model's performance by diving the dataset into subsets, testing the model on different pieces of the information

7. **Adjust neighboorhood size:** This is an important step because when validating the model, different neighboorhood sizes must be tested to ensure the right balance between underfitting and overfitting. This can be done by using performance metrics such as MSE and accuracy

8. **Evaluate the model:** After finding the optimal parameters, evaluate the final model on the test set using metric such as accuracy, precision, recall, F1-score for classification, and MSE or RMSE for regression tasks.

## 3.2 Transforming Variables in Titanic Dataset

In the Titanic dataset, the variables included in the model are categorical except for "Age". "Sex" and "Pclass" are categorical, but are not ecnoded. This can pose an issue because KNN models need to locate the position of "Age" on the relative scale of the feature data, and for categorical, the data must be encoded to ensure the handling of scalar values. Hence, "Age" must be scaled using standard scaling:

$$X_{\text{scaled}} = \frac{X - \mu}{\sigma}$$

where $X$ is the original feature, $\mu$ is the mean, and $\sigma$ is the standard deviation.

For categorical non-scalar values, they could either be mapped manually ({Male: 1, Female: 0}) or automatically using an encoder.

## 3.3 Performance

### 3.3.1 Steps

To evaluate the performance of the model, there are two metrics to account for: In-Sample Accuracy, how well the model performed in the training set and then the cross-validation accuracy, where the model cross-validates across subsets of the data. Then, there are certain distance metrics that evaluate different performances of the models. For example, Euclidean, Chebyshev, and Manhattan.

Euclidean is the default measurement for many applications because it calculates the straight-line distance in a Euclidean space.

$$d = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$$

Manhattan, also know as the L1 norm, is the sum of absolute differences between points across all dimensions. Meaning, similar to vector addition of $\overrightarrow{\mathrm{AB}} + \overrightarrow{\mathrm{BC}} = \overrightarrow{\mathrm{AC}}$, but instead of calculating the norm of $\overrightarrow{\mathrm{AC}}$, manhattan is a method that calculates the $x$ and $y$ coordinates. It is more benefitial in calculating outliers because it takes into account the underlying path, rather than the generalized vector.

Chebyshev is the maximum difference between two vectors along any coordinate dimension. So, if there are two distances in a 2-D grid, $(x_1, y_1)$ and $(x_2, y_2)$, the metric calculates $\max(y_2 - y_1, x_2 - x_1)$. In other words, it finds the worst-case scenario in the data-point. This can be useful in applications where the worst-case scenario is the most important metric such as Chess.

