**FUNDAMENTALS OF MACHINE LEARNING**

**BA-64060-002**

**FINAL PROJECT 17 – MUSHROOM CLASSIFICATION**

**Project Member: Hitaishi Bairapureddy**

**Project Title: Mushroom Classification**

**Project Goal:**

This Project is based on the Mushroom dataset sourced from Kaggle. The main goal is to classify the mushroom into poisonous and edible using its characteristics. Accurate classification and identification of mushrooms is crucial for various applications, including food safety, ecological studies, and mycological research. This dataset likely contains various features describing the physical characteristics of mushrooms, such as cap shape, color, gill attachment, and spore print color. These features play a crucial role in differentiating between edible and poisonous species.

**Data Overview**:

This dataset sourced from kaggle is in CSV format and contains information on about 61,069 hypothetical mushroom specimens. Each row of the data contains information about one mushroom, and each column provides information regarding a certain characteristic of the given mushroom. These characteristics can be grouped into the following:

**Target Variable**: Edibility (class) - A categorical variable that specifies whether a mushroom is edible (e) or poisonous (p).

**Morphological Features:**

**Cap characteristics**:

Cap diameter grid\_3x3 - The diameter of the mushroom cap in centimeters.

Cap shape - A categorical variable that describes the shape of the mushroom's cap. (Example values: bell, conical, convex, flat, etc.)

Cap surface - A categorical variable that describes the surface texture of the cap. (Example values: fibrous, grooved, scaly, smooth, etc.)

Cap color - A categorical variable that describes the color of the cap of the mushroom. (Example values: brown, white, yellow, etc.)

Does bruise or bleed check - A binary variable that specifies whether the mushroom bruises or bleeds when injured.

**Gill characteristics:**

Gill attachment - A categorical variable that describes how the gills are attached to the stem. (Example values: adnate, adnexed, decurrent, etc.)

Gill spacing - A categorical variable that describes how far apart the gills are spaced from one another. (Example values: close, distant, none).

Gill color - A categorical variable that describes the color of the gills. (The same options as cap color, plus "none" for mushrooms that lack gills).

**Stem characteristics:**

Stem height grid - The height of the mushroom stem in centimeters.

Stem root – Presence and type – a categorical variable

**Data Preparation:**

1. **Summary Statistics**:

We have done summary statistics for all numerical variables, such as mean, median and standard deviation for continuous variables such as cap diameter and stem height.

Frequency tables for categorical variables such as cap shape and gill color.

**2. Data Dimensions & Structure**:

The initial data set was 61,069 samples and the number of columns of features is 20 with one target variable “class”.

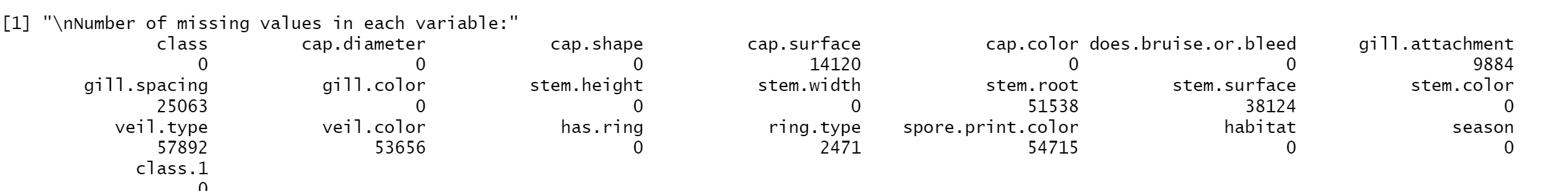
**3. Categorical Variable Encoding**:

The categorical variables representing cap shape, surface texture and color are encoded into numerical forms such as one-hot encoding and label encoding.

It will make machine learning algorithms understandable.

**4. Missing Value Imputation**:

The data set was checked for missing values.



For this, we used K-Nearest Neighbors (KNN) to fill in the missing values. It will estimate the missing values based on the features of a similar set of data points.

A close-up of a white background

Description automatically generated

**5. Outlier Detection and Removal:**

Outliers were detected in the data set, possibly using z-scores or IQR.

