**Automated Tool for Assisting Data Science Projects**

**Introduction to Data Management and Processing**

**DS 5110**

**Authors:**

**Simran Girdhar Kodwani**

**Crains Sudhirkumar Patel**

**Devarsh Harshad Bhupatkar**

**Madhunil Anil Pachghare**

Table of Contents

**Summary3**

**Methods4**

Type chapter title (level 2)5

Type chapter title (level 3)6

1. **Summary**

In any data science project, there are a few steps of exploring the data that are common regardless of the dataset or domain of the data for example finding if there are null values in the dataset, if there are outliers in any of the feature columns, et cetera, which a user has to do repeatedly for every project. This tool aims to automate a few processes like these for the users and also aid people who are new to the field of data science. Additionally, if a user wants to do a regression task, the tool can run linear regression on the dataset to give a baseline before more complex methods are tried.

Missing data is a common problem in practical data analysis. They are simply observations that we intend to make but did not. In datasets, missing values could be represented as ‘?’, ‘Nan’, ’NA’, blank cell, or sometimes ‘-999’, ’inf’, ‘-inf’. There are three main problems that missing data causes: missing data can introduce a substantial amount of bias, make the handling and analysis of the data more arduous, and create reductions in efficiency.

A basic strategy to use incomplete datasets is to discard entire rows and/or columns containing missing values. However, this comes at the price of losing data, which may be valuable (even though incomplete). A better strategy is to impute the missing values, i.e., to infer them from the known part of the data.

Real life datasets usually have a large number of variables. This can dramatically impact the performance of machine learning algorithms fit on data with many input features, generally referred to as the “curse of dimensionality”. Dimensionality Reduction helps with these problems while trying to preserve most of the relevant information in the data needed to learn accurate, predictive models.

In general, the project aims to automate the process of finding and imputing missing values, recommending the reduction of dimensions in a dataset with multiple features and also displaying a scree plot to the user, plotting graphs for exploratory data analysis and creating plots to find their relationships with respect to each other, et cetera.

There are multiple modules that have been created for this tool. To solve the problem of missing values and for the process of imputing them, functions like "IterativeImputer" from sklearn have been used. For reducing dimensions of the data through PCA, the user can select the maximum variance they want to capture and the number of dimensions infer how much minimum variance they want to capture after reducing the dimensions based on the scree plot. Exploratory data analyses of features have been automated depending on whether the variable is nominal or ordinal.

1. **Methods**
2. Programming Language: Python
3. Data Description:

To describe and analyze the data, an understanding of the nature of data is important as the type of data influences the type of analysis that has to be performed on it. The tool has an integrated pandas\_profiler. It gives the number of rows and columns in the dataset, the count of missing values in the dataset, the count of duplicate rows and the number of categorical and numerical features in the dataset.

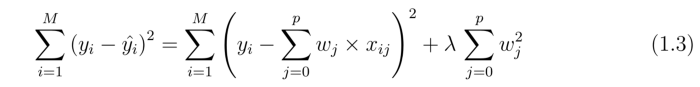
It provides summary statistics of all variables that include count of distinct values, percentage of unique values, mean values, minimum value in the column, range, standard deviation, coefficient of variation, sum, mean absolute deviation, skewness, variance, et cetera. Additionally, it plots a correlation heat map between different variables of the dataset.

1. Imputation of NA Values

For any categorical variable in the dataset, the NA values are imputed using the mode of the column. The column containing categorical data was then merged with the original dataset containing numeric columns after being converted to one hot vector. The get\_dummies function converts the categorical data column in the dataset to dummy numerical values which converts a categorical column with “m” unique values to “m-1” different features.

For any continuous variable in the dataset, the NA values are imputed using the IterativeImputer function from sklearn. The tool uses Ridge Regression for regression amongst the variables. It is a technique of estimating the coefficients of multiple-regression models where independent variables are highly correlated. It includes an L2 penalty to the loss function. L2 penalty equals the square of the magnitude of coefficients of the model. The coefficient of the model removes influence of certain variables.

A parameter called “lambda” controls the weight of the penalty to the ridge regression loss function.



M is the rows of the dataset, p is number of features and w is the coefficients of the ridge regression model. When lambda tends to 0, the model will resemble a linear regression model.

After the imputation of numeric columns, the categorical columns that were converted to numerical columns (after imputation using mode) using the get\_dummies function are converted back to categorical columns. The final output of this step is a dataset with all NA values in the columns (both categorical and numeric) imputed.

1. Exploratory Data Analysis

The main purpose of EDA is to look at data before making any assumptions. It can help identify errors, as well as better understand patterns within the dataset, determine outliers or anomalous events, and find interesting relations among the variables. The tool plots a histogram for all continuous variables. It plots a scatter plot for two continuous variables and a boxplot for continuous and categorical variables.

It plots a n\*n grid (where n is the number of variables) of plots where the plots in the diagonal of the grid are histograms and the upper triangle of the grid and lower triangle of the grid are scatter plots between the corresponding variables in the row and column.