### 3.3.2 Results

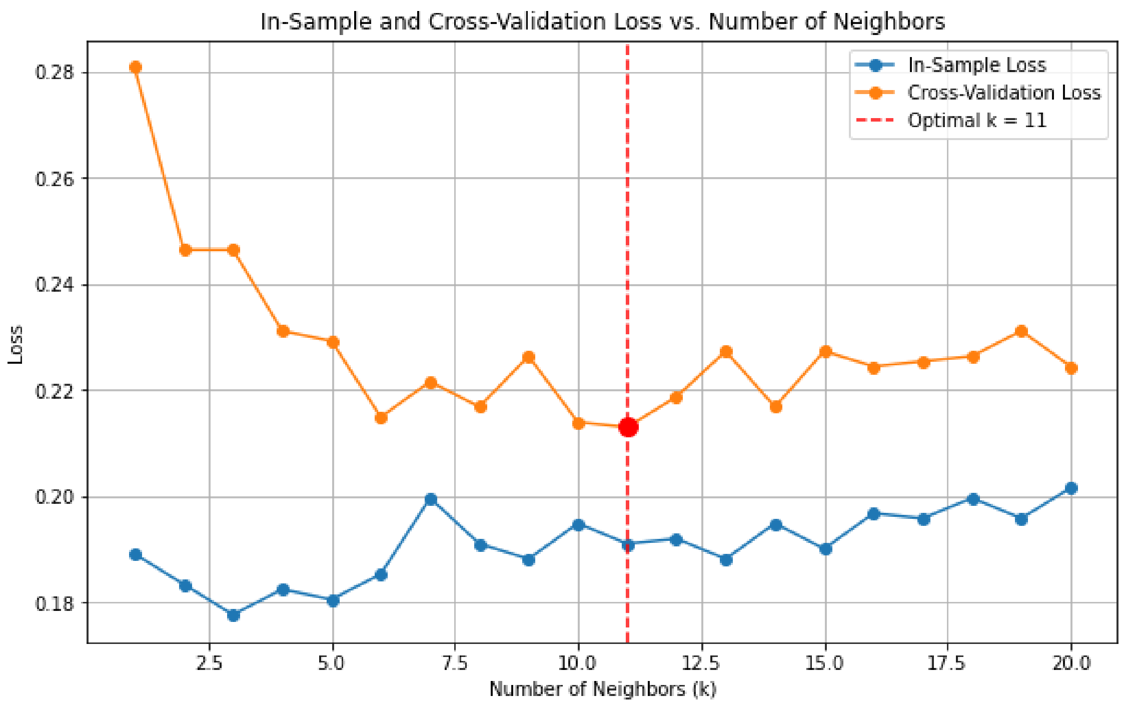

**Figure 3.** In-Sample and Cross-Validation Loss vs. Number of Neighbors

Optimal Number of Neighbors (k): 11

Minimum Cross-Validation Loss: 21.30%

Cross-Validation Accuracy Mean: 78.8%

Cross-Validation Accuracy for Chebyshev: 76.89%

Cross-Validation Accuracy for Euclidean: 77.08%

Cross-Validation Accuracy for Chebyshev: 77.07%

### 3.3.3 Inference

1. The graph plots both in-sample loss and cross-validation loss against the number of neighbors. Small k leading to lower bias but potentially higher variance and generalization, while larger k potentially increasing bias but lowering variance.
2. The red dashed line indicates that the optimal number of neighbors, k=11, has been chosen based on the cross-validation loss. This suggests that at k=11, the model achieves a balance between bias and variance, offering the best generalization performance on unseen data.
3. The model outperformed Decision Tree and Logistic Regression when measuring the average cross-validation accuracy and using the Chebyshev accuracy. However, to compare apples-to-apples, only comparing the average cross-validation accuracy can be taken into account.

## 3.4 Sensitivity of Distance Metrics

- Scale: Distance calculations can be disproportionately affected by features with large numerical ranges. For example, in Euclidean distance metric, a feature with a large scale can dominate the distance calculation, overpowering features with smaller ranges. This affects the bias of the model because it weighs in the dominant feature.
- Outliers: Euclidean distance is very sensitive to outliers since the squaring of the differences can magnify the extremes. The can skew the distance calculations, making the model overly sensitive.

### 3.5 KNN vs Logistic Regression

The Logistic Regression model has an accuracy of 77.6% when avergaing the cross-validated samples, whereas KNN has an accuracy of 78.8%. So, when choosing which model should be used for the Kaggle competition, there are multiple factors to be considered. If accuracy performance is a metric, then KNN is about 1% better in predicting. Moreover, having k = 11 is not very high since it was cross-validated across multiple datasets, which means that the model is not over or underfitting. However, if computational efficiency and interpretability are important factors for the Kaggle competition, then choosing Logistic Regression would be the better choice.

All in all, for the Kaggle competition, the higher the accuracy of the model, the better, so KNN should be the more optimal option even though it is slightly more accurate than the logistic regression model.