Since the number of outliers is relatively low, we removed them from the dataset for the model to focus on the core distribution of data.

A close-up of a number

Description automatically generated

**6. Data Preparation Impact:**

Since of missing value imputation and outlier removal, the final data set size is reduced to 55,762 samples.

**Exploratory Data Analysis**:

**1. Target Variable Distribution**:

The target variable, "edibility" (class), is binary:

Class 1: Poisonous (represented by 1)

Class 2: Edible (represented by 2)

Probable frequency tables were used to visualize class distribution. This helps in understanding the proportion of edible and poisonous mushrooms in the dataset.

A graph with black text

Description automatically generated

A number on a white background

Description automatically generated

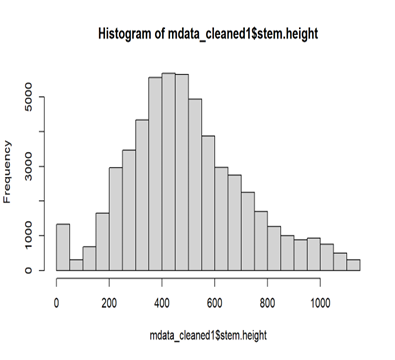
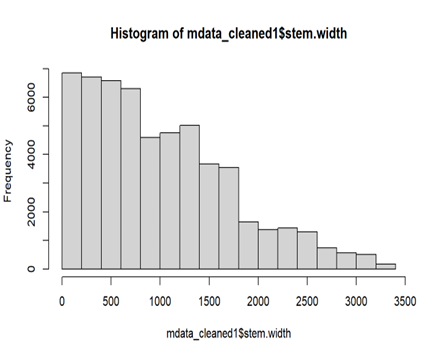
**2. Numerical Variable Analysis:**

**A diagram of a diagram

Description automatically generated**

Boxplots were used to explore the distribution of the numerical variables such as cap diameter and stem height.

Some possible outliers were detected from the boxplots. These outliers may have distorted the data distribution.

A graph of a graph

Description automatically generated

Histograms were used to visualize the distribution of these numerical variables.

These histograms probably indicate that the data is not perfectly symmetrical but skewed to the right (right-skewed). This suggests that a more significant portion of the data points most likely lie towards the lower end of the value range.

While outliers were removed in the data preparation, the boxplots from the EDA indicated that some outliers were possibly remaining.

**3. Categorical Variable Analysis**:

Frequency tables were prepared to analyze the distribution of the categorical variables such as cap shape, surface texture, and color.

These tables help understand the dominance of specific categories in each variable.

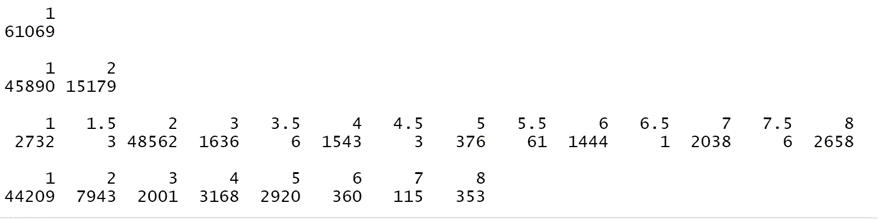
The results of distribution analysis states that the "veil type" variable had only one unique value. Since this variable does not discriminate the data, this can be excluded from the dataset.

A screenshot of a computer program

Description automatically generated

A white sheet with black numbers

Description automatically generated



**Feature Selection:**

1.Feature Importance Assessment:

There are several methods which were used to measure the importance of features:

**Analysis of Variance (ANOVA):** ANOVA was likely used to evaluate the statistical significance of each feature in relation to the target variable edibility. The report mentioned that the all variables provided high significance. It could be that:

The features are indeed important.

There may be strong correlations among features, therefore, the unique impact of each of them will be very difficult to detect.

A screenshot of a computer

Description automatically generated

**Correlation Matrix**: Because of potential correlations, a correlation matrix was likely produced. The report mentioned a high correlation between "cap surface" and "gill color." Because this is the case, perhaps the redundancy here would be that one feature gives almost the same information as the other feature.