1. Principal Component Analysis

Principal Component Analysis, or PCA, is a dimensionality-reduction technique, which is used to reduce the dimensions of large data sets. It does so by transforming a large set of variables into a smaller set such that it contains a considerable amount of the information. There is a tradeoff between accuracy and simplicity. Data sets with a small number of variables are easier to explore and visualize. This makes analyzing data much easier and faster for machine learning algorithms without large number of variables to process.

Principal components are new variables that are constructed as linear combinations of the initial variables. These linear combinations are done in such a way that the principal components are not correlated and most of the information within the initial variables is squeezed or compressed into the initial components.

Since the number of principal components in data is equal to the number of variables in the data, principal components are chosen in a manner that the initial principal component accounts for the largest possible variance in the data set. The second principal component is calculated with the context that it is uncorrelated with (i.e., perpendicular to) the first principal component and that it accounts for the next highest variance. This continues until a total of p principal components have been calculated, equal to the original number of variables.

The function find\_pca() in the tool performs PCA on the numeric data columns in the dataset. It plots a scree plot which shows the percentage of variance explained by the principal components.

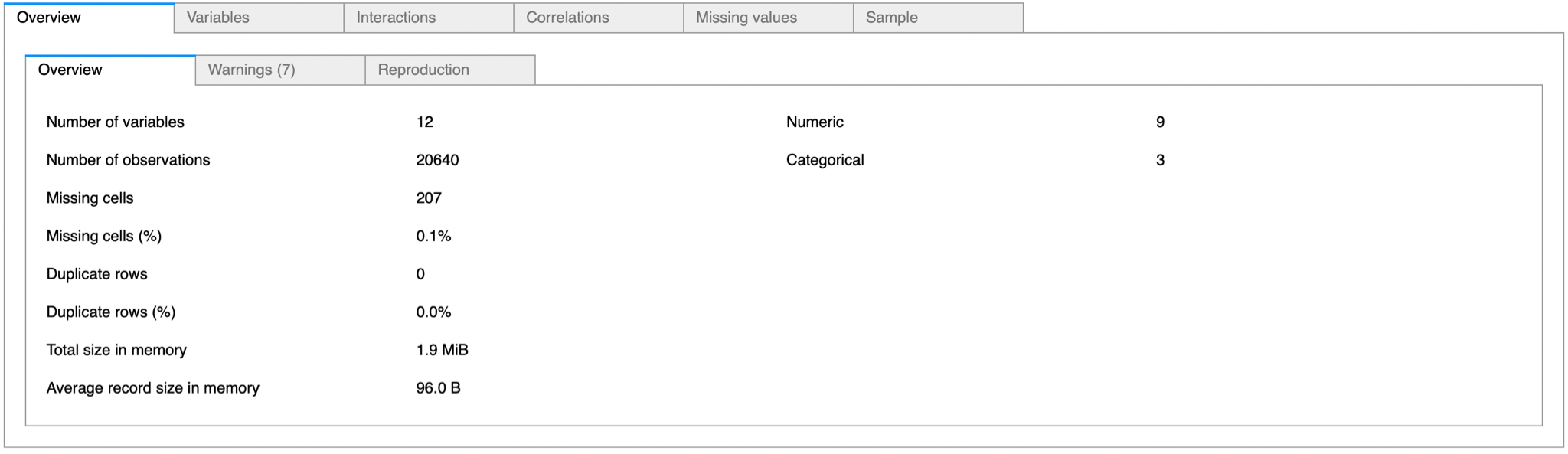
1. Linear Regression

Linear Regression is a predictive analysis technique. It is a linear approach to modeling the relationship between a scalar response and one or more explanatory variables. The forward\_selected() function in the tool performs regression. It splits the input dataset into training and validation sets. It divides into a 80% training set and 20% validation set. It selects the best model using a forward selection method, where the regression starts with no predictor and keeps adding a predictor. The process stops when the RMSE either increases at a subsequent step or the drop in the RMSE is not significant. The RMSEs are calculated on the validation set and the best model is selected. Linear regression gives a baseline for any regression task that a user wishes to perform.

