Fundamentals of data analytics assignment 3

Data analytics in action

Brilliant Jepkogei Kiptoo

24699314

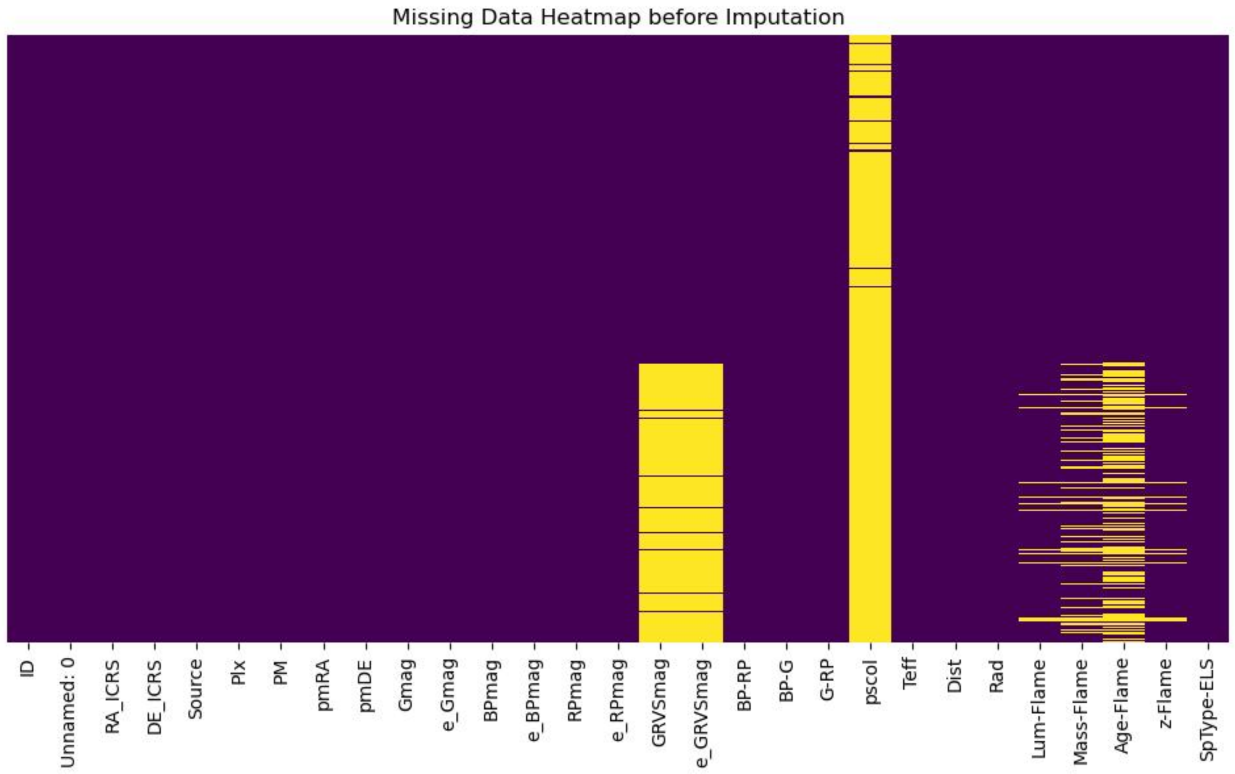
Data Mining Problem Description

The data mining problem in focus here is to create appropriate classifiers in order to estimate the "SpType-ELS" attribute of the dataset which contains measurements and parameters of observations of stars by the Gaia space observatory. Specifically, the classification task aims to categorize each star instance accurately into one of two classes A and B.

