Sanju Sathiyamoorthy 20908541

Rishi Sarkar 20894095

Preprocessing

For the wine and abalone datasets, the previous preprocessing code was reused.

Forest Fire Dataset

The preprocessing for the forest fire dataset was uniquely handled due to the highly skewed number of data points and range of target values. To combat this, the target values from the field 'Area' were modified by taking their log values. This allowed for a smaller scale of values to be analyzed from a scale of 0 to 10 instead of 0 to 1000.

Here is the graph of the comparison of the frequency of values between the normal target and the log transformed target.

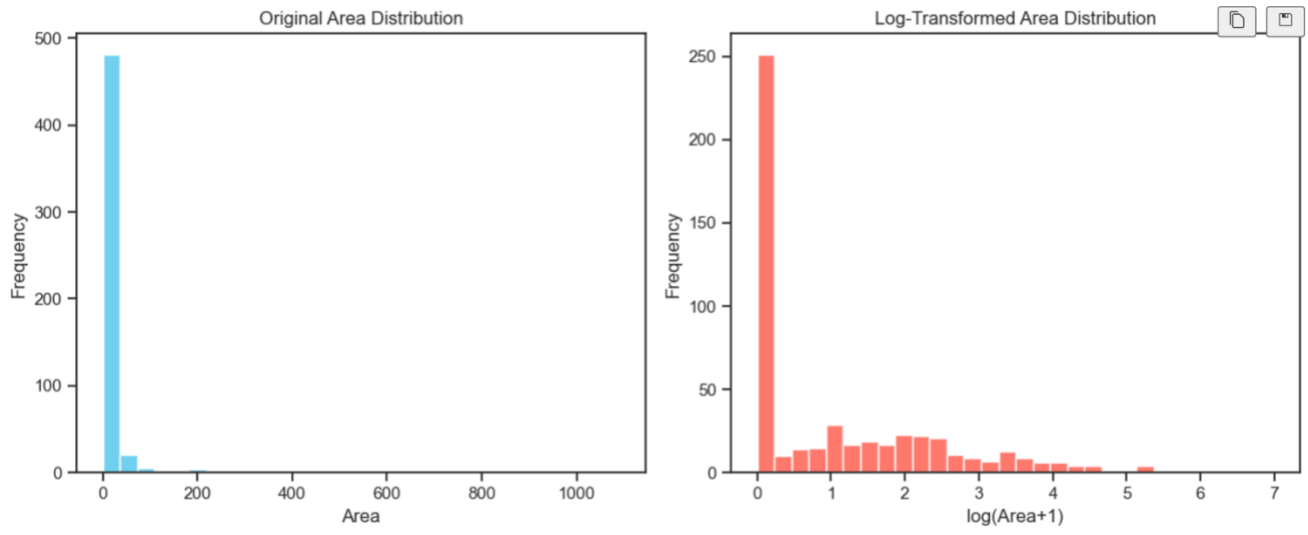

Figure 1: Preprocessing on Forest Fire Dataset

MLE and MAP Derivation

1.

a. Define the form of the likelihood term

For N independent samples, the likelihood is the product as follows.

$$L(b) = \prod_{\{i=1\}}^{N} f(x_i \mid a, b) = \prod_{\{i=1\}}^{N} [a\ b\ x_i^{a-1}(1 - x_i^a)^{b-1}]$$

b. Take the log of the likelihood

$$l(b) = \ln L(b)$$

$$l(b) = \sum_{i=1}^{N} [\ln a + \ln b + (a-1)\ln x_i + (b-1)\ln(1 - x_i^a)]$$

$$l(b) = N \ln a + N \ln b + (a-1)\sum_{i=1}^{N} \ln x_i + (b-1)\sum_{i=1}^{N} \ln(1 - x_i^a)$$

c. Derivative of log-likelihood

$$\frac{d\ l(b)}{db} = \frac{N}{b} + \sum_{i=1}^{N} \ln(1 - x_i^a)$$

$$\frac{N}{b} + \sum_{i=1}^{N} \ln(1 - x_i^a) = 0$$

d. Solve for target (b)

$$\frac{N}{b} = -\sum_{i=1}^{N} \ln(1 - x_i^a)$$

$$b = \frac{-N}{\sum_{i=1}^{N} \ln(1 - x_i^a)}$$

Due to 1-x_i always being above 0, the estimator is always positive.

2.

a. Define the form of the log-posterior term

$$p(b \mid x) \propto L(b) * p(b)$$

$$\ln\big(p(b \mid x)\big) \propto \ln\big(L(b)\big) + \ln\big(p(b)\big)$$

$$\ln\big(p(b \mid x)\big) \propto N \ln a + N \ln b + (a-1)\sum_{i=1}^{N} \ln x_i + (b-1)\sum_{i=1}^{N} \ln(1-x_i^a) + \ln\left(\frac{1}{\sqrt{2\pi\sigma^2}}\right) - \frac{(b-\mu)^2}{2\sigma^2}$$

Ignoring constant terms that are independent of b, this simplifies to the following.

$$\ln\big(p(b \mid x)\big) \propto N \ln b + (b-1)\sum_{i=1}^{N} \ln(1-x_i^a) - \frac{(b-\mu)^2}{2\sigma^2}$$

b. Take the derivative of the log-likelihood and set to zero

$$\frac{\mathrm{d}}{\mathrm{db}}\ln\big(p(b \mid x)\big) = \frac{N}{b} + \sum_{i=1}^{N} \ln(1-x_i^a) - \frac{b-\mu}{\sigma^2} = 0$$

$$\frac{N}{b} + \sum_{i=1}^{N} \ln(1-x_i^a) = \frac{b-\mu}{\sigma^2}$$

$$N\sigma^2 + b\sigma^2 \sum_{i=1}^{N} \ln(1-x_i^a) = b(b-\mu)$$

$$b^2 + \left(-\mu + \sigma^2 \sum_{i=1}^{N} \ln(1-x_i^a)\right) b - N\sigma^2 = 0$$

Using the quadratic equation, here is the following derivation for b.

$$b = \frac{\left(\mu - \sigma^2 \sum_{i=1}^{N} \ln(1-x_i^a)) \pm \sqrt{(\mu + \sigma^2 \sum_{i=1}^{N} \ln(1-x_i^a))^2 - 4N\sigma^2} \right)}{2}$$

Regression

For each of the dataset's a multitude of regression models were used to predict the target variable of quality, rings and logarithmic area for the wine, abalone and forest-fire datasets respectively. For the sake of clarity, each dataset will have its own section evaluating the different models' performance for that specific dataset. All model performance photos will be included in the appendix.

Wine Dataset:

	KNN	Decision Trees	Random Forest	Gradient Tree Boosting
RMSE	0.8878	0.9191	0.8289	0.8714
Variance	0.0001	0.0005	0.0001	0.0005

Every model has clear differences in terms of how they handle variance and root mean squared error (RMSE). KNN in specific performs well if its K value is optimal. At low values the model always overfits thus performing worse on the testing data, however as it rises it smooths out predictions. However, as the k value continues to increase past an optimal point it loses too much fine details and thus begins to underfit. The RMSE levels out after k=6, suggesting this to be a sweet spot. It has a best RMSE of 0.8878 and its variance of 0.0001 shows that it's very stable across different training folds.

Decision Trees follow a different pattern. As the depth of the tree increases, the RMSE initially drops, but after then rebounds after a certain point. It shows some signs of overfitting as the tree starts getting too complex and likely starts to memorize the noise and thus poorly generalizes. With a best RMSE of 0.9191 and variance of 0.0005, it's clear that this model is less reliable and has more fluctuation between the folds.

Random Forest solves some of the issues that decision trees have by averaging multiple trees and thus reducing its variance which is seen by the value of 0.0001. Its RMSE of 0.8289 is also the lowest of all models, accompanied with the variance suggests a highly consistent and accurate model. The RMSE improvement sharply slows down after 50 trees. This makes Random Forest a strong choice for generalization without overfitting.

Gradient Boosting also has a good performance as the model continuously improves itself with each iteration, leading to a best RMSE of 0.8714, which is better than KNN and Decision Trees, but slightly behind Random Forest. The slight downside is it has a variance of 0.0005, which is tied for the highest, however it is still relatively negligible. This may happen due to the fact that boosting builds on errors from previous iterations, making it more prone to amplifying this noise.

Abalone Dataset:

	KNN	Decision Trees	Random Forest	Gradient Tree Boosting
RMSE	1.4920	1.5307	1.4822	1.4796
Variance	0.0029	0.0054	0.0002	0.0013

K-Nearest Neighbors (KNN) follows a similar pattern as before, where RMSE decreases sharply as k increases before stabilizing at around k=10. This indicates that at smaller k values, the model capture too much noise and overfits thus leading to poor performance on the testing split. Larger values however smooth predictions and thus reduce overfitting. The best RMSE of 1.4920 is decent, however not as good as the ensemble models. The variance of 0.0029 is noticeably higher than the entire wine dataset, likely due to the inherence increased noise in the Abalone dataset.

Decision Trees show a similar U-shaped RMSE curve, in which shallow trees underfit while deeper trees overfit. The best RMSE is 1.5307 which is one of the highest amongst all the models, and the variance of 0.0054 is the worst variance value seen so far over both datasets. This may be because decision trees are highly sensitive to small changes. The decision tree once again has the worst performance relative to other likely due to its simplicity.

Random Forest once again has a relatively low best RMSE of 1.4822 which is slightly better than KNN. It also has the lowest variance for this dataset of 0.0002, suggesting that the model generalizes well and is less prone to overfitting than the rest. The RMSE curve flattens quickly after about 50 trees, showing diminishing returns in performance like its previous performance.