A close up of a diagram

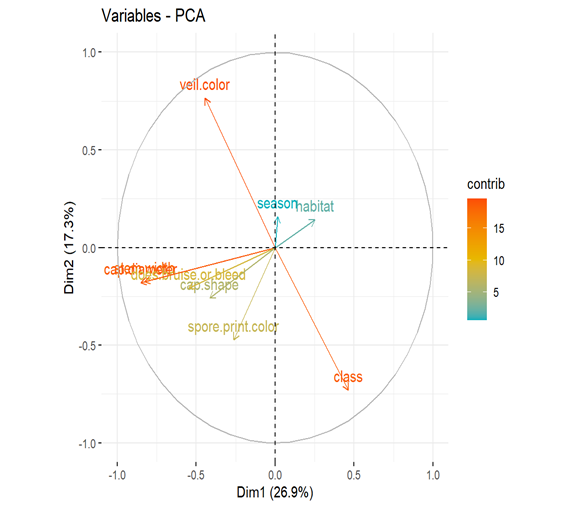
Description automatically generated

**Principal Component Analysis (PCA):** PCA was used to look for latent structure in the data and find the features that have the most contribution to the explained variance.

The results mentioned the "elbow" in the explained variance plot pointing to an optimal number of principal components of around 4 or 5. In other words, the load of the 4th or 5th principal component is sufficient to explain a considerable amount of variance in the data.

As complementary analysis, a PCA biplot was likely used for visualizing the relationship between the features and the principal components.

A diagram of a graph

Description automatically generated

Visualizing the biplot suggests features positively correlated with class, in particular, Habitat.

Features that show strong negative correlation with the "edible" class were likely identified, in particular, Stem width, Cap diameter.