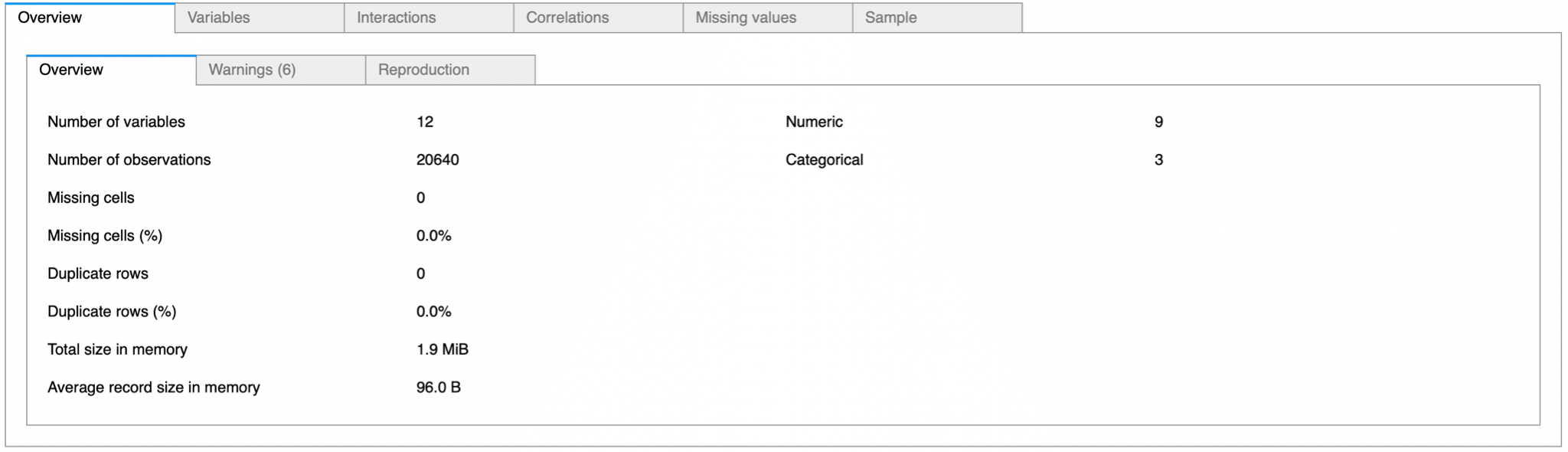
1. **Results**

The tool developed gives results for any dataset and tries to give a description of the dataset features, give a basic EDA by drawing graphs with all permutations of selecting 2 features, a scree plot after applying PCA on numeric values and a baseline for regression tasks by applying linear regression.

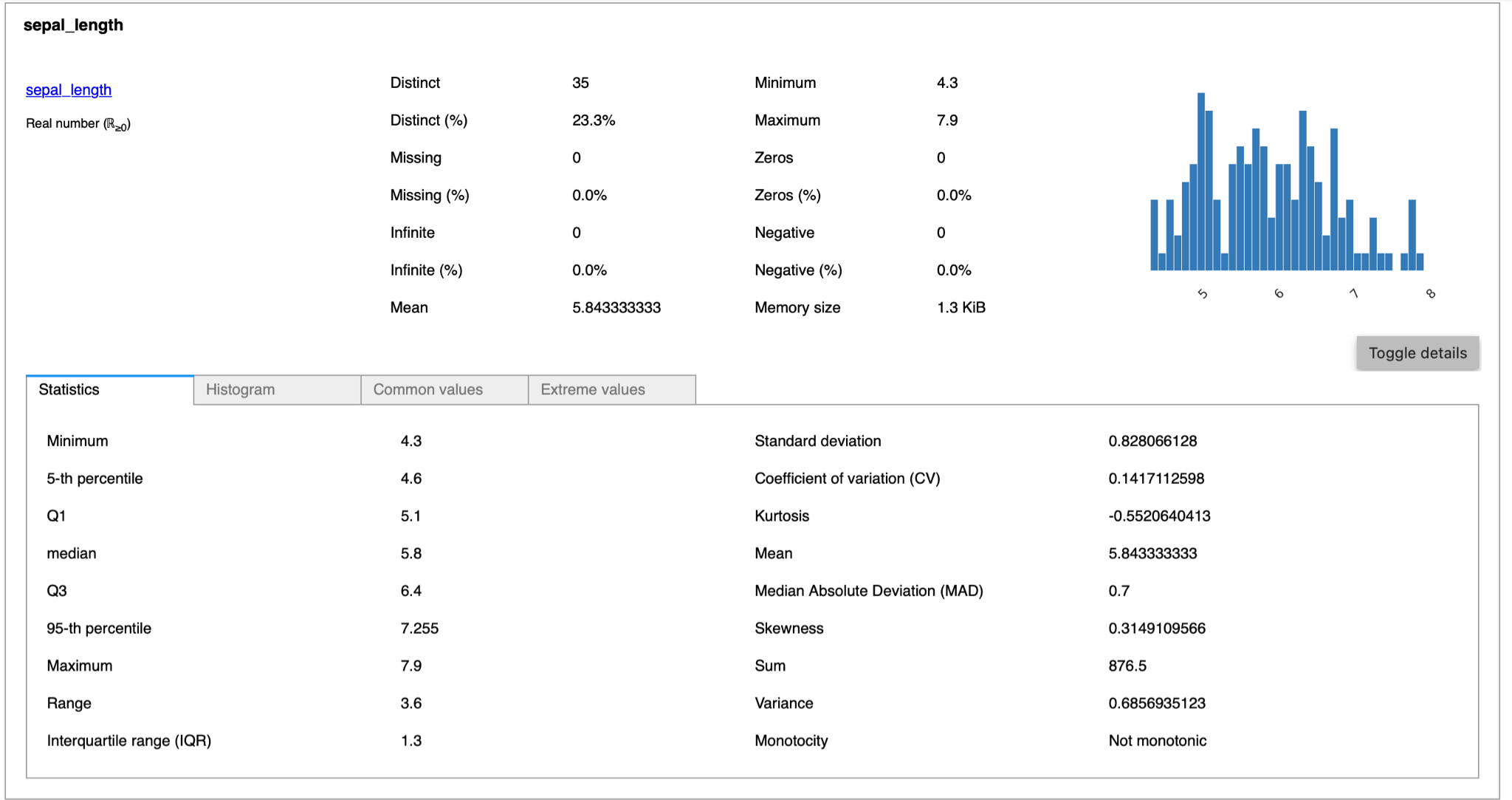
Pandas-profiler gives the description of the dataset and the variables in the dataset. The figure below gives the description of the number of missing values, the number of variables, number of numerical, categorical and boolean values.



