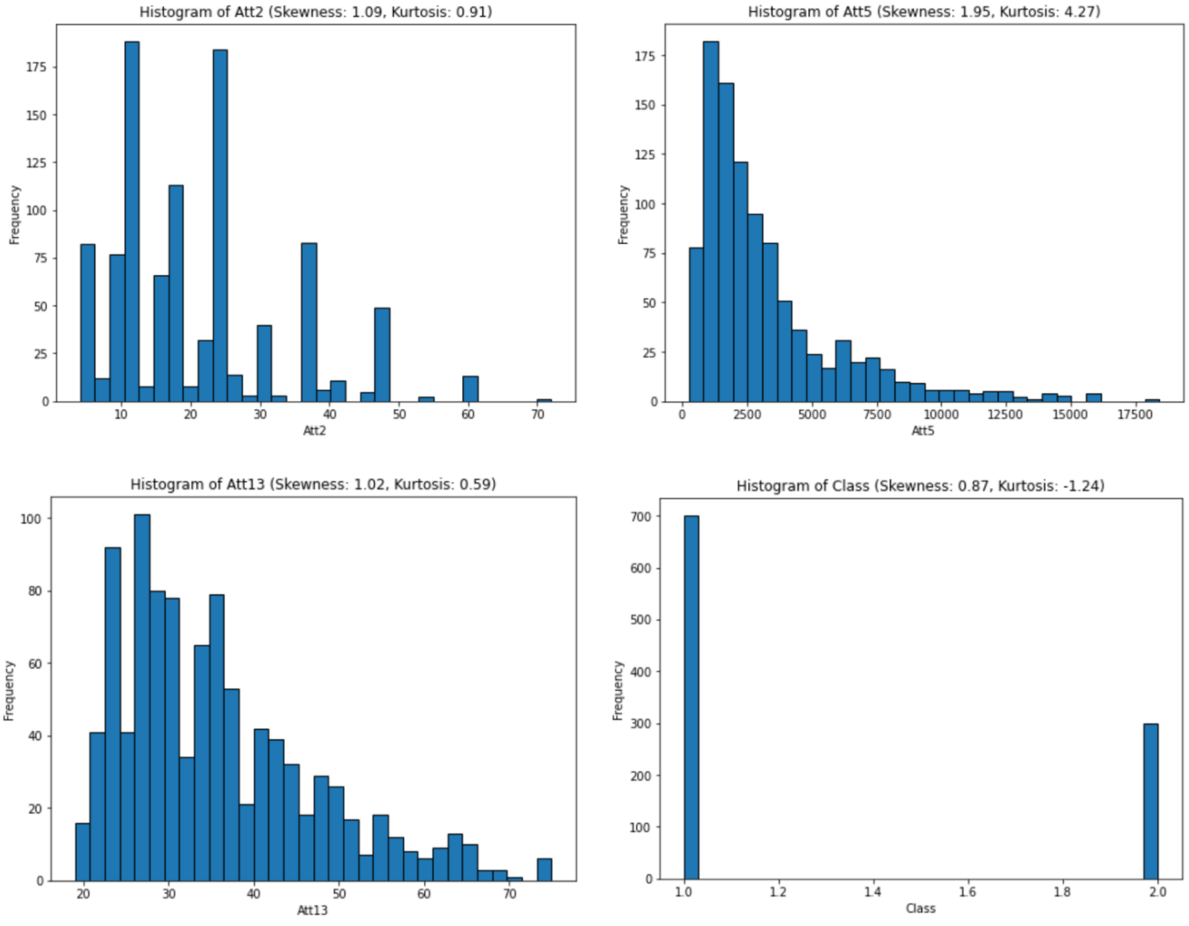
Data Understanding

1. Performing EDA

Within the dataset, there are 1000 instances and 21 features. Additionally, 13 of the features are categorical and 8 of the features are numerical. According to the Pearson correlation, the top three numerical features with the highest correlation to the target variable of credit risk are Att2, Att5, and Att13. Att2, representing the duration in months, shows a correlation coefficient of 0.2149 with credit risk. Att5, representing the credit amount, shows a correlation coefficient of 0.1547 with credit risk. Lastly, Att13, representing age in years, shows a correlation coefficient of -0.0911 with credit risk.