Gradient Boosting also performs well, with the lowest RMSE of 1.4796. This indicates that it captures complex relationships slightly better than Random Forest. However, its variance of 0.0013 is higher, meaning it is more variable across the k fold training split. The RMSE curve has a steep initial drop but then slightly increases, which could be due to overfitting at higher boosting stages.

Forest Fire Dataset:

	KNN	Decision Trees	Random Forest	Gradient Tree Boosting
RMSE	1.2206	1.3420	1.2140	1.2214
Variance	0.0024	0.0072	0.0001	0.0008

KNN follows the same trend as the previous 2, where RMSE decreases as k increases. There is a sharp decline until k =2, followed by a steadier decline until k=8 after which it levels out

completely. This once again suggests that smaller k values overfit, while larger values generalize better. The best RMSE of 1.2206 is relatively strong, though not the lowest. It's variance of 0.0024 however is not the best indicating that it varies more over different data splits.

Decision Trees performed the worst, with the highest RMSE of 1.3420 and the largest variance of 0.0072. This shows significant instability and inaccuracy which has been consistent over the datasets. The RMSE curve consistently does not follow so much of a U shape but instead increases with tree depth, indicating that deeper trees overfit badly. This overfitting plateaus at about a depth of 13. This suggests that the dataset is not ideal for decision trees.

Random Forest delivers the best results, with the lowest RMSE of 1.2140 and almost no variance (0.0001). This means it produces the most accurate predictions with stability across all different folds. The RMSE improvement sharply slows down after 50 trees, reinforcing that additional trees provide diminishing returns. The model effectively reduces overfitting by averaging multiple decision trees.

Gradient Boosting, surprisingly, does not outperform Random Forest here. Its RMSE of 1.2214 is slightly worse than Random Forest, and its variance of 0.0008 is higher, meaning it is more sensitive to the specific training splits. Unlike the previous datasets, the RMSE curve increases linearly with more boosting stages. This suggests that additional iterations are leading to overfitting rather than better generalization as the best value occurs at the lowest number of boosting stages.

Representation Learning

3.1 t-SNE plots

The wine dataset shows 2 major clusters. This suggests that there is some distinction, however with some noticeable overlap. Furthermore, the clusters are not that compact relative to how close they are to each other. This may indicate that while certain features help differentiate quality levels, others may also introduce noise, thus preventing a clean separation.

The abalone dataset forms a very clear spiral-like structure with 3 distinct sub structures, showing that age follows a very smooth and continuous pattern. The lower ringed fish all appear on the left sub structure. The presence of younger and older abalones suggests that the dataset's features align well to predict the age.

The forest fires dataset appears scattered, with no clear structure or clusters. This suggests that the features do not strongly correlate with the target variable in a way t-SNE can capture. This may possibly be due to an extremely biased dataset towards an area of 0. It may also be due to external factors such as weather conditions thus introducing randomness.

3.2 PCA

The Abalone dataset shows that the first principal component explains 85.93% of the variance, while the second component captures only 4.49%. This suggests that most of the dataset’s variability is captured in a single direction. The graph shows a very clear pattern resembling 3 distinct curved bands that start tight at the left and expand as it moves right. These clusters likely represent the influence of strongly correlated factors such as height, weight and length.

The Wine dataset has a more balanced distribution, with 29.77% variance in the first principal component and 23.16% in the second. This suggests more so that multiple features contribute to variability instead of a few key ones. There is a separation into 2 major clusters but the lack of a clear pattern suggests that PCA on its own may not entirely be useful for wine-quality determination.

The Forest Fires dataset shows that the first principal component explains 25.39% of the variance, while the second captures 13.51%. This suggests that no single direction dominates the data's variability and likely that it will take multiple components to capture any meaningful patterns. Similar to the t-SNE results, the scatter plot does not suggest any clear structure or pattern.

3.3 Secondary Feature Extraction Method

ICA was chosen as the secondary extraction method for the wine dataset as it is well-suited for separating independent signals in mixed data. ICA was used to attempt to extract independent

features. The plot shows 2 major clusters similar to the t-SNE but without any strong patterns suggesting that wine quality does not separate cleanly in a linear way.

Isomap was selected for the Abalone dataset as it is supposed to be effective for capturing nonlinear structures. The Isomap analysis revealed a very strong and distinct pattern that consisted of 3 lines going outwards spaced evenly around a point in a circular fashion. The pattern had very strong clustering and correlated strongly with the target variable as data points with lower rings were at the outer edges of the lines while the higher rings were closer to the center. This suggests that the dimensionality reduction with Isomap uncovers important correlations that PCA could not fully capture.

For the Forest Fires dataset, Isomap was used because fire size is influenced by many complex, nonlinear factors such as weather, geography, and fuel sources. Similar to the previous 2 sections, for this dataset there are no distinct patterns relative to the other datasets. In this dataset however, there were 2 clusters with low density that were formed. This semi structure implies that it is better at preserving nonlinear relationships in the data.

3.4 Scree Plots

The Wine dataset reaches 95% cumulative variance with about 8 components, suggesting that reducing the dimensions can still be effective in preserving most of the information. The curve follows a logarithmic shape showing a gradual flattening, indicating that the additional components contribute less significantly.

The Abalone dataset captures 90% variance with only 2 components, showing that a very small number of dimensions can represent the info and suggests that PCA is very effective for dimension reduction of this dataset. Like before, the flattening suggests that additional components do not add much.

The Forest Fires dataset requires about 10 components to reach 95% variance, implying that it is more complex and spread across multiple dimensions. Another thing to note is that there is a sharp elbow-like curve which likely indicates the point at which more components provide diminishing returns. This sharp turn occurs at about 8-10 components. For this dataset, dimensionality reduction may not be as useful.

3.5 Final Results

	KNN	Random Forest	Gradient Boosting
Wine	0.8948	0.8588	0.8675
Wine-PCA	0.8940	0.8623	0.8722
Wine-ICA	0.8994	0.8615	0.8723

	KNN	Random Forest	Gradient Boosting

Abalone	1.496	1.4769	1.4789
Abalone-PCA	1.573	1.5456	1.5454
Abalone-Isomap	1.5639	1.5548	1.5597

	KNN	Random Forest	Gradient Boosting
Forestfires	1.2046	1.2085	1.2525
Forestfires-PCA	1.2071	1.2037	1.2379
Forestfires-Isomap	1.2110	1.2030	1.2136

Random Forest consistently has the lowest RMSE, making it the most reliable model, while Gradient Boosting performs similarly for the Abalone dataset and KNN performs similar for the forest fires. PCA improves results for the forestfires dataset and performs worse for most others. One point that's surprising is how Abalone has the highest root mean squared error even though there is such a clear and strong relationship between certain features and the target variable which was shown by the various component analyses. This likely suggest that the lack of even representation due to an imbalanced dataset is the problem. This same imbalance persists in the forestfires dataset however it still has a lower average RMSE which is surprising.
Overall, ensemble methods such as random forest and gradient boosting are the most reliable. It is not evident if dimensionality reduction helps or not suggesting that its effectiveness varies depending on the dataset and the complex relationships within it.