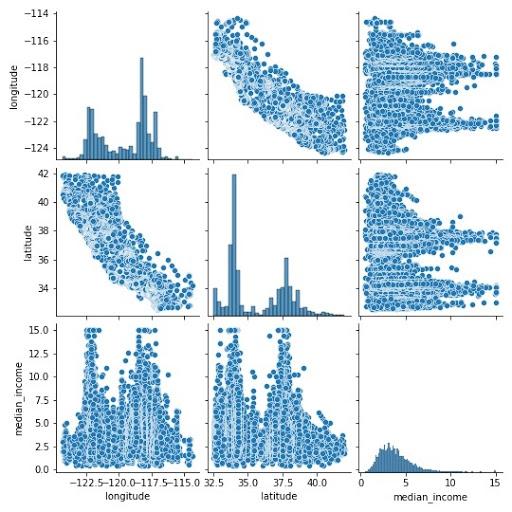
The figure below contains the output of the dataset after passing it through the iterative imputer module after the missing values are imputed.



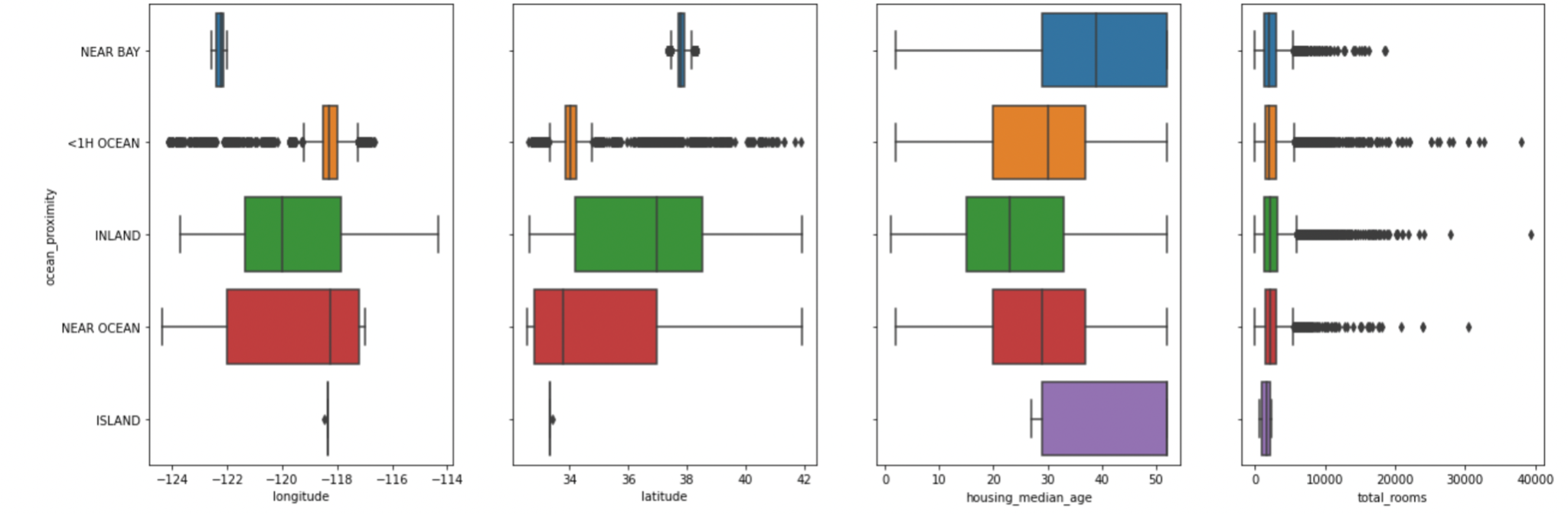
The figure below shows the description of the features. The statistics of the variables like minimum value, maximum value, the mean, sum, variance, range, quantile values, IQR, etcetera which allows the user to get an idea of the distribution and different statistics associated with all the individual features.