The square root rule was used to determine the number of bins used to draw the histograms below for Att2, Att5, Att13, and Class (credit score). The square root takes the square root of the total number of observations and rounds the value up to an integer. It is based on the idea that the number of bins should scale with the square root of the number of instances. Each feature/variable being plotted has 1000 instances and $\sqrt{1000} = 31.62$; therefore 32 bins were used for each histogram (*Adjusting the Number of Bins in a Histogram*, n.d.).



The data for feature Att2 (duration in months) seems to show not much of a pattern. Data is slightly skewed to the right (positive skewness). The skewness is 1.09 and the kurtosis is 0.91. This means that there is leptokurtic kurtosis, meaning there is a heavier tale, and outliers are more likely. The data for feature Att5 (credit amount) is skewed to the right, showing positive skewness. The skewness is 1.95, and the kurtosis is 4.27. This means that there is leptokurtic (positive) kurtosis, meaning outliers are more likely with the heavier tale. The data for feature Att13 (age in years) is slightly skewed to the right, showing positive skewness. The skewness

is 1.02, and the kurtosis is 0.59. Leptokurtic (positive) kurtosis is present, meaning outliers and extreme values are more likely with the heavier tale. The data for feature Att13 (age in years) is slightly skewed to the right, showing positive skewness. The skewness is 1.02, and the kurtosis is 0.59. Leptokurtic (positive) kurtosis is present, meaning outliers and extreme values are more likely with the heavier tale. The skewness for the class label "Class" is 0.87 (positive), and the kurtosis is -1.24. This means that platykurtic (negative) kurtosis is present. For this feature, there are only two possible class labels, 1 (good) or 2 (bad).

There 65 missing values for one feature in the dataset, being Att1 which represents the status of existing checking account. No other features in the dataset contain missing values. The percentage of missing values for the incomplete feature Att1 is 6.5%. This was calculated by $\frac{65}{1000} \times 100 = 6.5$.

2. Justification of the type of machine learning task

This is a classification task (a binary classification problem) because we are asked to predict the credit risk (good or bad) of a customer based on multiple features. We are given labelled data, and we are asked to classify data points to a class label (supervised machine learning). There are two possible class labels within the "Class" column of the dataset, where '1' represents good credit risk, and '2' represents bad credit risk. This machine learning task could not be a regression problem because the feature "Class" in the dataset is not a continuous outcome (y) and is not a numerical feature. Additionally, this machine learning task is not a clustering task because we are given labelled data, and a clustering task works with unlabelled data (unsupervised machine learning).

Data Preprocessing

To prevent data leakage, the dataset has been partitioned into a training set and a test set. The size of the test set has been set to 0.3, that is 30% of the dataset has been reserved for testing, and the other 70% will be used for training. The self-defined 'preprocess' function utilises the pipeline function to convert categorical data to numerical data, impute missing values for Att1 with the most frequent value, and lastly standardise numerical features. The column transformer does not standardise all data, as difficulties arose when trying to attempt that. After completing data preprocessing, X_train returns an array (later converted into a panda data frame) that has no categorical variables, no missing values, and standardised values for numerical features (as per the instructions that specified the order data preprocessing needed to be completed in). There are 50 features with the preprocessed dataset, and the number of observations remains the same with 700 rows, as no rows were removed.

To prepare data for machine learning algorithms, categorical data needs to be converted to numeric data. For the German credit dataset, some categorical variables are ordinal (follow a natural rank order), whereas some of them are nominal (no order). Ordinal Encoding is suitable for features Att1 (status of existing checking account), Att6 (savings account/bonds), and Att7 (present employment since), which are assumed to have a meaningful order. The reason why these variables are ordinal is because they have a natural order that increases. For example, Att7 has categories regarding time (in years). This means that these categorical features are converted to numeric values that preserve their ordering. In contrast, the remaining categorical attributes in the German credit dataset seem to follow no order (nominal features), therefore One Hot Encoding is used, and dummy variables are made for Att3, 4, 9, 10, 12, 14, 15, 17, 19 & 20.