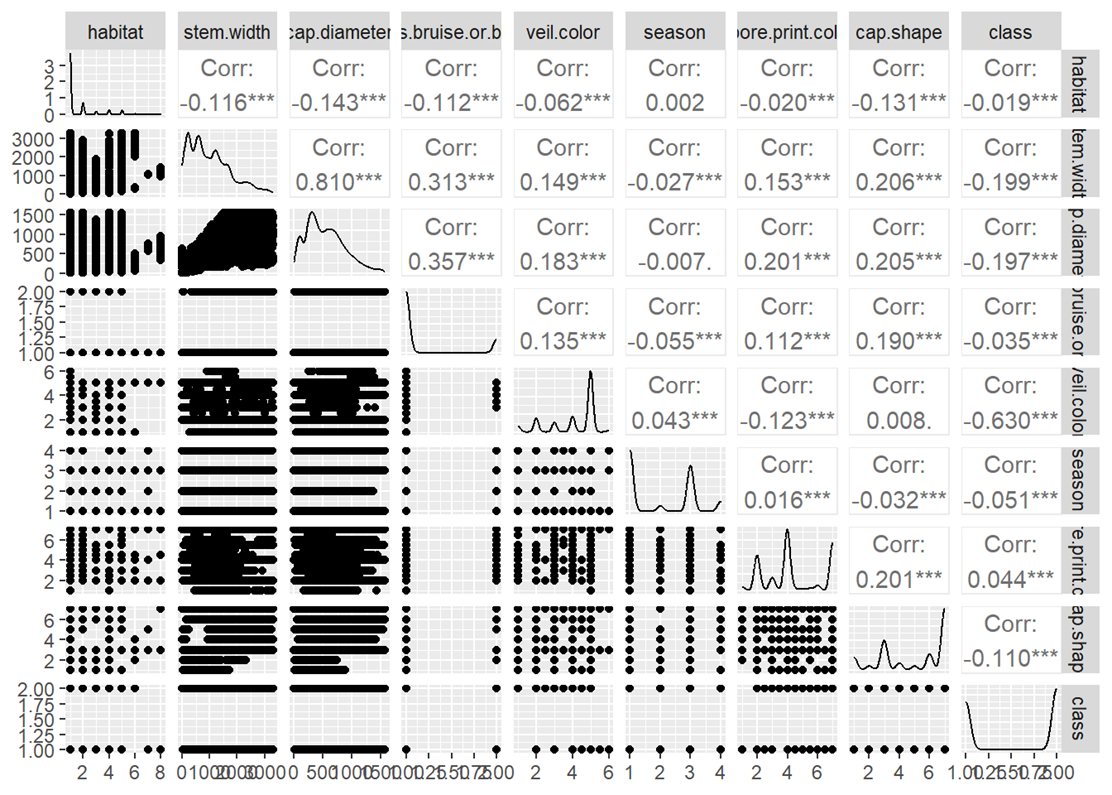
**2. Feature Selection and Reduction:**

Considering the above techniques:

Highly correlated features: Because "Cap surface" correlates very strongly with "gill color," this was removed.

Overfitting considerations: Removed "class.1e" and "class.1p" which might be dummy variables created for the target class during model training. Removing those after identifying model overfitting is a reasonable thing to do.

**3. Pair Metrics**



**Predictive Analytics:**

1**. Model Selection and Training**:

Two classification algorithms were tested:

K-Nearest Neighbors (KNN): This algorithm classifies data points based on the similarity to their nearest neighbors in the training data.

**Assumptions:**

* Tuning hyperparameter -k
* Use of a distance metric – In this project, I have used Euclidean distance metric
* Comparable Scale of metric – Normalized the data

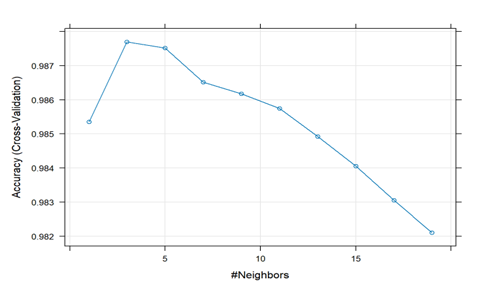
Naive Bayes: This probabilistic classifier predicts class membership based on Bayes' theorem and assumes independence between features.

**Assumptions:**

* Independence of features – since we have removed highly correlated variables and thus ensuring independence of features.

**2. Optimal K**

The k value is tested from k=1 to k=20 and found that k=3 gives optimal value



**3. KNN Model Optimization:**

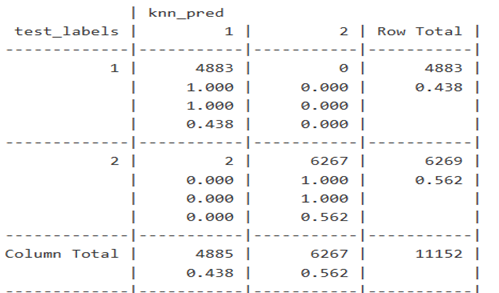
KNN performance is highly dependent on the value of "k," the number of nearest neighbors used in classification.

Normalization was likely done on the data to ensure that all features contributed equally to the distance calculations in KNN.

Through experimentation, the best "k" value was found to be 3. This means using the three nearest neighbors from the training data proved to have the most accurate predictions on unseen data.

The data was separated into training (80%) and testing (20%). The training set was used to build the models, and the testing set was used to evaluate their performance on unseen data.

KNN model built using "k" values of 3, 4, and 5 and the KNN model reached 99.87% accuracy. This means the model correctly classified nearly all of the mushrooms in the testing set. A confusion matrix was likely created to see how the model performed in terms of True Positives (correctly classified edible), True Negatives (correctly classified poisonous), False Positives (misclassified poisonous as edible), and False Negatives (misclassified edible as poisonous).