## 4 Regression - Wine Quality

### 4.1 Features and Meanings to Wine Experts

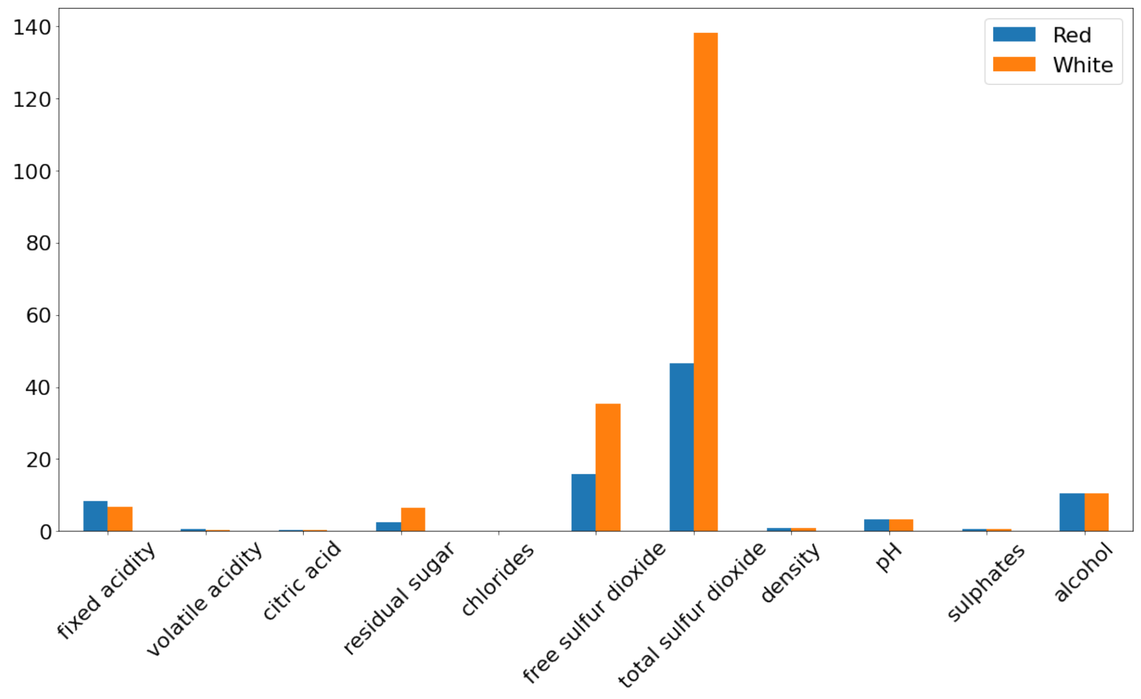

**Figure 4.** Average of Features for Both Red and White Wines

To compare the difference in chemical composition of both types of wine and how the intuition of wine experts might relate to this composition, it is worth considering the features that are different in both: Fixed Acidity, Residual Sugar, Free Sulfur Dioxide, and Total Sulfur Dioxide.Acidity contributes to the taste, stability, and balance of wine, with a focus on how acidity levels affect the sensory experience, making it either tart and zesty or flat and lifeless[5]. Residual sugar for experts is a determination of how sweet the wine is. However, different wines have different levels of sweetness and is not related to their quality[6]. Sulfur Dioxide preserves the freshness and prevents oxidation and microbial growth in wine, and it is crucial in maintaining wine quality and does not affect taste[7].

5. https://winefolly.com/deep-dive/understanding-acidity-in-wine/
6. https://www.decanter.com/learn/residual-sugar-46007/
7. https://www.cdpr.ca.gov/docs/dept/factshts/so2.pdf

## 4.2 Feature's Statistical Significance

### 4.2.1 Red Wine

| **Feature** | **Correlation** |
|---|---|
| Fixed Acidity | 0.124 |
| Volatile Acidity | -0.39 |
| Citric Acid | 0.226 |
| Residual Sugar | 0.013 |
| Chlorides | -0.129 |
| Free Sulfur Dioxide | -0.051 |
| Total Sulfur Dioxide | -0.185 |
| Density | -0.175 |
| pH | -0.058 |
| Sulphates | 0.251 |
| Alcohol | 0.476 |

**Table 1.** Feature correlation with Red Wine quality

The most relevant feature is Alcohol with a correlation of 0.476. An argument can also be made for lower correlation features such as Sulphates (0.251), Volatile Acidity (-0.39), and Citric Acid (0.226), but the correlation is significantly in the low correlation band.