Within the German credit dataset, there is only one feature that is missing data which is Att1, an ordinal categorical feature. Within the feature itself, 6.5% of rows are missing values. A deletion approach could have been taken, that is deleting the feature or deleting all rows that have a missing value. The feature Att1 is the status of existing checking account, and it could be a relevant feature in the dataset. Rows could be deleted, but that means important data from 65 other features may be deleted. Additionally, missing data may not have been introduced in the MCAR (missing at completely random) mode. That is, the likelihood of missing checking account data in a survey depends on the respondent's credit history, but not the actual checking account data itself. Because of this uncertainty, for the German credit dataset, it is best to impute missing values for Att1 with the most frequent along the column. To do this, SimpleImputer was used, thus preparing the data for classification models.

To ensure numerical features are on the same scale, they have been standardised using StandardScaler from sklearn. This scales the data to have a standard deviation of 1 and centres the data around 0. Scaling numeric features can help to improve the performance of machine learning models. Take note that once the categorical data was converted to numerical data, it was not scaled using standard scaler due to difficulties with the code. In future this would be done, to ensure all data was on the same scale. Looking at the data frame (X_train) however, it looks like the encoded categorical data is on a similar scale to the rest.

Feature Ranking

The top five features selected by the feature ranking method using sklearn.feature_selection.mutual_info_classif are indexed at 0, 47, 7, 25, and 21. Based on the attribute description text file and the appendix provided these values correspond to:

0 = Attribute 1: Status of existing checking account
47 = Attribute 8: Instalment rate in percentage of disposable income
7 = Attribute 3: credit history, A34: critical account/other credits existing (not at this bank)
25 = Attribute 10: Other debtors / guarantors, A102 : co-applicant
21 = Attribute 9: Personal status and sex, A93 : male : single

When using these top five features obtained through feature ranking, an accuracy score of 62.29% is obtained when performed on the test set. This is a greater accuracy compared to using all features, where an accuracy score of 59.47% is obtained when performed on the test set. Therefore, it is better to use the top five features than the original feature set because the accuracy score is higher for the top five features. The reason for this might be because the feature ranking function selects the most relevant features to the target variable. Using the top five features may also reduce the risk of overfitting and can help mitigate the curse of dimensionality. The curse of dimensionality occurs when the number of dimensions in a dataset increases, thus causing data points to become sparser in the high-dimensional space. Additionally, the elimination of redundant features through feature ranking reduces the noise in the data, allowing the model to focus on more meaningful patterns.