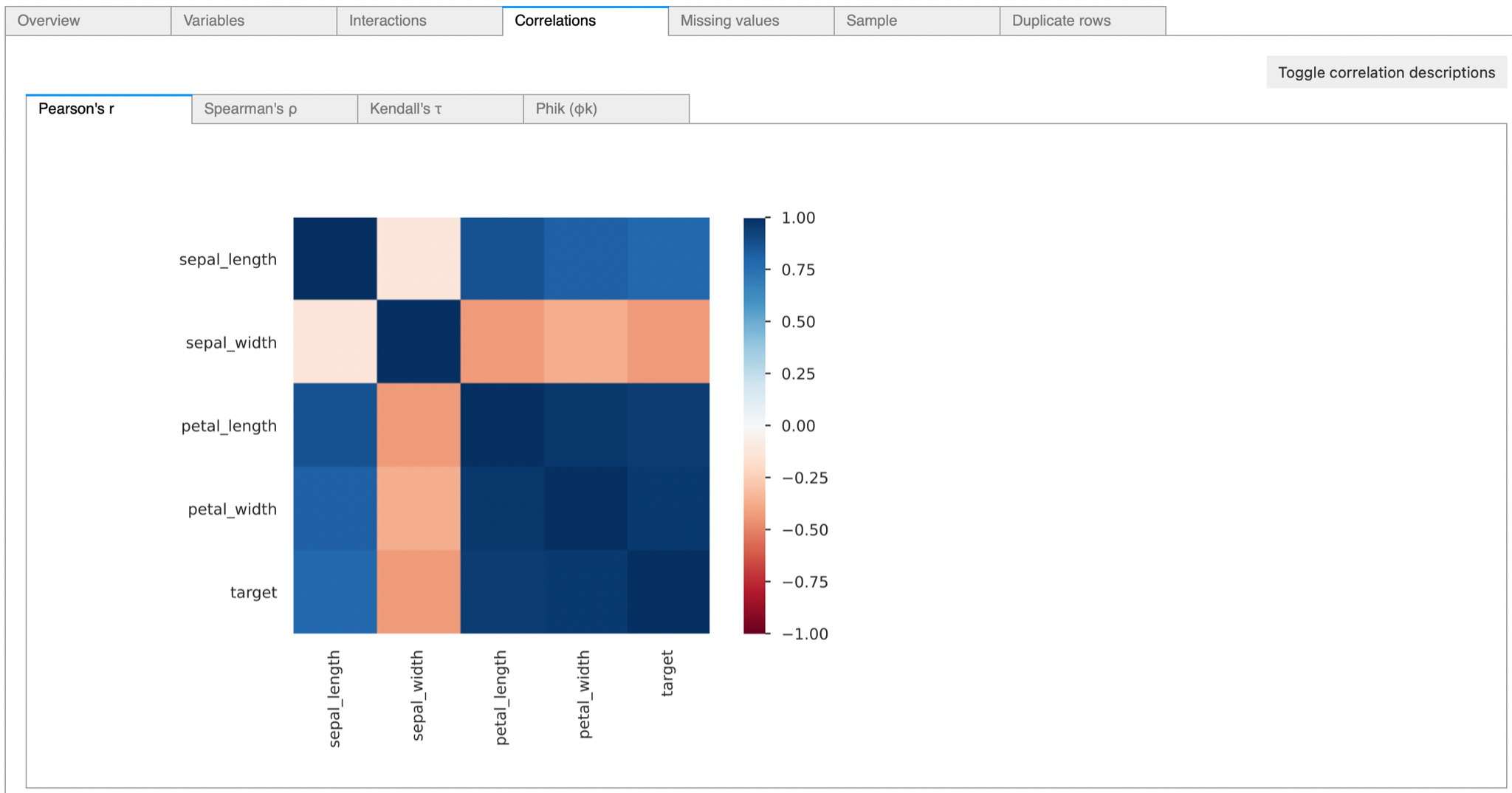
The figure shows every combination of numerical variables and histogram for each variable



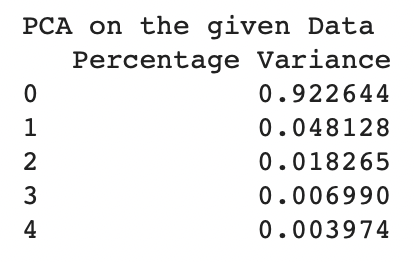
The figure below shows all combinations of categorical variables with all the continuous variables for the user to analyse on the given dataset.