A white sheet of paper with numbers and lines

Description automatically generatedA white sheet of paper with numbers and lines

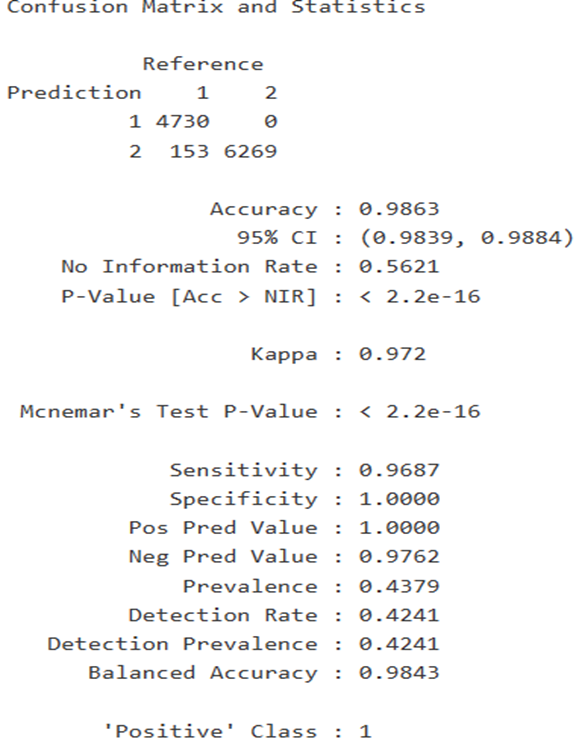
Description automatically generated

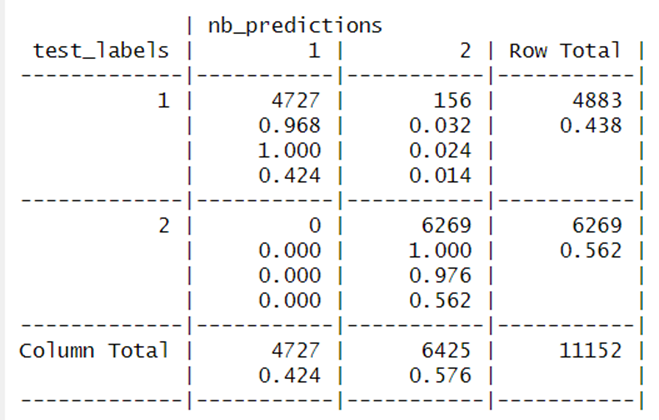
k =3 k = 4 k = 5

**4. Naive Bayes model:**

A Naive Bayes model was also trained and tested and the model achieved 98.63% accuracy. While this was not as high as KNN, it still suggests a good overall performance of classifying mushrooms.

A confusion matrix was likely made for the Naive Bayes model as well.

****



**Evaluating the Model:**

1. **Prevention of Overfitting and Cross-Validation**

The accuracy of 98.7 percent generated before was too good to be true for overfitting. To validate the results and accuracy, cross-validation is performed on both the models.

Cross-validation is a method where data is divided into k folds. In the case of k-fold cross-validation, k-5 of the folds are used for training the model and k-5 of the folds are used for testing the model. The process is then repeated k times—each time, one fold of the data is used for testing. That provides a more robust estimate of how well the model may perform on unseen data.

The reported accuracy of 98.7 percent was stable with k-fold cross-validation. That said, this indicates that the model can generalize to data not seen before and is not memorizing the training set.

**2. KNN Model Performance**:

The best model to use KNN is k=3, and the accuracy came out to be as high as 99.97 percent under cross-validation. This further proves the quality of the model.

The average accuracy across k values (3, 4, and 5) was consistently high, between 99.91 percent and 99.94 percent. This further proves that the KNN model was not overly sensitive to k across a reasonable range.

The KNN model was stable and dependable, proving it generalized well to new, similar data.

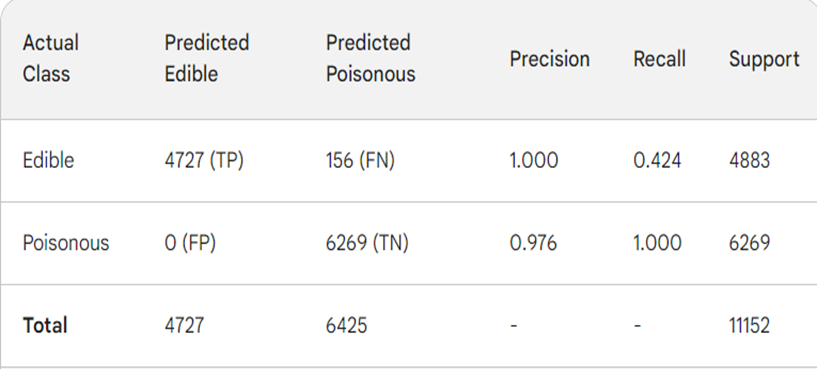
A screenshot of a computer

Description automatically generated

**3. Naive Bayes Model Performance**:

The Naive Bayes model also performed well under cross-validation with consistent high accuracy. This proves the fact that the model is robust and could handle the specific characteristics of the mushroom dataset.