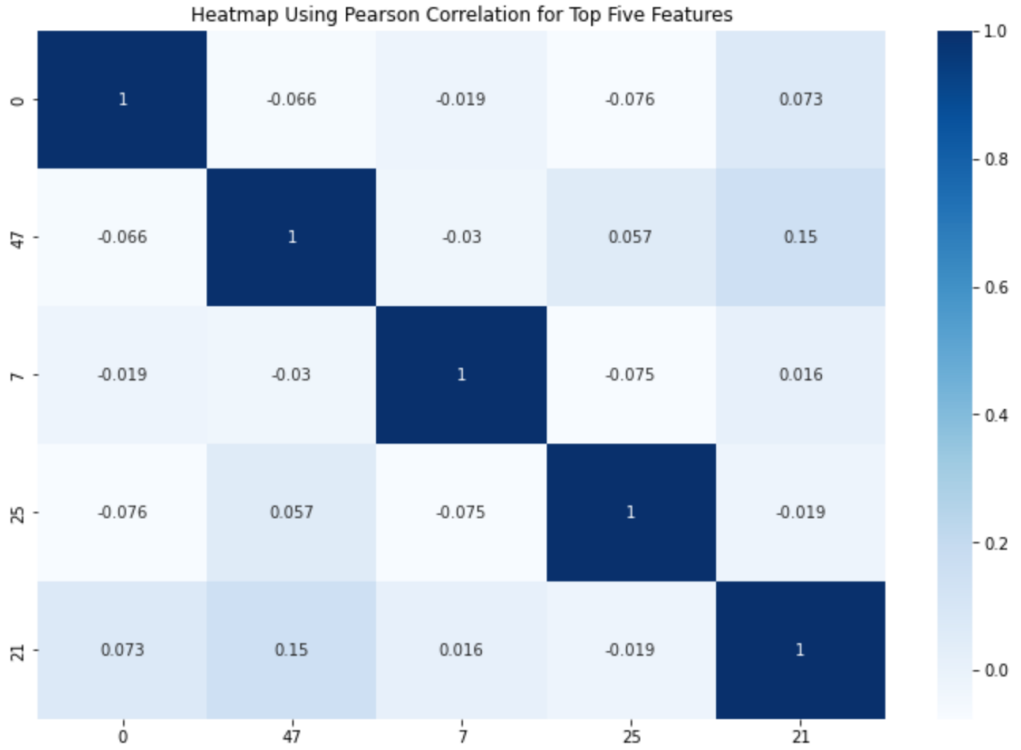


Figure 1: Heatmap showing the Pearson correlation between the top five features

The heatmap shows that there is not a strong correlation between any of the top five features, as there is a lot of light blue on the heat map. The strongest positive correlation is between 21 and 47, as a correlation coefficient of 0.15 is obtained. This indicates that as the value of 21 increases, the value of 47 also increases. The weakest correlation is between 7 and 0, where a negative correlation coefficient of -0.019 is obtained. The correlation between 47 and 0 is relatively weak and negative as it is -0.066, indicating that as the value of 47 increases, the value of 0 tends to decrease. The correlation of -0.076 between 25 and 0 indicates that there is a negative correlation, where when the value of 25 increases, 0 tends to decrease too. In contrast, as the value of 21 increases, the value of 0 also increases as there is a positive yet relatively weak correlation of 0.073 between these attributes. Due to there being no moderate or strong correlations between features, there is no evidence of moderate to strong positive or negative linear relationships between features.

Sequential Forward Feature Selection

The implemented SFFS algorithm uses a wrapper feature selection approach because the wrapper method evaluates subsets of feature performance using the k-nearest-neighbors classifier. In this case, the wrapper method determines the best subset of features based on the KNN model's performance. The ML algorithm (KNN in this example) is trained many times because it is fitted for each iteration of feature subsets created by adding the remaining features to the selected feature subset. The ML algorithm is defined by the ‘clf’ parameter in the sequential_score function, and in the sequential_feature_selection function it specifies ‘clf’ to be a KNN classifier with 3 neighbors. The method selects features through training and validation, and it selects them based on their performance when combined with features that have already been selected. This will continue until there is the same number of selected features as set in the parameter ‘no_features’ of the sequential_feature_selection function. To justify further, the SFFS algorithm does not use a filter method because it uses a KNN classifier, and filter uses no learning algorithm. Additionally, KNN cannot use the embedded method as

it cannot do embedded feature selection, showing that this SFFS algorithm uses a wrapper feature selection approach.

The implemented SFFS algorithm is a feature subset selection approach. This is because it evaluates the whole feature subset through an iterative process to improve the feature subset. It does not evaluate performance based on individual features like a feature ranking approach does and is therefore a feature subset selection approach.

When the number of selected features d is set to 5, the top five features selected by the sequential selection algorithm are indexed at 25, 46, 11, 6, and 8. These values correspond to:

25 = Attribute 10: Other debtors / guarantors, A102 : co-applicant
46 = Attribute 5: Credit amount
11 = Attribute 4: Purpose, A43 : radio/television
6 = Attribute 3: Credit history, A33 : delay in paying off in the past
8 = Attribute 4: Purpose, A40: car (new)