Appendix

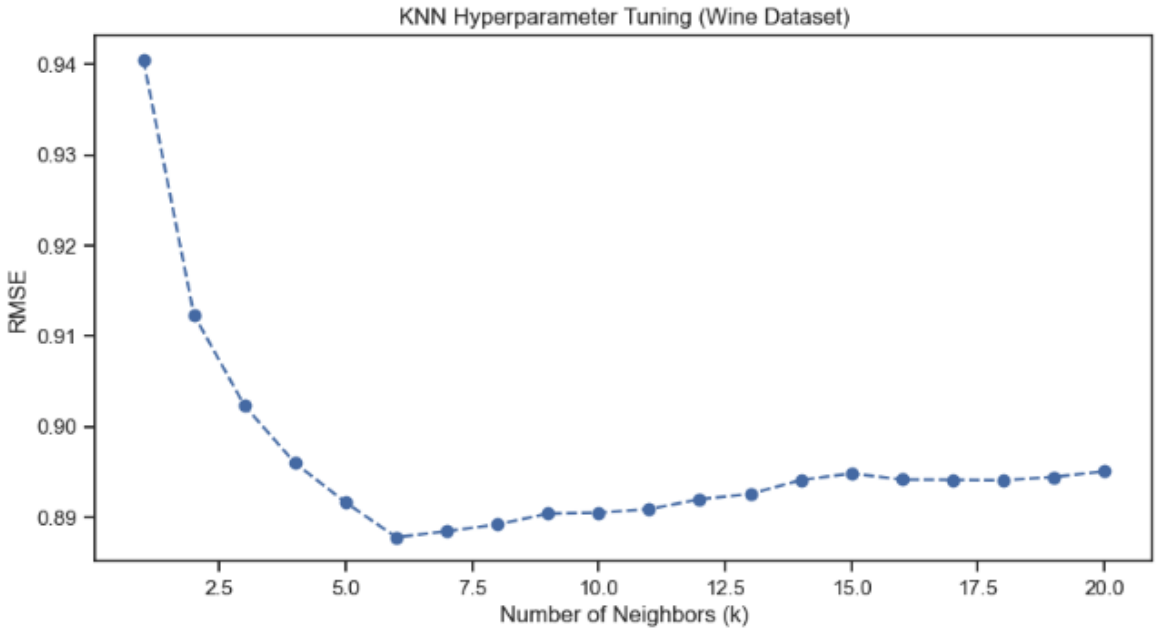

Figure 2: KNN Hyperparameter Tuning - Wine Dataset

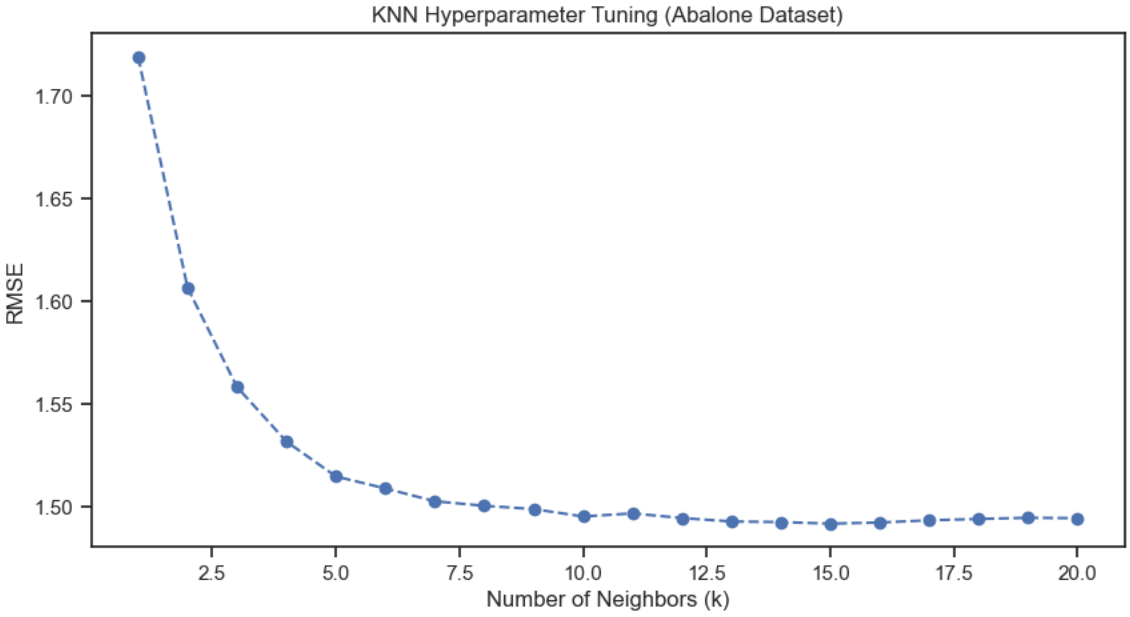

Figure 3: KNN Hyperparameter Tuning - Abalone Dataset

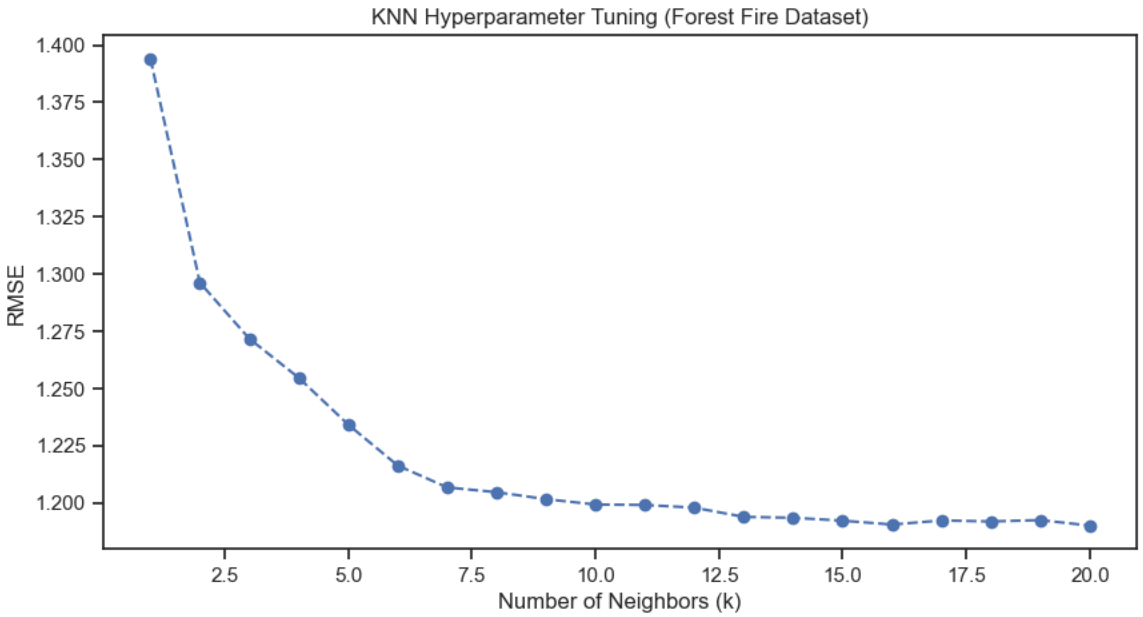

Figure 4: KNN Hyperparameter Tuning – Forest Fire Dataset

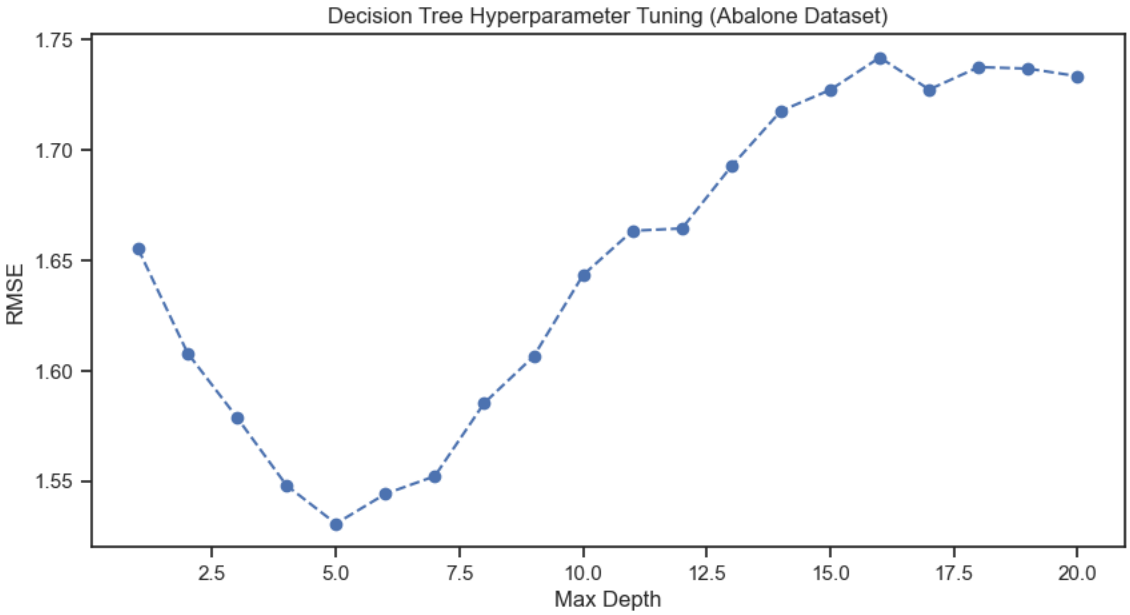

Figure 5: Decision Tree Hyperparameter Tuning - Abalone Dataset