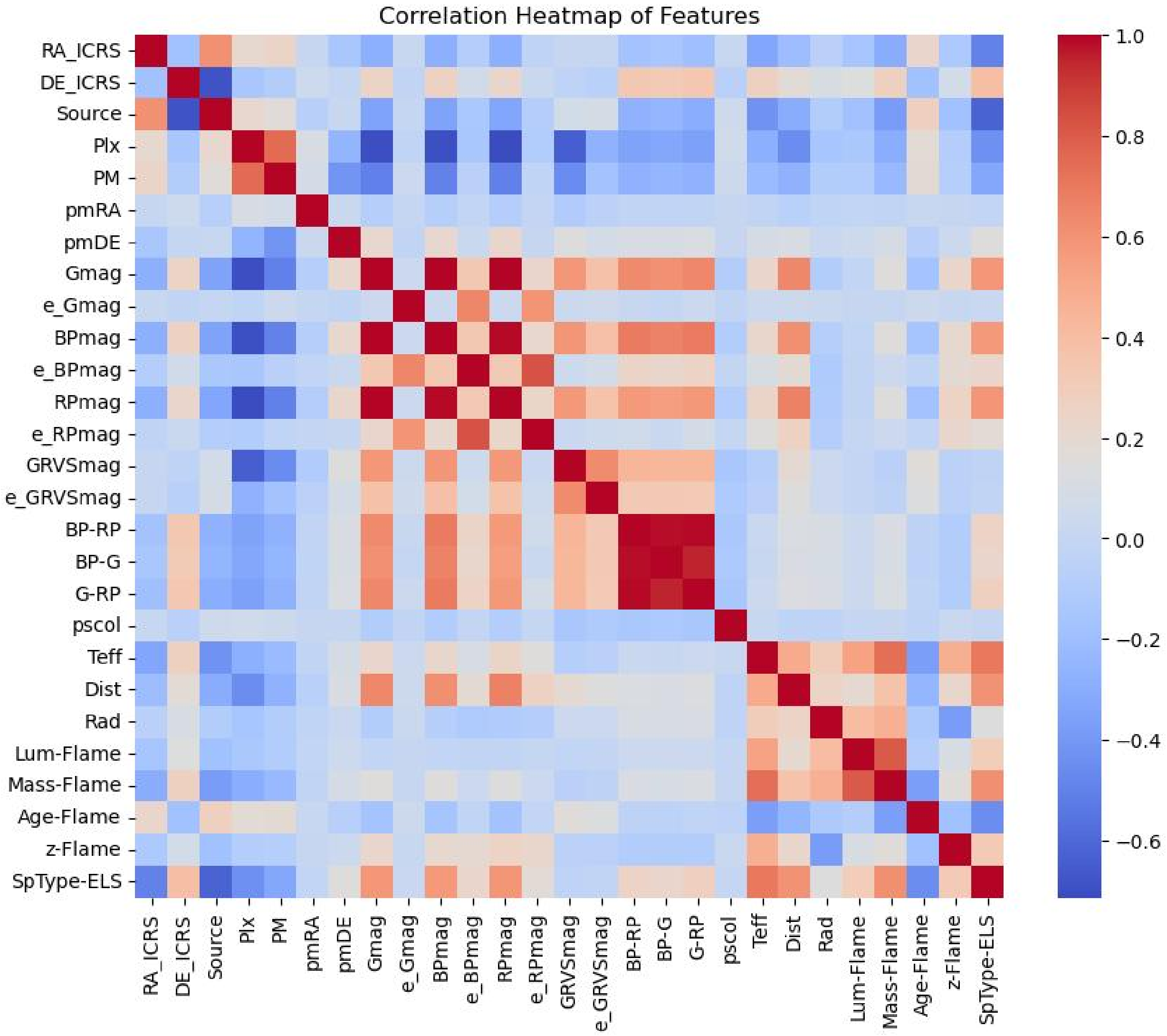
Spectral classification of stars is actually a very important issue within contemporary astronomy, since the primary aim of stellar classification is in fact its usage for the determination of stars' physical parameters, their stages of evolution as well as analyzing the universe, galaxies and other celestial bodies' composition and evolution. In particular, the Gaia mission sponsored by European space agency is a scientific project whose primary goal is to provide a virtual 3D edition of our galaxy and the Milky Way along with observing performances, including position and motions of billions of stars as well. Perhaps, the most critical and elementary achievement for promising an astronomer of reaching his ambitious goal is the precision of stellar classification by their spectral types.

Data Preprocessing and Transformations

Prior to creating the model, various data preprocessing steps were carried out to guarantee the data's quality and appropriateness for the classification task. Initially, there were missing values for various features in the training dataset, so they were addressed by replacing the missing numerical values with the mean of each column. Categorical columns were kept unchanged to maintain their integrity. The columns 'ID' and 'Unnamed: 0' were deemed unnecessary for classification and were removed from the dataset. In order to simplify the modeling, the target variable 'SpType-ELS' was encoded using LabelEncoder from scikit-learn, converting the categorical labels 'A' and 'B' into numeric values 0 and 1.



A number of visualizations were made in order to aid in understanding the dataset and its features. To make sure both classes were fairly represented, the target variable's distribution was examined. To find columns with missing data before imputation, a heatmap was created. A correlation heatmap was used to investigate feature correlations and offer insights into possible links between various features.