### 4.2.2 White Wine

| **Feature** | **Correlation** |
|---|---|
| Fixed Acidity | -0.114 |
| Volatile Acidity | -0.195 |
| Citric Acid | -0.009 |
| Residual Sugar | -0.098 |
| Chlorides | -0.210 |
| Free Sulfur Dioxide | 0.008 |
| Total Sulfur Dioxide | -0.175 |
| Density | -0.307 |
| pH | 0.100 |
| Sulphates | 0.054 |
| Alcohol | 0.436 |

**Table 2.** Feature correlation with White Wine quality

The most relevant feature is Alcohol with a correlation of 0.436. An argument can also be made for lower correlation features such as Density (-0.307) and Chlorides (-0.210) but the correlation is significantly in the low correlation band.

## 4.3 LASSO and Cross-Validation Feature Selection

### 4.3.1 Steps

Lasso regression is a type of linear regression that applies a penalty to the size of the coefficients which is a method for feature selection. It modifies the least squares objective function by adding a penalty term that is proportional to the absolute value of the coefficients. This penalty term is controlled by a parameter $\lambda$, which determines the strength of the penalty. In other words, it is a regularization method. The objective function for Lasso regression can be represented as:

$$\text{Loss} = \text{Error}(Y - \hat{Y}) + \lambda \sum_{i=1}^{n} |w_i|$$

where ERROR is RSS, $\lambda$ is the regularization parameter that controls the magnitude of the penalty, and $w_i$ is the coefficient for the predictor.

*Note:* The features selected by the LASSO model are the ones that have non-zero correlation coefficients.

### 4.3.2 Red Wine Results

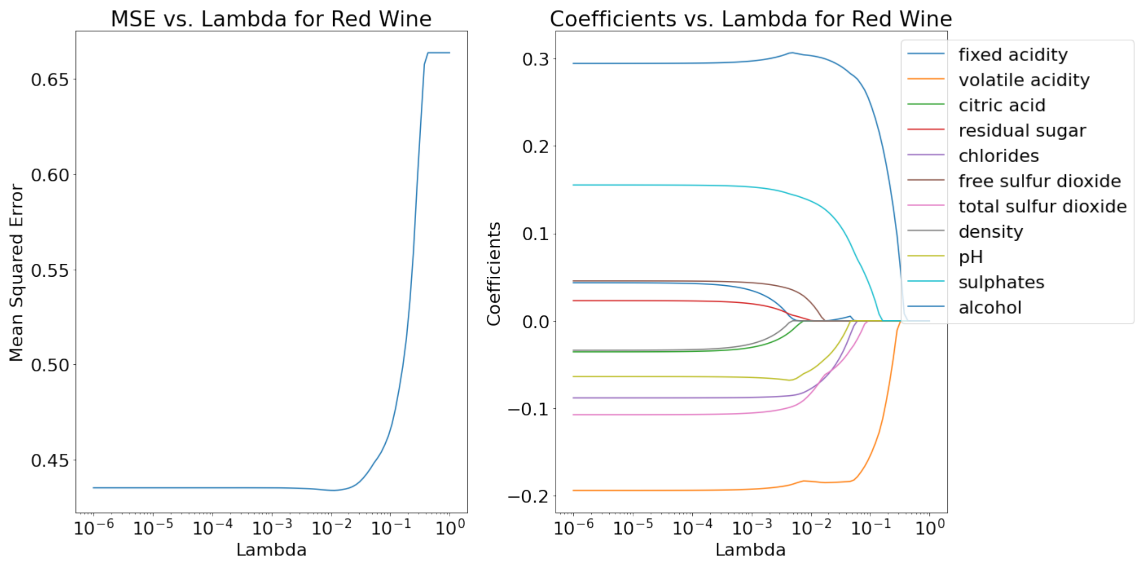

**Figure 5.** MSE of Cross-Validated Models (10 k-folds) among all lambdas on log scale for Red Wine

| **Selected Features By Lasso** |
|---|
| Volatile Acidity |
| Chlorides |
| Free Sulfure Dioxide |
| Total Sulfur Dioxide |
| pH |
| Sulphates |
| Alcohol |