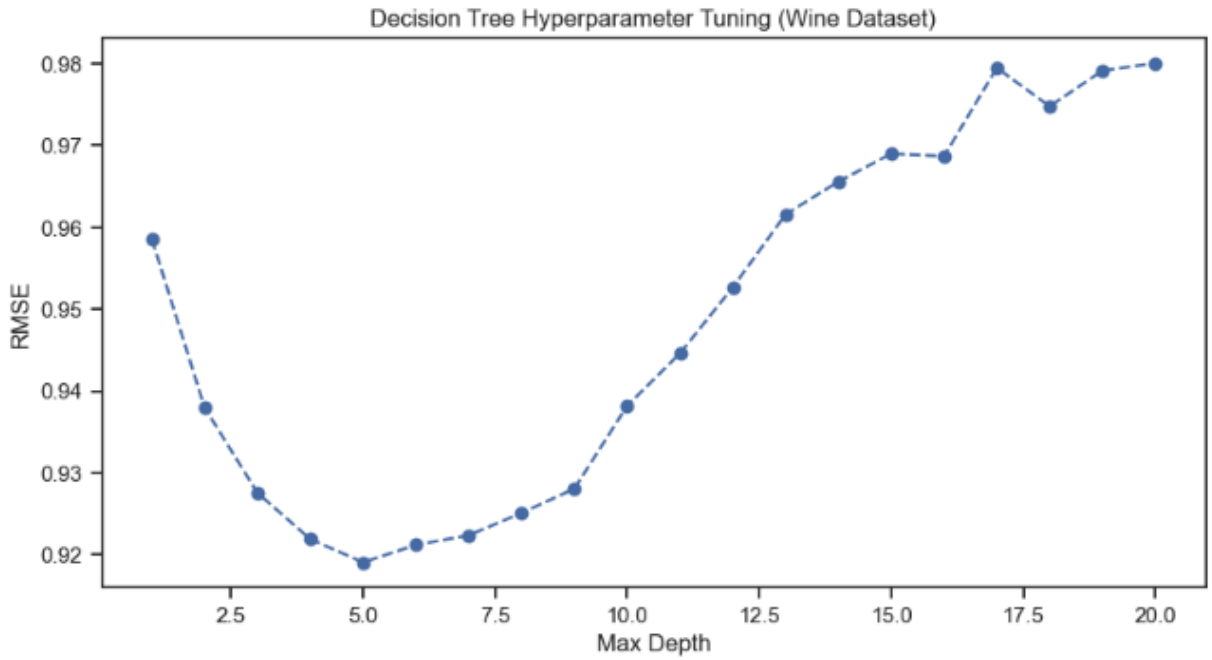

Figure 6: Decision Tree Hyperparameter Tuning - Wine Dataset

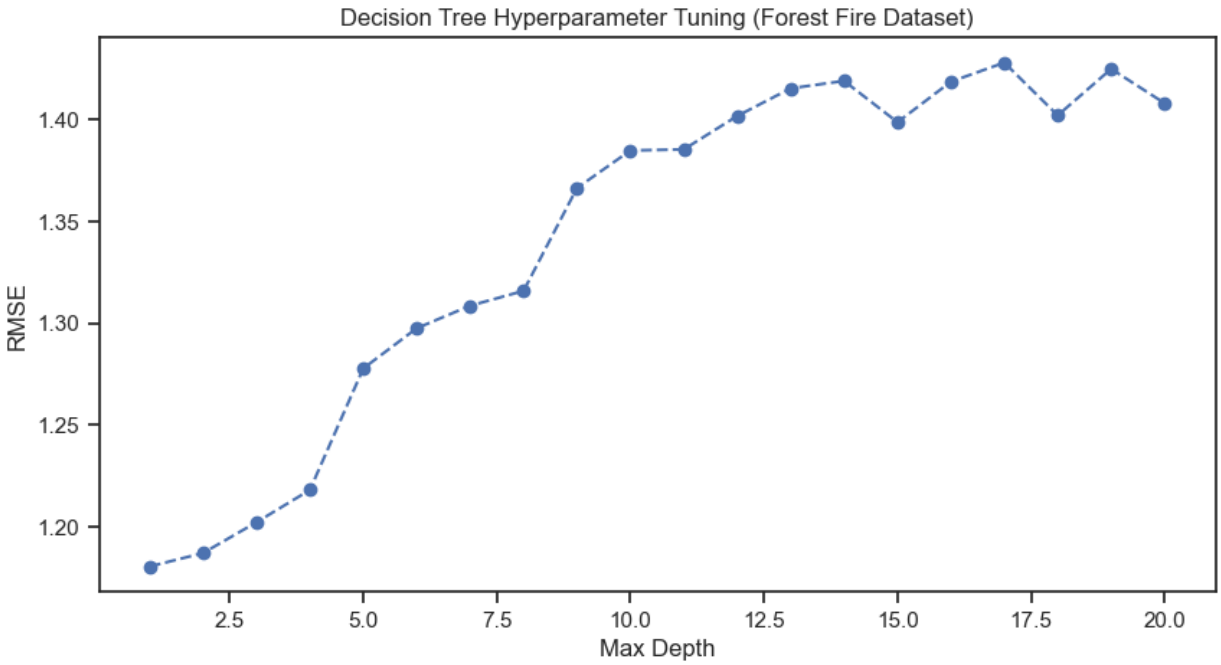

Figure 7: Decision Tree Hyperparameter Tuning – Forest Fire Dataset

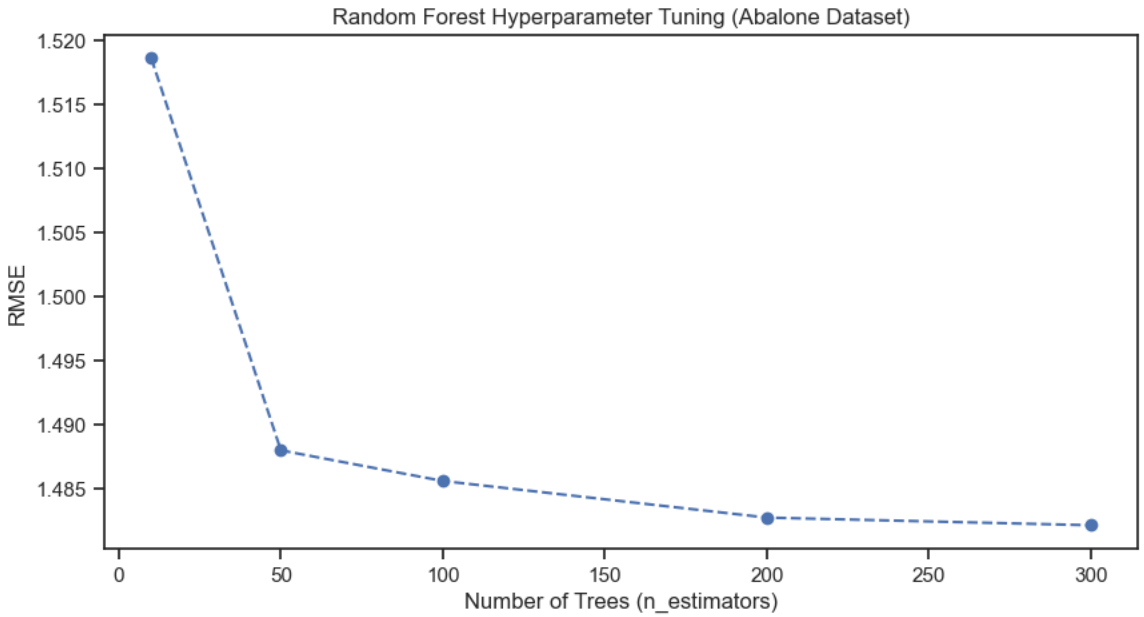

Figure 8: Random Forest Hyperparameter Tuning - Abalone Dataset

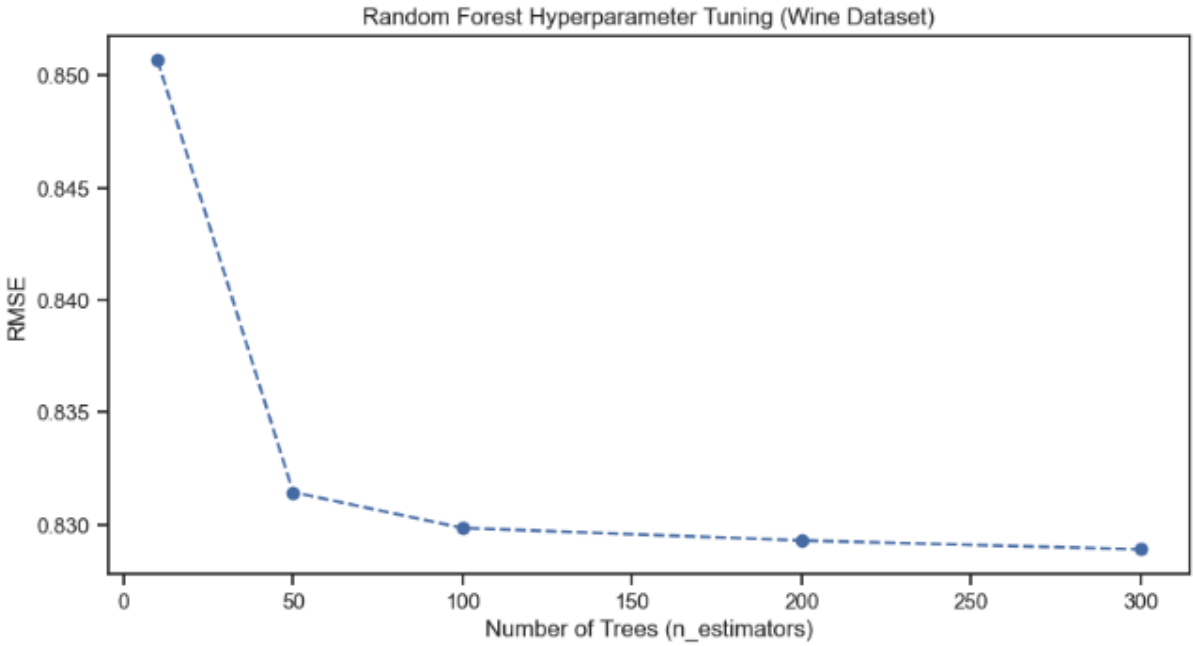

Figure 9: Random Forest Hyperparameter Tuning – Wine Dataset