The consistent performance across individual training and cross-validation metrics indicates the Naive Bayes classifier generalizes well toward new, similar data.

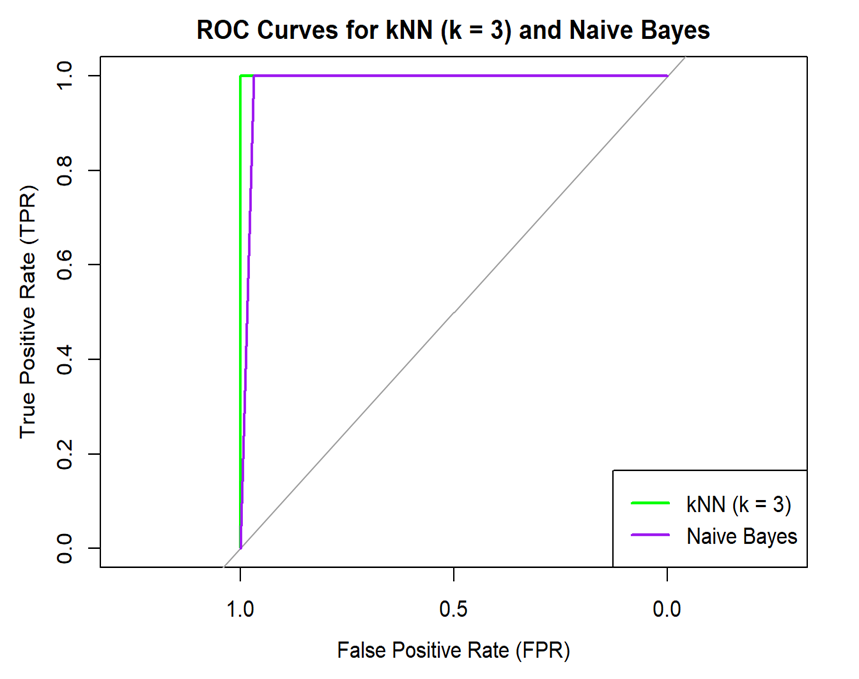


**4. ROC Curve Analysis**:

The ROC (Receiver Operating Characteristic) curves is used to better visualize the performance of both models.

Both the models, the KNN and Naive Bayes classifiers performed well under cross-validation, both having ROC curves close to the top-left corner. This implied that both.

While KNN has shown a slight edge in some points of the ROC curve, especially at lower thresholds, the overall performance of the two models was still very good.



**Conclusion:**

All in all, this analysis convinces quite well that KNN and Naive Bayes models achieved excellent performance in the classification of edible and poisonous mushrooms. It was confirmed through cross-validation that the high accuracy was not through overfitting. Furthermore, the ROC curves further supported the ability of the models in differentiating between the edible and poisonous classes.

KNN appeared the slightly better model with higher overall accuracy and a slight advantage in the initial part of the ROC curve.

The performance of both is quite strong, and they may be used in similar tasks depending on the requirements of a project and matters like model interpretability—Naive Bayes might be easier to interpret or computational efficiency—KNN can be computationally expensive for large datasets.

**Reference:**

**Data source**: <https://www.kaggle.com/datasets/vishalpnaik/mushroom-classification-edible-or-poisonous/data>.

**PCA:**

<http://www.sthda.com/english/articles/31-principal-component-methods-in-r-practical-guide/118-principal-component-analysis-in-r-prcomp-vs-princomp/#google_vignette>

**Pair Metrics:**

<https://www.geeksforgeeks.org/how-to-create-and-interpret-pairs-plots-in-r/>