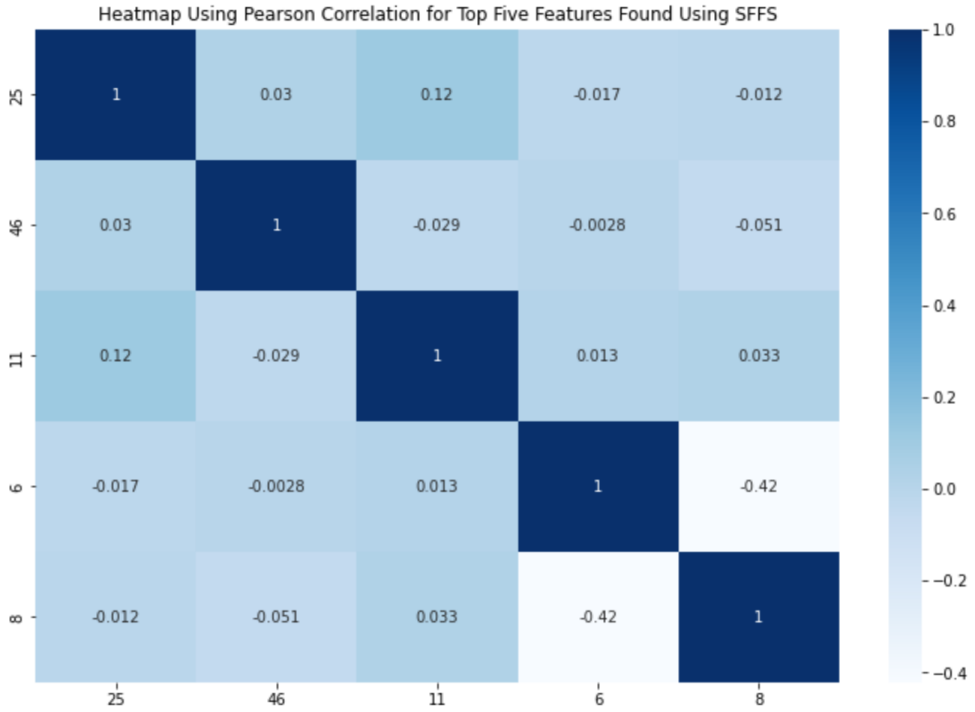


Figure 2: Heatmap showing the correlations between the top five features found using SFFS

Based on Pearson's correlation coefficients, the strongest correlation is between 11 and 25 as a correlation coefficient of 0.12 is obtained. The weakest correlation is between 6 and 46, as a correlation of -0.0028 is obtained. There is a positive correlation between 8 and 11, and 6 and 11, where correlation coefficients of 0.033 and 0.013 are obtained respectively. This indicates that as values of one of the features increase, values for the other feature tend to increase as well. Between 8 and 6 there is a strong negative correlation of -0.42. Similarly, there are negative correlations between 11 and 46 (-0.029), 6 and 25 (-0.017), and 8 and 25 (-0.012). A negative correlation suggests that as the value of one feature decreases, the value of the other feature decreases too. The 0.03 correlation between 46 and 25 suggests there is no strong linear relationship between these features due to the correlation being close to zero. Additionally,

there seems to be no strong linear relationship between 6 and 46 because a correlation coefficient of -0.0028 is obtained.

The testing performance of the implemented SFFS algorithm obtains an accuracy of 51.55%, whereas the feature ranking algorithm obtains an accuracy of 62.29%. This reflects how the SFFS algorithm performs better than the feature ranking algorithm does. The reason why the feature ranking algorithm performs better is because mutual information measures the reduction in uncertainty for one variable given a known value of the other variable, it measures mutual dependency. For feature ranking, mutual information can help identify features that are highly relevant to the target variable. Features with high mutual information are likely to have a strong relationship with the target variable (class). The SFFS algorithm may not capture the full extent of the relationships between features and the target variable. Another reason why the accuracy may be worse for the SFFS algorithm is that the optimal subset of features may be a large number of features that is not 5.