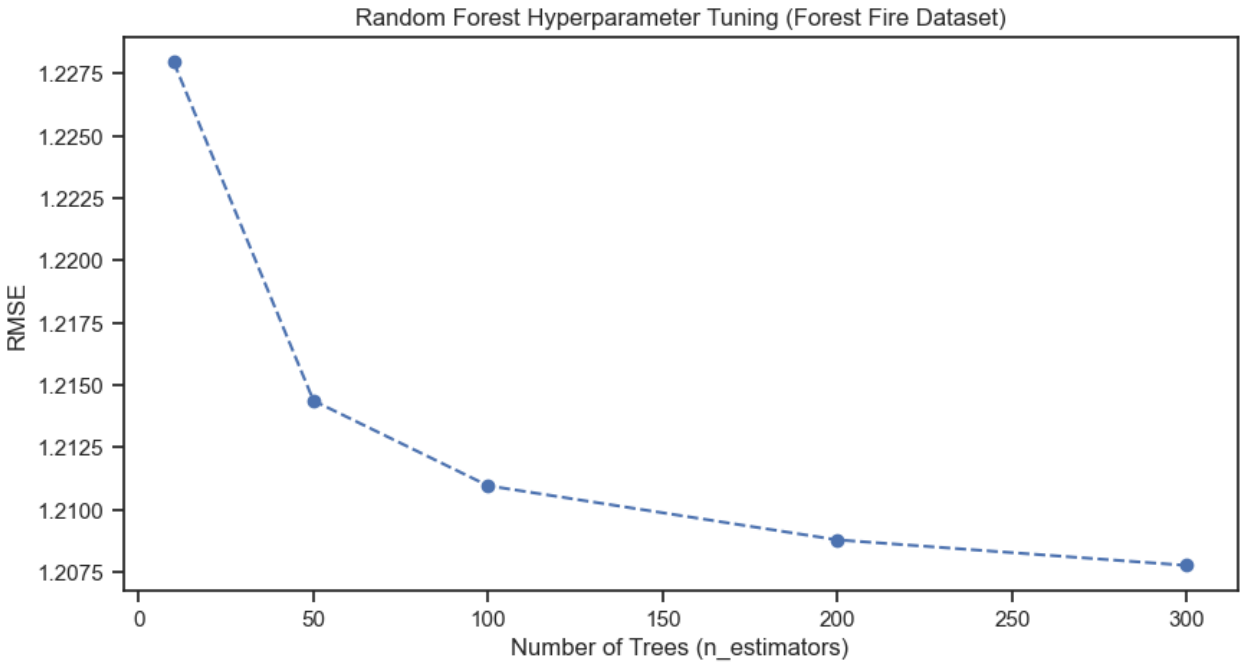

Figure 10: Random Forest Hyperparameter Tuning – Forest Fire Dataset

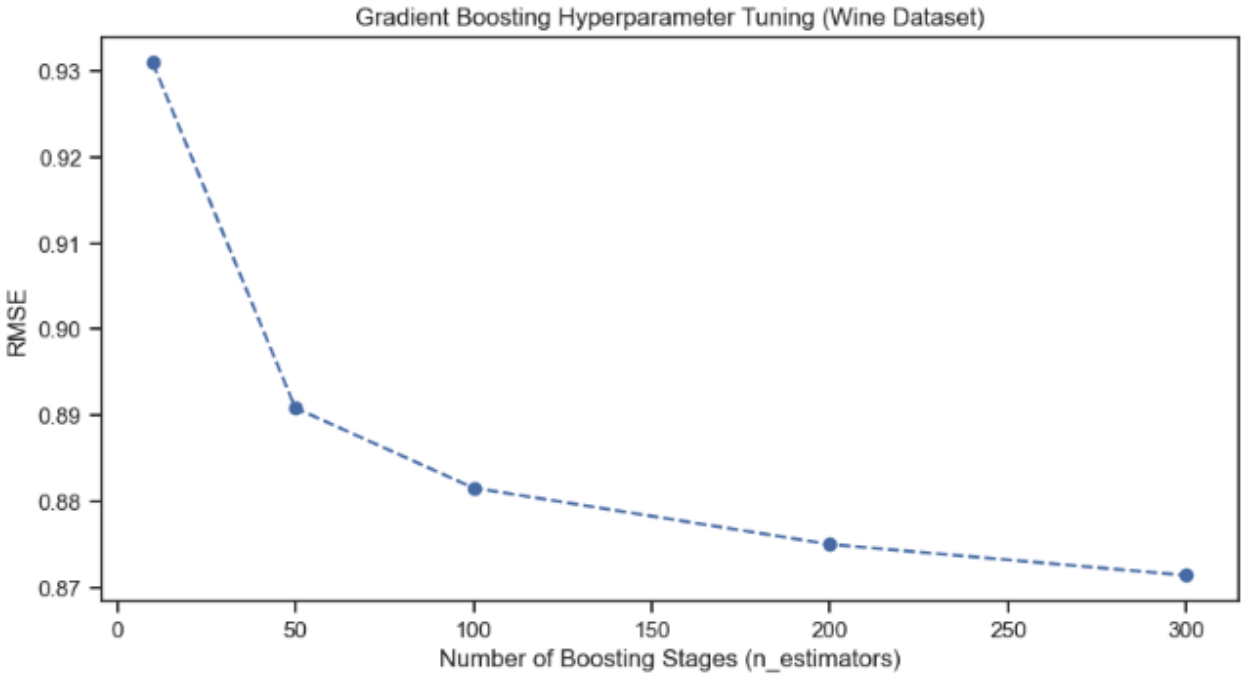

Figure 11: Gradient Boosting Hyperparameter Tuning - Wine Dataset

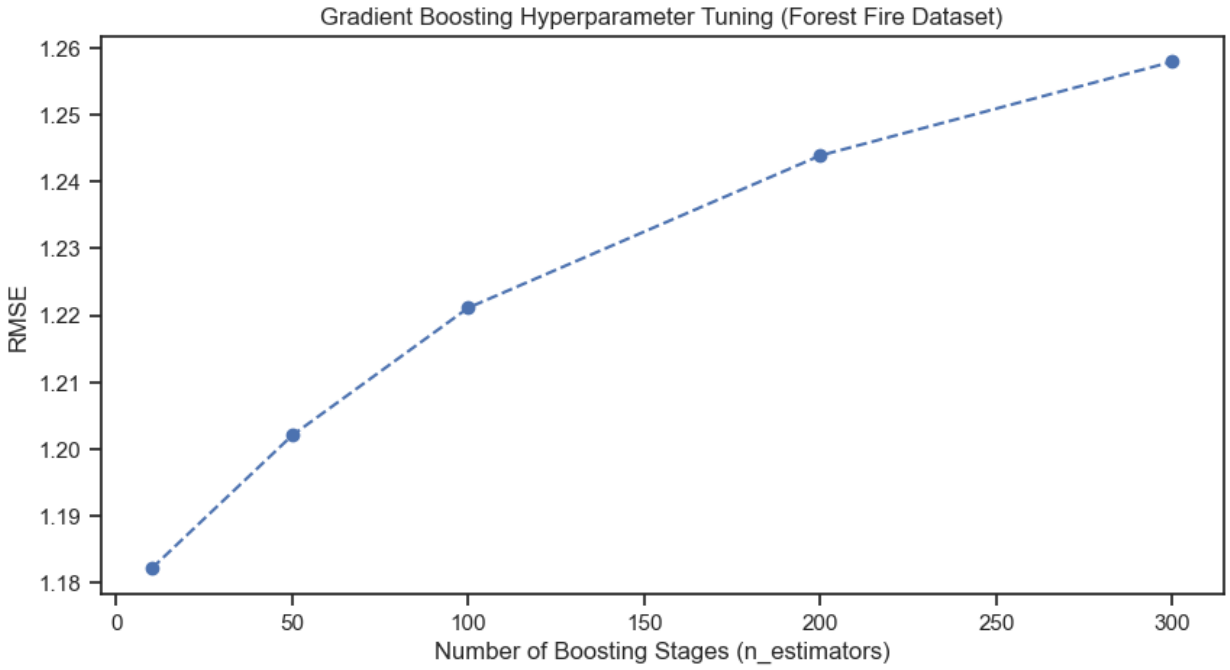

Figure 12: Gradient Boosting Hyperparameter Tuning – Forest Fire Dataset

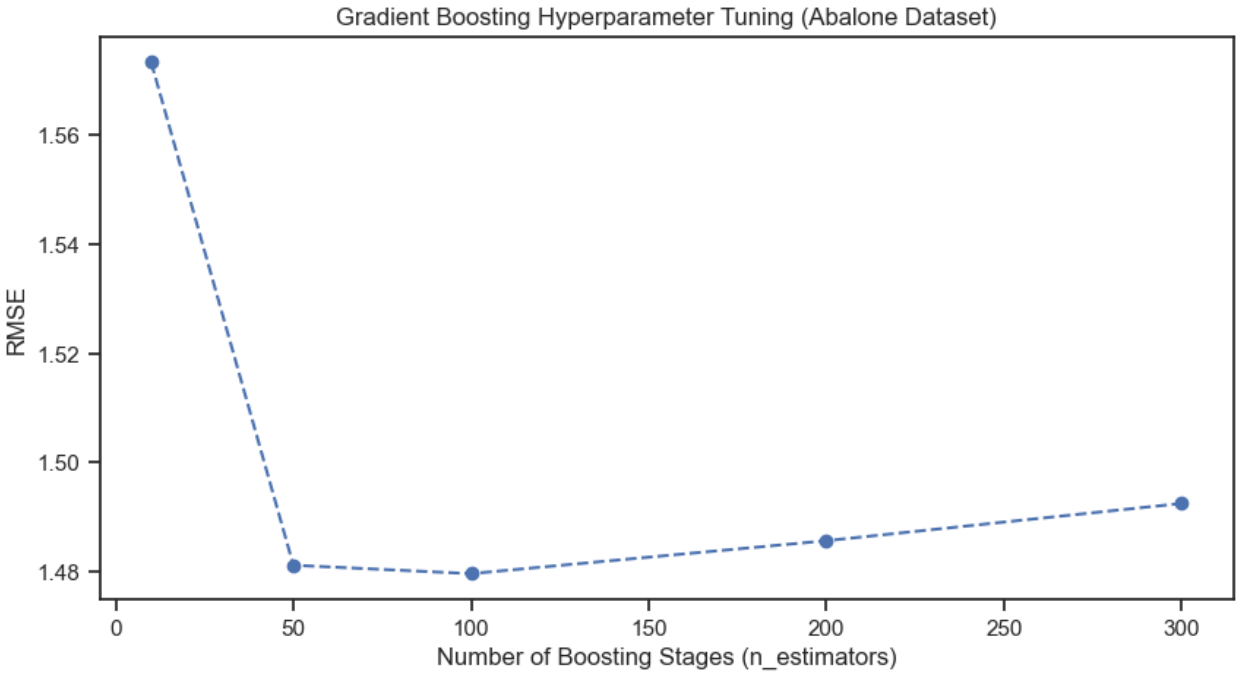

Figure 13: Gradient Boosting Hyperparameter Tuning - Abalone Dataset