**Table 3.** Lasso feature selection for Red Wine

### 4.3.3 White Wine Results

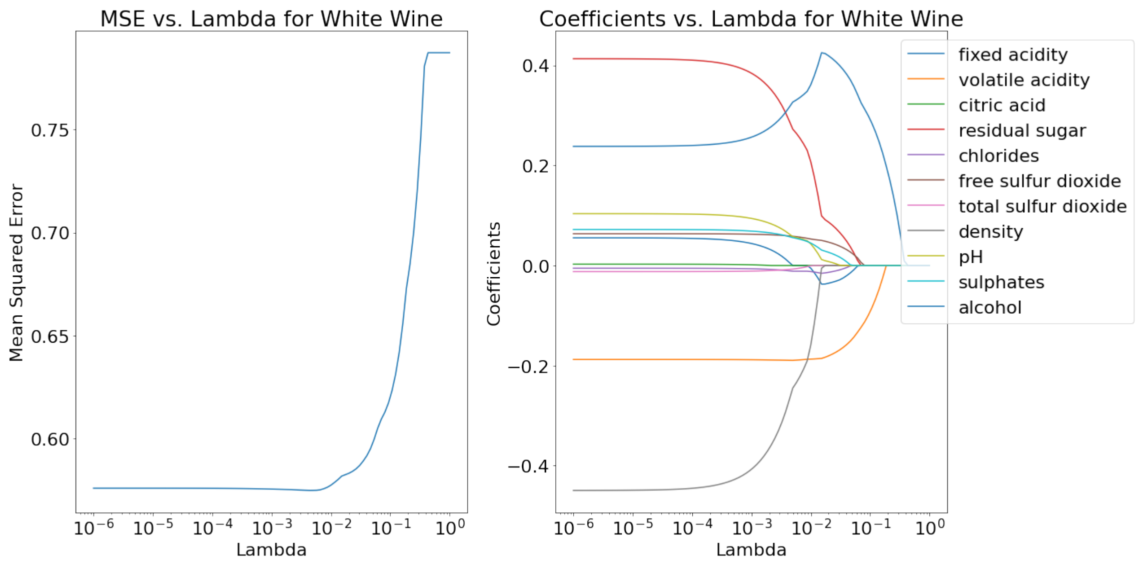

**Figure 6.** MSE of Cross-Validated Models (10 k-folds) among all lambdas on log scale for White Wine

| **Selected Features By Lasso** |
|---|
| Fixed Acidity |
| Volatile Acidity |
| Residual Sugar |
| Chlorides |
| Free Sulfure Dioxide |
| Total Sulfur Dioxide |
| Density |
| pH |
| Sulphates |
| Alcohol |

**Table 4.** Lasso feature selection for White Wine

In Figures 5 and 6, the graphs on the right showing the coefficients' correlations vs the regularization metric (or penalty) are a represenation of how much penalty it takes to drop a feature. Almost all features survived higher level of penalty for White Wine, only depleting to 0 after $10^{-2}$ on the log scale, but for Red Wine, more variables converged to 0 before $10^{-2}$. This is shown in how the LASSO model selected its features.

Between absolute threshold and LASSO, there is a tradeoff. If the goal is to build a predictive model that balances complexity and performance with an integrated approach to feature selection, LASSO is better. On the other hand, if the goal is quick initial feature reduction, setting an absolute correlatiopn threshold is better.

It could be beneficial to use both methods in a complementarily. Begin with a correlation threshold to quickly eliminate clearly irrelevant features and then apply LASSO to refine the feature set and optimize the model.

## 4.4 LASSO Features in KNN

### 4.4.1 Steps

Similar to the Titanic dataset, the data must be pre-processed by encoding categorical values and scaling numerical ones. Then, select the features from the Red Wine dataframe that have been selected by LASSO. Then, train the KNN model, evaluate it by cross validating across different subsets, then evaluate MSE and $R^2$.