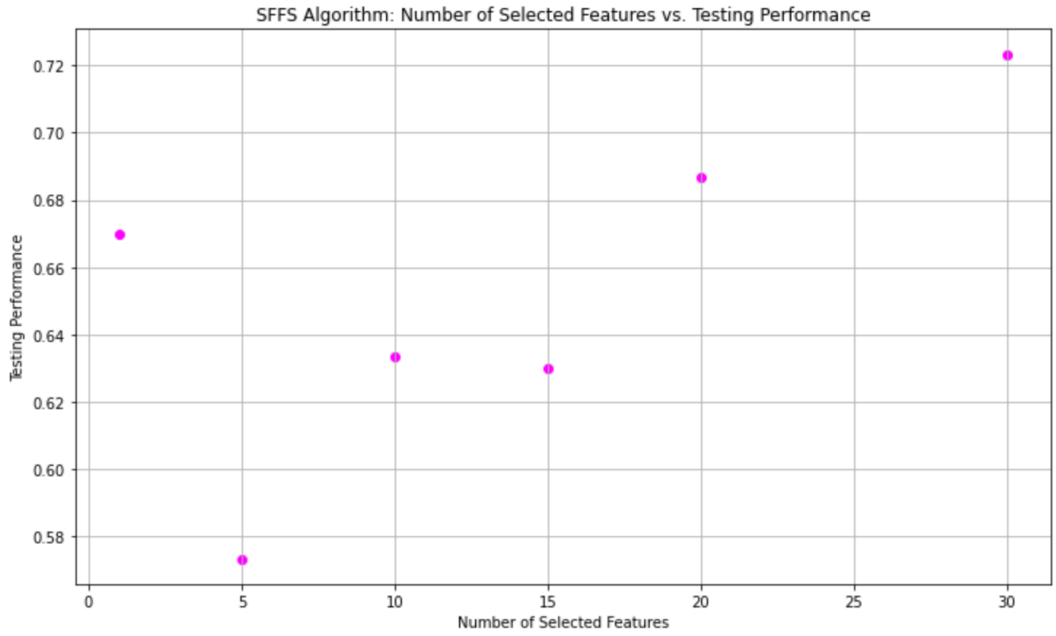


Figure 3: Scatterplot visualising the relationship between the number of selected features and testing performance

The testing performance changes depending on the number of selected features there are for the SFFS algorithm. The lowest performance is obtained when the number of selected features is set to five, and the best performance is obtained when the number of selected features is set to 30. Based on the scatterplot, there could potentially be a trend identified, but a greater number of testing performances at different selected features would have to be looked at to confirm this. From this scatterplot, it looks like at a higher number of selected features there is a higher testing performance. This scatterplot could be showing a positive relationship between testing performance and the number of selected features. With the number of selected features being 1, the test performance is around 67%, when selected features increase to 5, the testing performance decreases below 58%. At 10 selected features the testing performance increases again, then slightly decreases when selected features is set to 15. The testing performance then increases when the number of selected features increases from 15 to 20, then from 20 to 30. This shows that the optimal subset for this data might be when the number of selected features is larger.

Appendix for feature indexes

0. Att1(ordinal feature)
1. Att6 (ordinal feature)
2. Att7 (ordinal feature)
3. A30 : no credits taken/ all credits paid back duly (nominal feature)
4. A31 : all credits at this bank paid back duly (nominal feature)
5. A32 : existing credits paid back duly till now (nominal feature)
6. A33 : delay in paying off in the past (nominal feature)
7. A34 : critical account/other credits existing (not at this bank) (nominal feature)
8. A40 : car (new) (nominal feature)
9. A41 : car (used) (nominal feature)
10. A42 : furniture/equipment (nominal feature)
11. A43 : radio/television (nominal feature)
12. A44 : domestic appliances (nominal feature)
13. A45 : repairs (nominal feature)
14. A46 : education (nominal feature)
15. A47 : (vacation - does not exist?) (nominal feature)
16. A48 : retraining (nominal feature)
17. A49 : business (nominal feature)
18. A410 : others (nominal feature)
19. A91 : male : divorced/separated (nominal feature)
20. A92 : female : divorced/separated/married (nominal feature)
21. A93 : male : single (nominal feature)
22. A94 : male : married/widowed (nominal feature)
23. A95 : female : single (nominal feature)
24. A101 : none (nominal feature)
25. A102 : co-applicant (nominal feature)
26. A103 : guarantor (nominal feature)
27. A121 : real estate (nominal feature)
28. A122 : if not A121 : building society savings agreement/life insurance (nominal feature)
29. A123 : if not A121/A122 : car or other, not in attribute 6 (nominal feature)
30. A124 : no property (nominal feature)
31. A141 : bank (nominal feature)
32. A142 : stores (nominal feature)
33. A143 : none (nominal feature)
34. A151 : rent (nominal feature)
35. A152 : own (nominal feature)
36. A153 : for free (nominal feature)
37. A171 : unemployed/ unskilled - non-resident (nominal feature)
38. A172 : unskilled – resident (nominal feature)
39. A173 : skilled employee / official (nominal feature)
40. A174 : management/self-employed/highly qualified employee/officer (nominal feature)
41. A191 : none (nominal feature)
42. A192 : yes, registered under the customers name (nominal feature)
43. A201 : yes (nominal feature)
44. A202 : no (nominal feature)
45. Att2 (numerical feature)
46. Att5 (numerical feature)
47. Att8 (numerical feature)
48. Att11 (numerical feature)
49. Att13 (numerical feature)
50. Att16 (numerical feature)
51. Att18 (numerical feature)

References

Adjusting the number of bins in a histogram. (n.d.). Campus.datacamp.com. Retrieved April 9, 2024, from https://campus.datacamp.com/courses/statistical-thinking-in-python-part-1/graphical-exploratory-data-analysis?ex=7#:~:text=The%20%22square%20root%20rule%22%20is