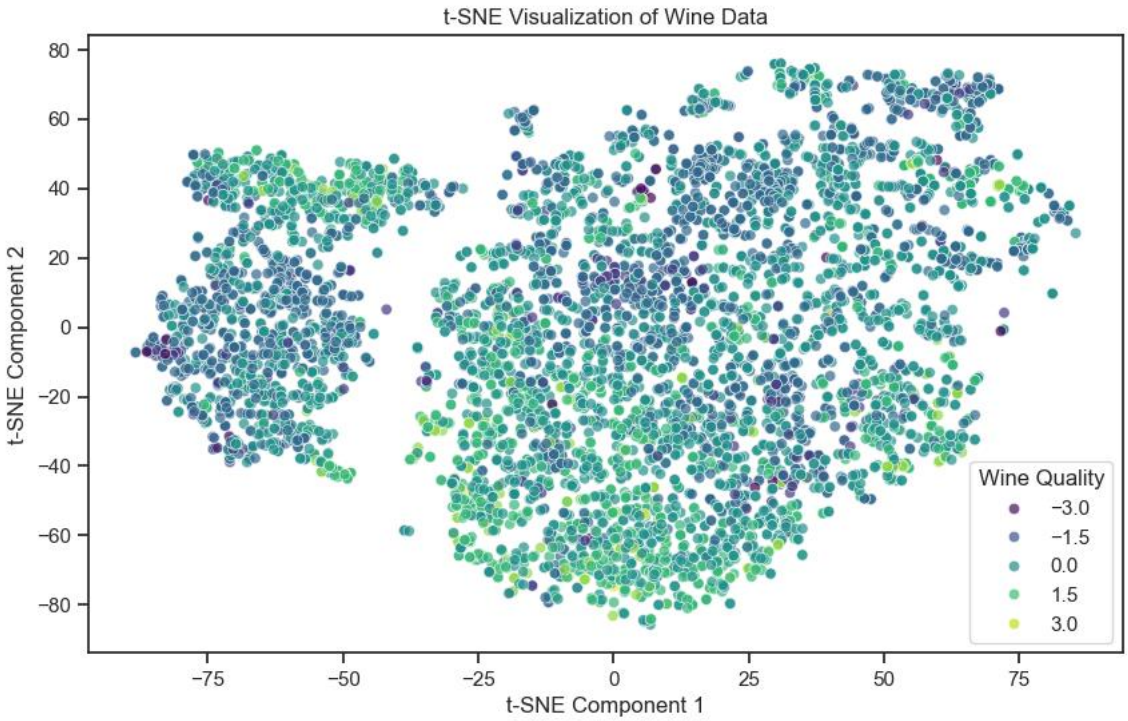

Figure 14: t-SNE Visualization of Wine Dataset

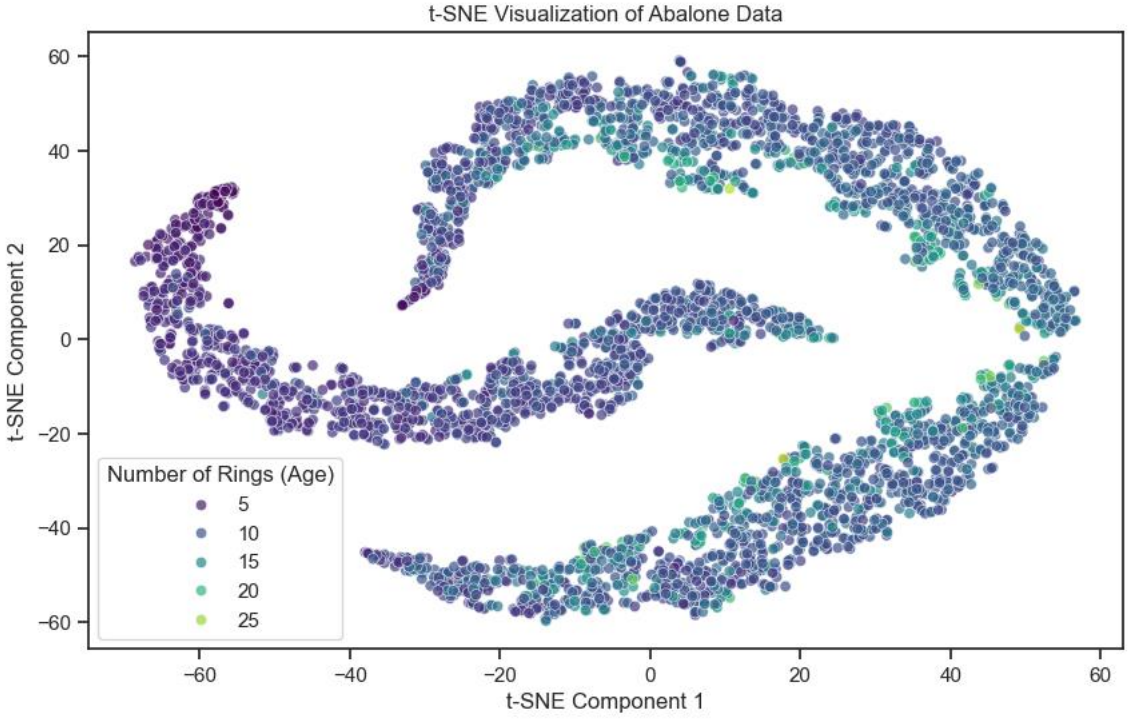

Figure 15: t-SNE Visualization of Abalone Dataset

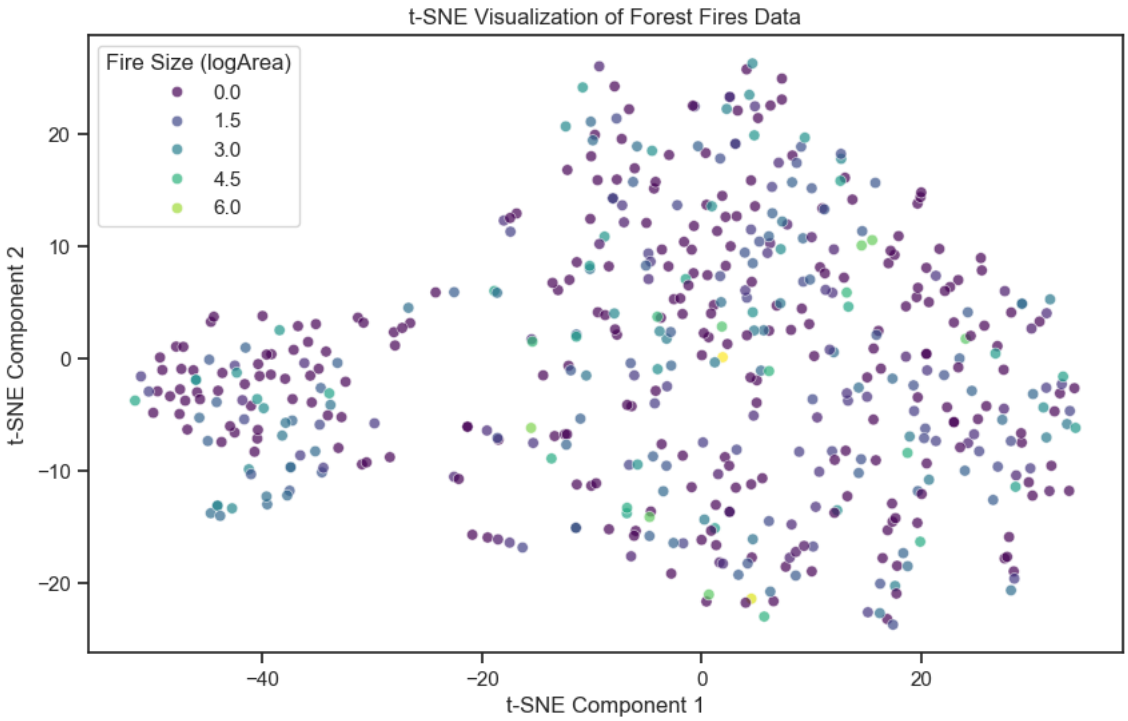

Figure 16: t-SNE Visualization of Forest Fire Dataset

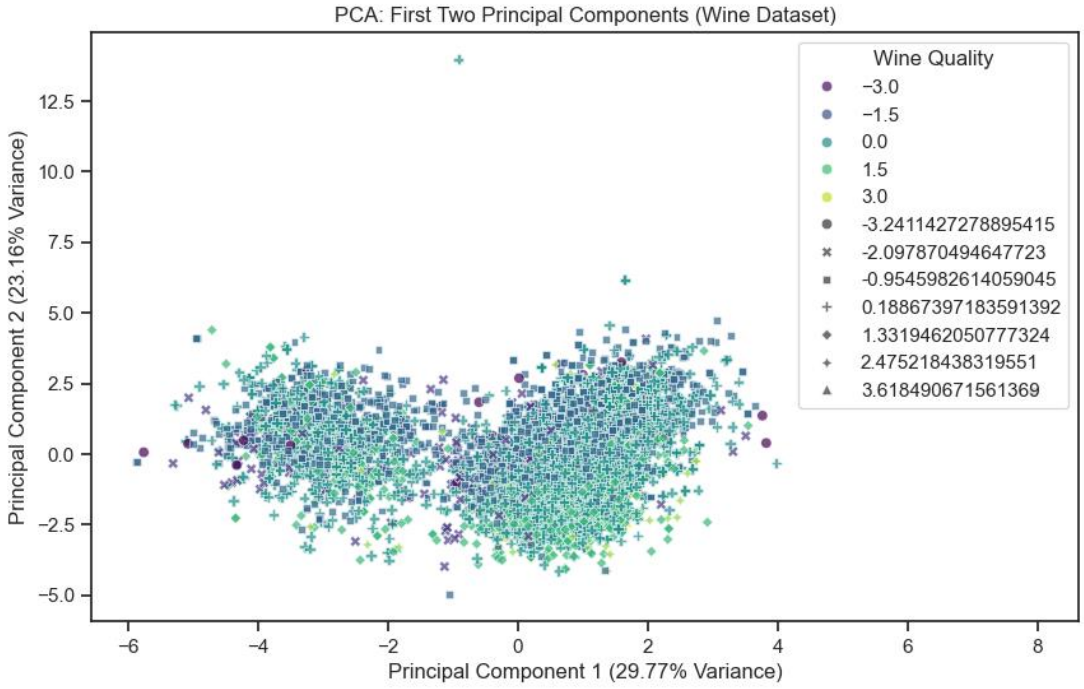

Figure 17: PCA: First Two Principal Components – Wine Dataset