### 4.4.2 Results

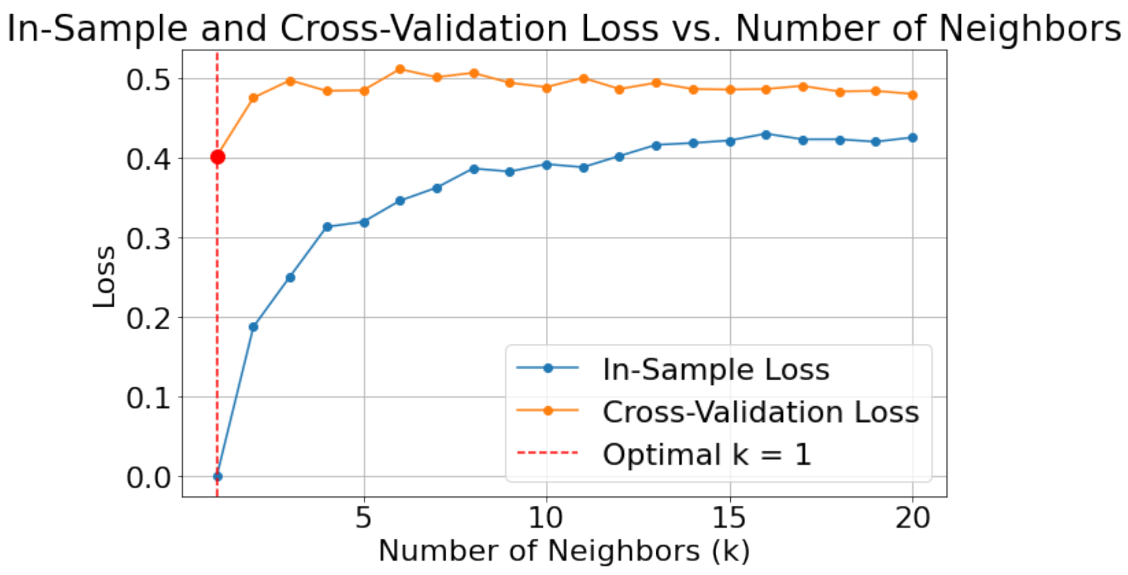

**Figure 7.** In-Sample and Cross-Validation (10-kfolds) Loss vs. Number of Neighbors for Red Wine

Cross-Validation Accuracy Average: 59%

MSE: 53.44%

$R^2$: 18%

The Cross-Validation Accuracy of 59% is on the low end of predicting the quality of wine. Moreover, and MSE of 53.44% is a high error, which means that on average, the model was able to predict the quality of wine with a 53.44% difference. To put that into perspective, if the model was predicting an exam grade, this can take a student from a C to an A, or B to F. Moreover, $R^2$ is how well the model can describe the data. An $R^2$ of 0.18 means that the model only explains 18% of the data (or capture 18% of the variability). Either the features selected by LASSO are not optimal, or a KNN model is inappropriate for this application.

## 4.5 KNN vs Linear Regression

The linear regression model's metrics are:

MSE: 39.12%

$R^2$: 40%

The linear regression model definitely outperformed the KNN model in predicting the quality of wine with an MSE of 39.12% and $R^2$ of 40%. This is a significant increase relative to the KNN model.

## 4.6 Advantages and Disadvantages

### 4.6.1 Linear Regression Advantages

Interpretability, efficiency, and straightforward relationships

### 4.6.2 Linear Regression Dis-Advantages

Linearity can sometimes lose complexity in data, sensitive to outliers, and very prone to collinearity and multicollinearity

### 4.6.3 KNN Advantages

Flexibility, simplicity, and non-parametric which allows for multi-class handling, easy implementation, and no rigid assumptions made about the data

### 4.6.4 KNN Dis-Advantages

Scalability, sensitivity to noise, and dependency on the right k-choice can make KNN perform poorly with large datasets as it is sensitive to noise, while also misconfiguring k leading to poor results as shown in this model.