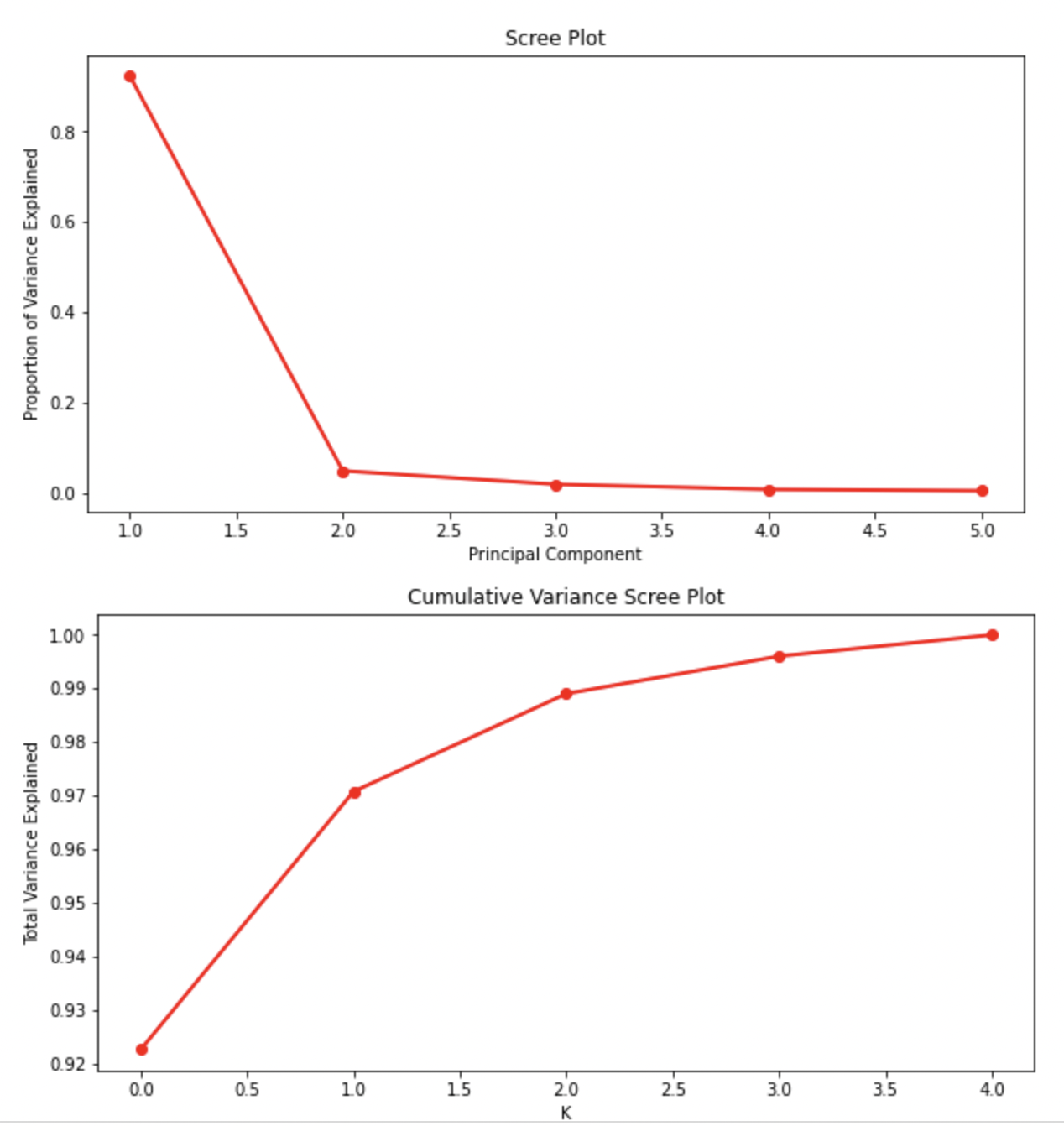


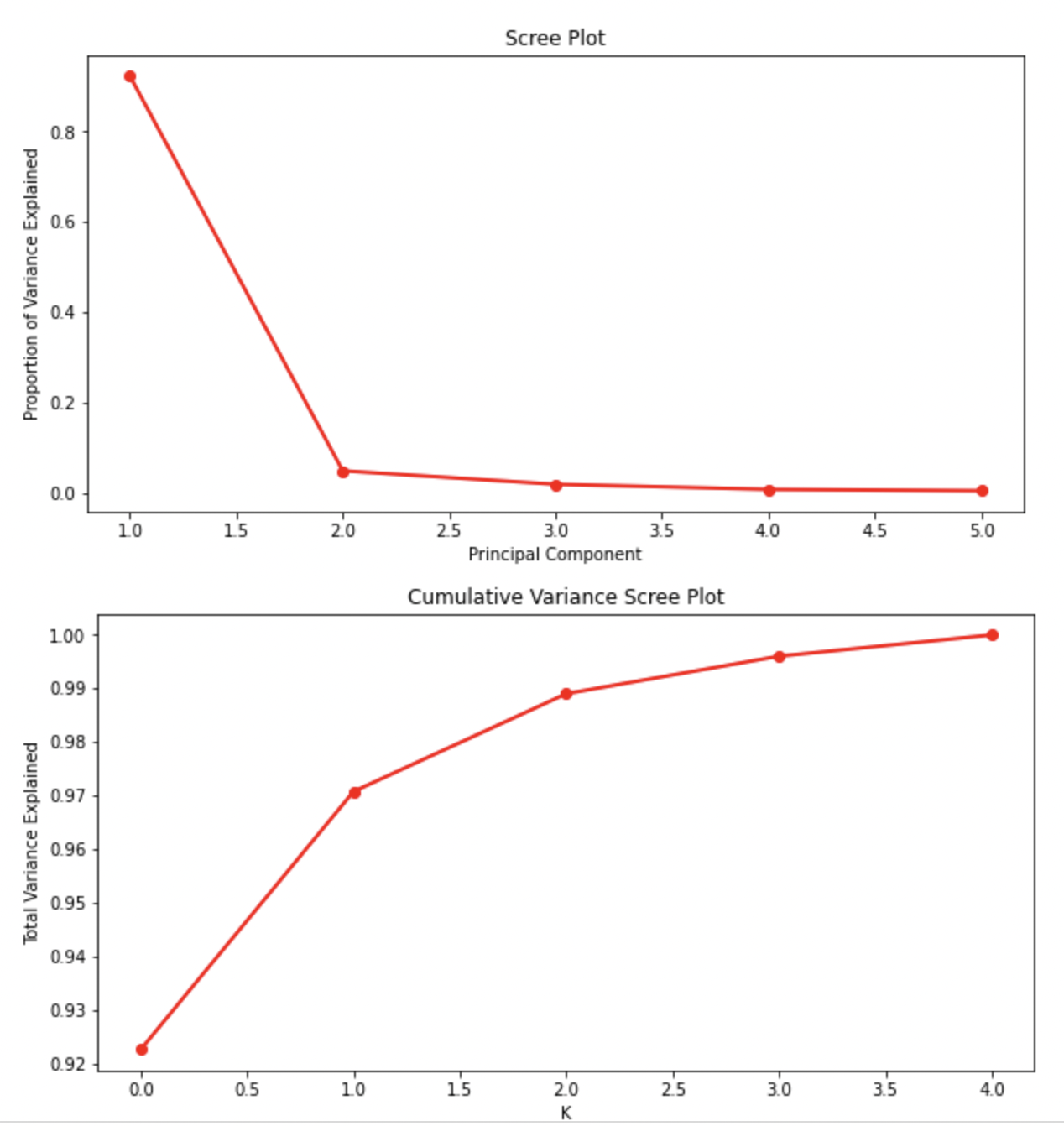
The image below show the correlation map between the variables present in the dataset entered by the user and gives a range of correlation matrix based on 4 correlation coefficients namely Pearson’s, Spearman’s, Kendall’s and ɸk.