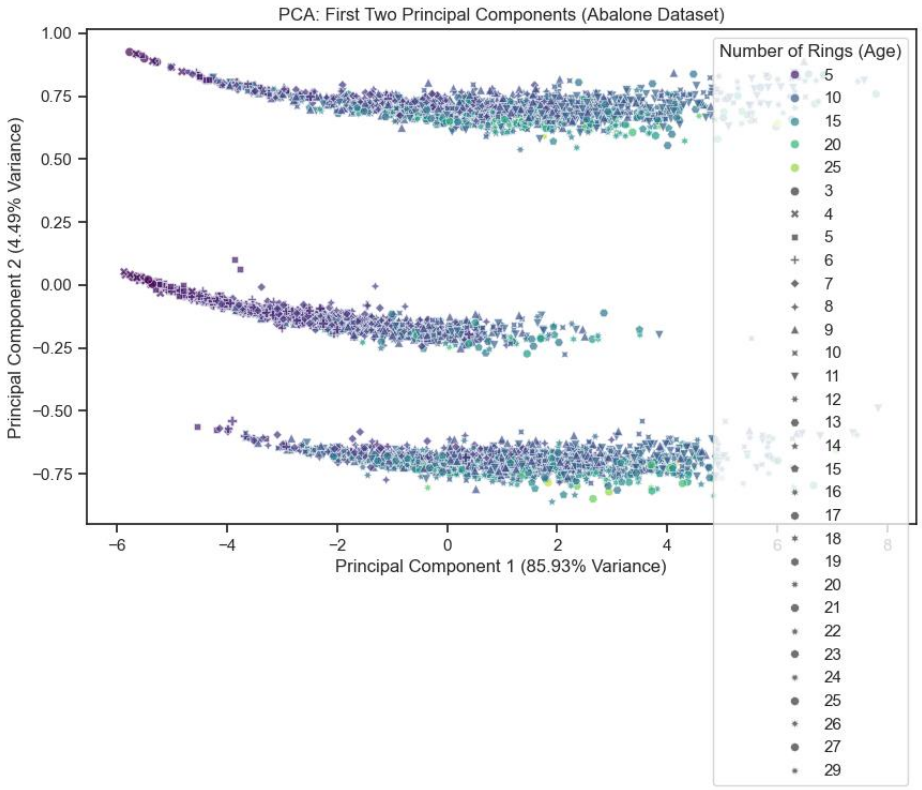

Figure 18: PCA: First Two Principal Components – Abalone Dataset

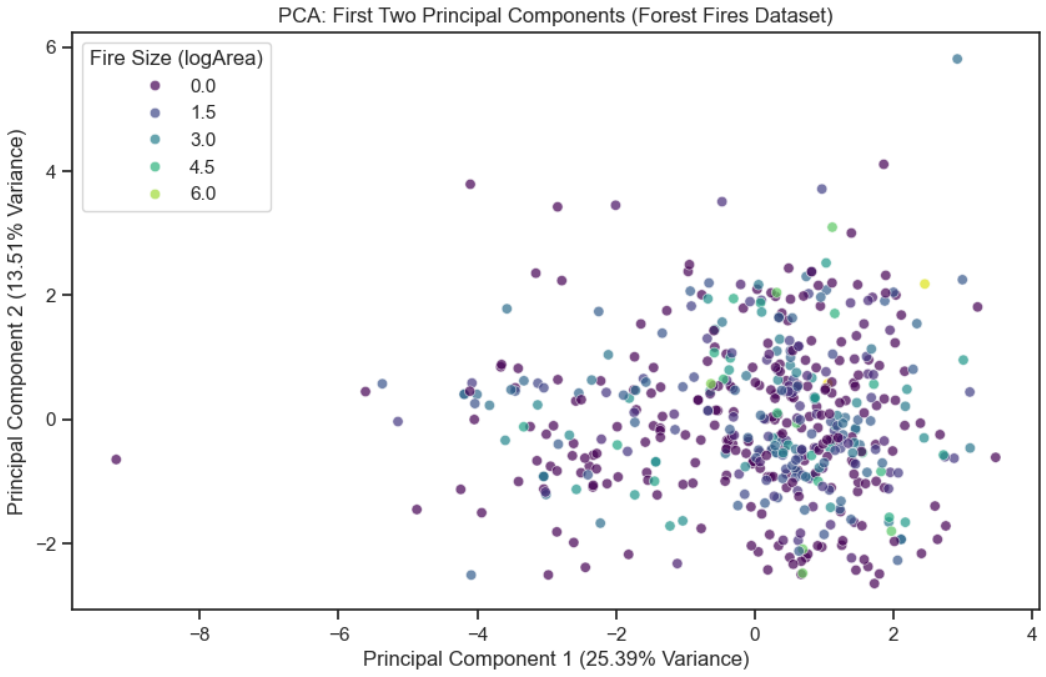

Figure 19: PCA: First Two Principal Components – Forest Fire Dataset

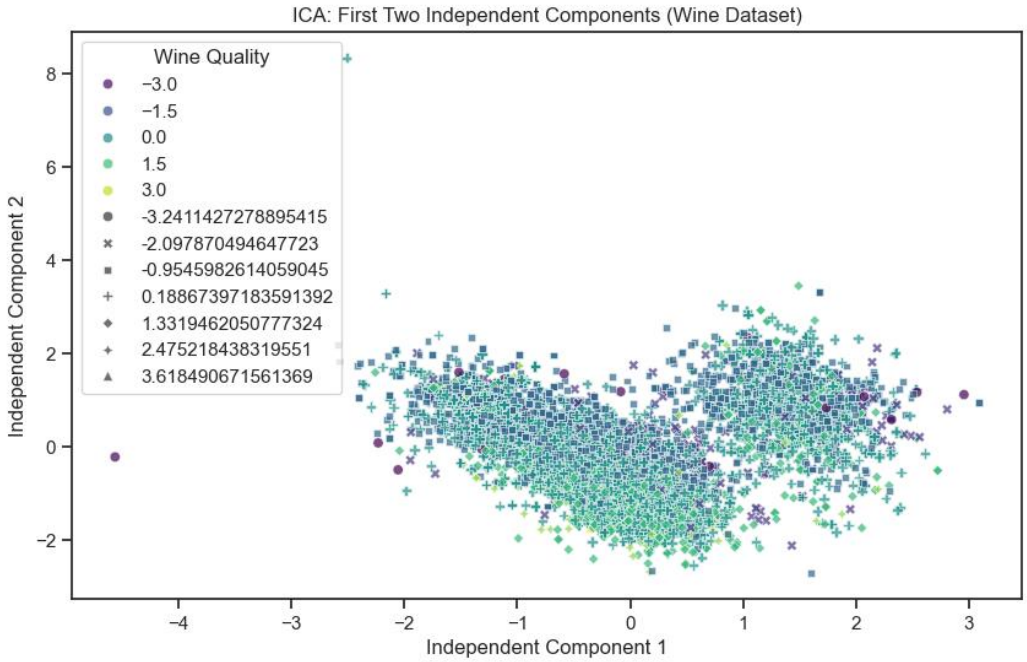

Figure 20: ICA: First Two Independent Components – Wine Dataset

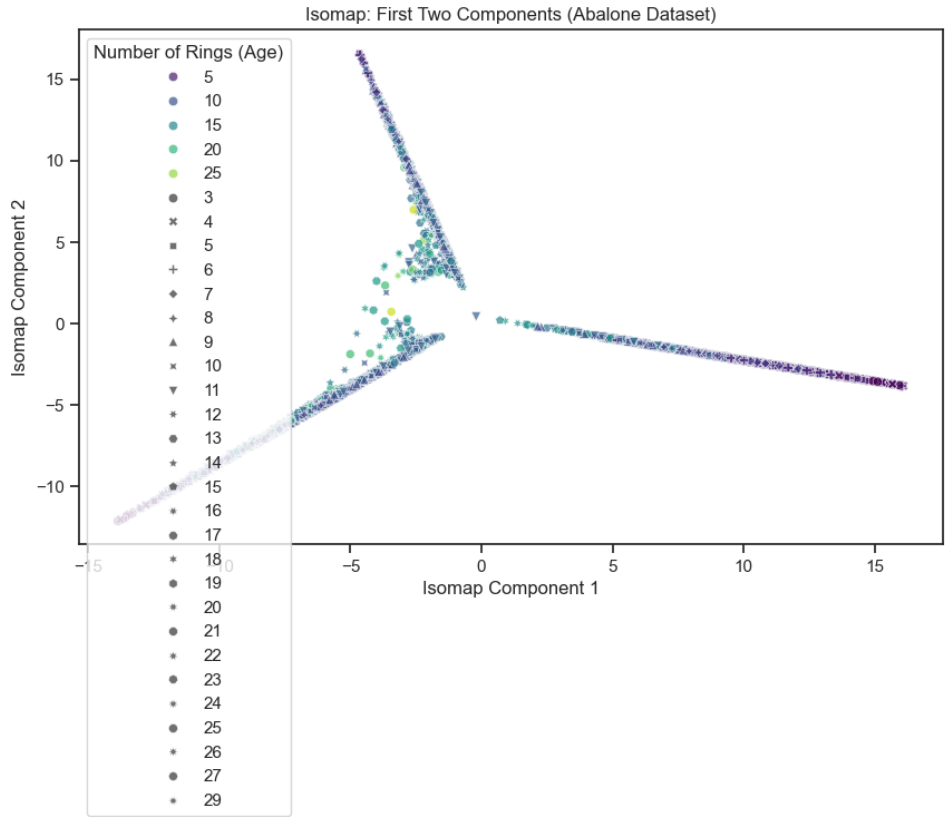

Figure 21: Isomap: First Two Components – Abalone Dataset

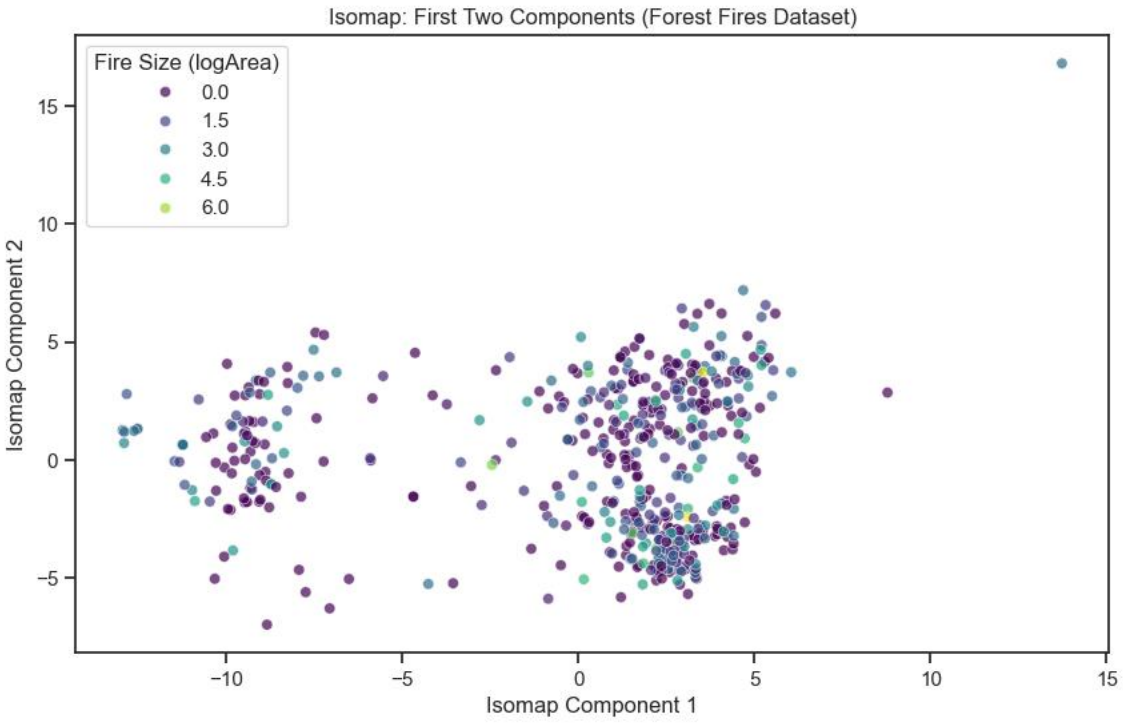

Figure 22: Isomap: First Two Components – Wine Dataset

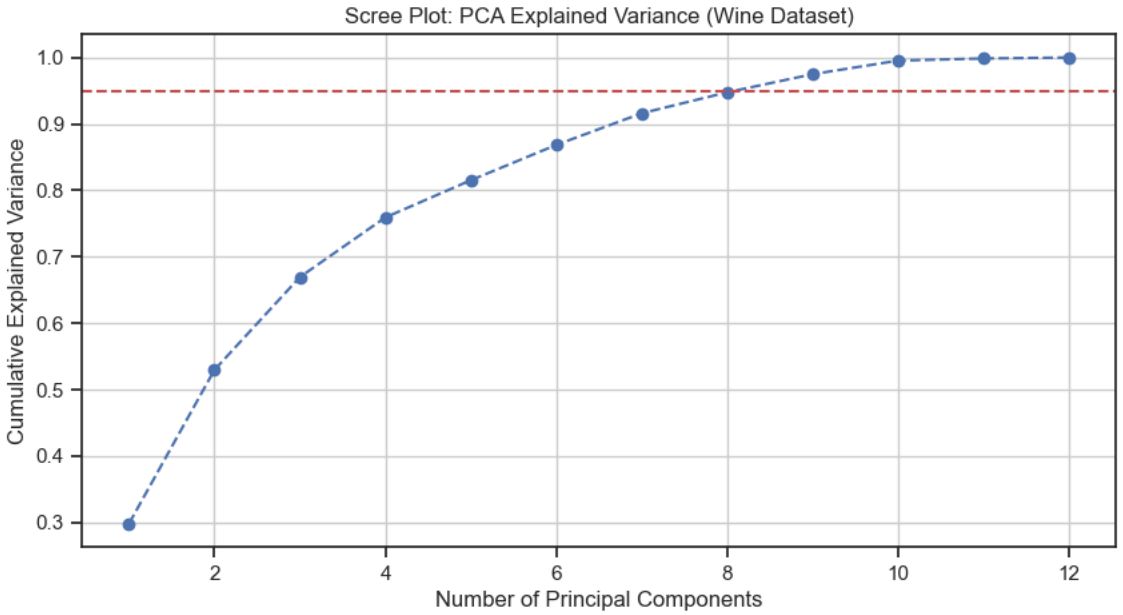

Figure 23: Scree Plot: PCA Explained Variance - Wine Dataset

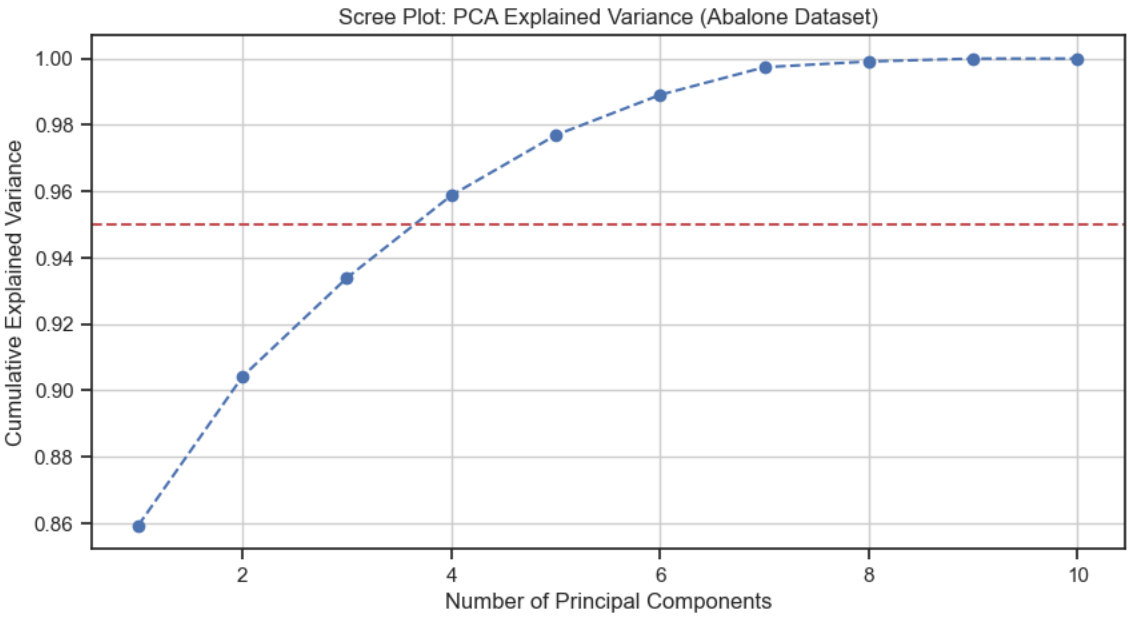

Figure 24: Scree Plot: PCA Explained Variance - Abalone Dataset

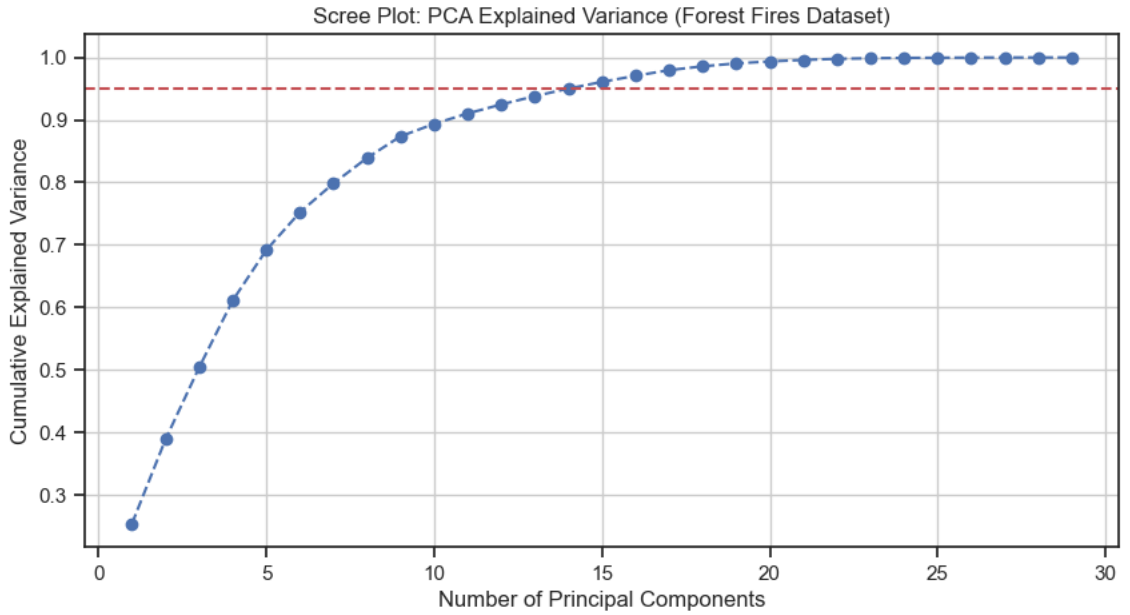

Figure 25: Scree Plot: PCA Explained Variance – Forest Fires Dataset