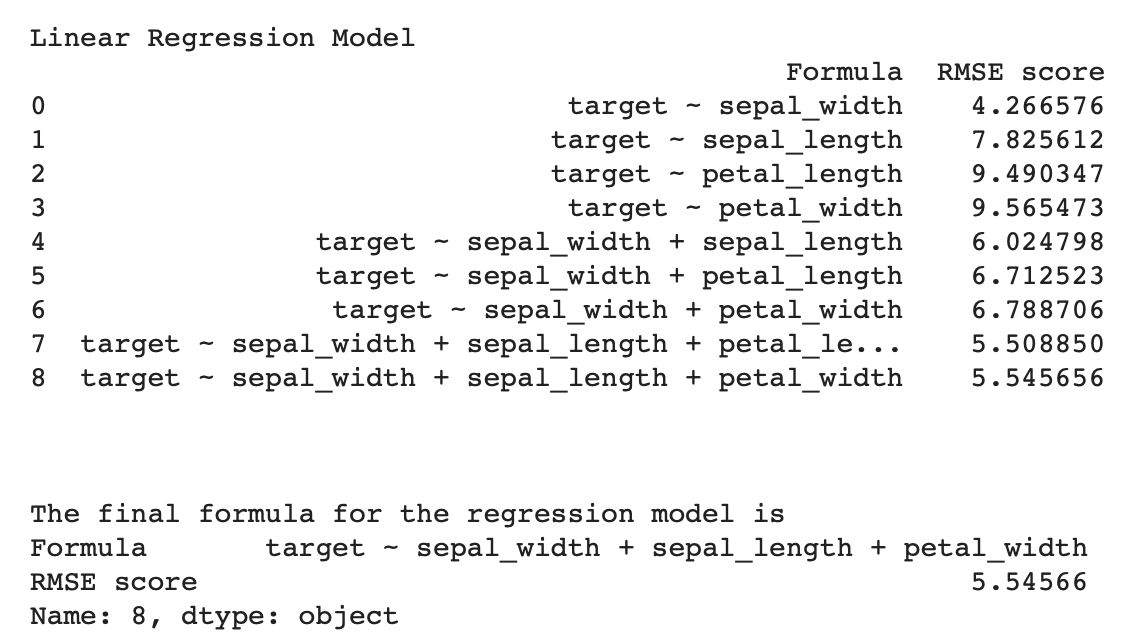
The below image is the output of the PCA function. It shows the percentage of variance captured in the principal component. For example the top principal component explains around 92% of variance while the second component explains 4% of the total variance and so on.



The below figure is the scree plot of the eigenvalues corresponding to the eigenvectors. The second plot shows the cumulative variance for each principal component.



The figure below shows the stepwise output of the linear regression function where the input feature which gives the least RMSE value on the validation set is selected till there is an increase in the RMSE value on including the next variable or the percentage drop in the RMSE value is not significant enough on selecting the next variable.



Due to the constraint of space and to clearly show the results of every part of the module, results from different datasets are shown in different modules so the reader can clearly understand every part of the tool.

1. **Discussion**

The project aims to help all Data Science enthusiasts and helps in reducing the workload of doing the same task repetitively which needs to be done in every Data Science pipeline irrespective of the domain and complexity of the project. The results in the initial analysis of the features done by the pandas profiler give accurate descriptions of the columns and given the amount of analysis given ranging from a basic understanding of the quantiles and mean value of the data to an understanding of the kurtosis and skewness of the individual variables.

These results would help anyone make better informed decisions as it gives information about the variables that a team might miss while doing exploratory data analysis. Additionally, all possible combinations of the variables are shown in a graph and more useful inferences can be taken by the team. Additionally, it is very important in any regression task to get a baseline for further results that need to be beaten. In order to do that, the tool will also get a baseline by doing a linear regression task on the dataset which is often ignored by many new Data Science enthusiasts.

There are many different things that can be done in the tool to make it more efficient for the user and allowing the user to manipulate it by adding hyperparameters that a user can manipulate. For example, in iterative imputer Ridge Regression is used in order to impute the values which could also be done using other regression techniques ranging from Linear Regression to Lasso Regression or a more advanced technique using Trees or Forests. Additionally, the imputer stops after 10 iterations in order for the tool to be computationally viable, the user could be given the option to change that if they want to wait for more iterations depending on the complexity of the results.

In the end, the tool still works only for regression tasks in the end which may not be the end goal of the user. The user might be interested in doing a classification task or even doing some sort of statistical inference on the data which could be covered as the future scope. Additionally, in a very high dimensional data, the linear regression task found using stepwise selection might take a lot of time and hence it limits the dimensionality of the data to an extent and for comparison even Ridge or Lasso regression could be used.

1. **Statement of Contribution**

**Pandas-Profiler**

Simran Girdhar Kodwani

Madhunil Anil Pachghare

**Exploratory Data Analysis**

Simran Girdhar Kodwani

Madhunil Anil Pachghare

**Automatic Imputer**

Crains Sudhirkumar Patel

Devarsh Harshad Bhupatkar

**PCA and Linear Regression**

Crains Sudhirkumar Patel

Devarsh Harshad Bhupatkar

1. **References**

**[1] “Pandas-Profiler : Generates profile reports from a pandas Dataframe”**

[**https://pandas-profiling.github.io/pandas-profiling/docs/master/rtd/**](https://pandas-profiling.github.io/pandas-profiling/docs/master/rtd/)

**[2] “sklearn.impute.IterativeImputer function ”**

[**https://scikit-learn.org/stable/modules/generated/sklearn.impute.IterativeImputer.html**](https://scikit-learn.org/stable/modules/generated/sklearn.impute.IterativeImputer.html)

**[3] “Stepwise linear regression”**

[**https://online.stat.psu.edu/stat501/lesson/10/10.2**](https://online.stat.psu.edu/stat501/lesson/10/10.2)

**[4] “Nian Shong Chok Pearson’s versus Spearman’s and Kendall’s Correlation coefficient for continuous Data”**

[**http://d-scholarship.pitt.edu/8056/1/Chokns\_etd2010.pdf**](http://d-scholarship.pitt.edu/8056/1/Chokns_etd2010.pdf)

**[5] “Sidharth Prasad Mishra\*, Uttam Sarkar, Subhash Taraphder, Sanjay Datta, Devi Prasanna Swain, Reshma Saikhom, Sasmita Panda and Menalsh Laishram Multivariate Statistical Data Analysis- Principal Component Analysis (PCA)”[available]**

**[6] “Ian T. Jolliffe, Jorge Cadima Principal component analysis: a review and recent developments”[available]**

**[7] “Mohammad Aljarrah, Yousef Al-Jarrah Using stepwise regression to investigate customers’ propensity to change cellular phone providers”[available]**

**[8] “John W. Emerson, Walton A. Green, Barret Schloerke, Jason Crowley, Dianne Cook, Heike Hofmann,Hadley Wickham The Generalized Pairs Plot”[available]**

1. **Appendix**

The github link to the implemented code and project is as follows:

“Project for Introduction to Data Management and Processing DS 5110”

[Automated Tool for Assisting Data Science Projects]

<https://github.com/crainspatel/DS5110>