Histograms with kernel density estimates were also used to examine the distributions of a few chosen numerical features. This helped spot any potential outliers or skewed distributions that would need more preprocessing. The distribution plots were useful in helping to understand ‘shape’ or the nature of distribution of the chosen features in the dataset. In a somewhat similar manner, the histograms with kernel density estimates demonstrated that the distributions of some of the numerical were be unimodal and multimodal type.

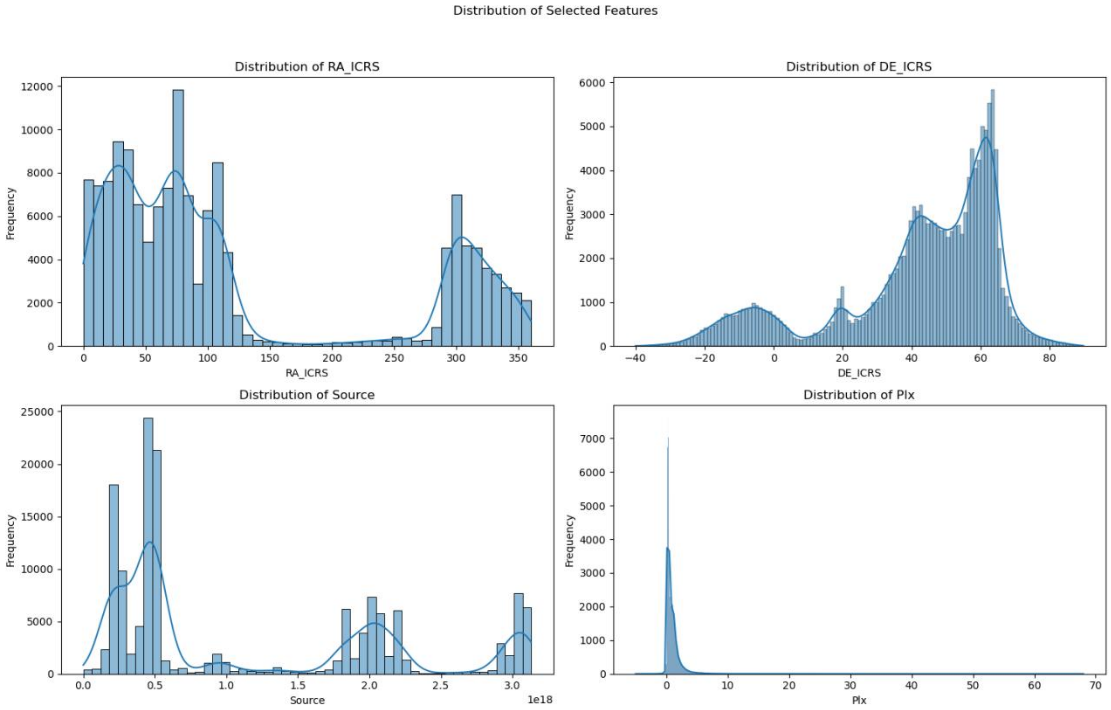
For example, the ‘RA_ICRS’ feature describes the right symmetry and , which may take a variety of values, and the calculated distribution appeared to be bimodal with a number of strong peaks and high variability across the range. So, there is an idea that there could be additional subgroups or clusters the data was not initially split into, which can be further analyzed for features or data splitting.

While the ‘DE_ICRS’ feature had again identified more domains, the distribution pattern of the scores were relatively different from that of the ‘DE_IPs’ Its score distribution displayed a more condense and skewed distribution with only one most frequent score, the rest having a smaller peak. This suggests that the distribution of the data points is relatively less spread and might not have extreme outliers or skewed towards a particular value as compared to other features in the dataset.

The distribution of the data points using the ‘Source’ feature was significantly right-skewed, meaning that the majority of the data points were concentrated towards values that are relatively small, while there was a slow decay towards high values. Sometimes these distortions may have relatively more observations in one tail than the other or very extreme values that may influence the results; such distortions may need transformation or use of robust techniques.

Finally, the ‘Plx’ feature depicted quite a Gaussian distribution, which had a bell-shaped figure with an equal number of observation values above and below the mean. This means that if this hypothesis is true, then the modeling assumptions based on the lateral distribution of the data points around the store’s central tendency would be less useful because the points in x2 feature should lay around this value.

These distribution plots support the fact that one should never neglect spending time learning about the given features and what might be hidden into them prior to actually applying any model. High variability in data, the presence of multiple variables, or outlier values might require further data pre-processing, feature engineering, or selecting the right model to fit these types of data and produce more accurate estimates and predictions.



Following the required preprocessing stages, an 80:20 split of the training data was made into training and validation sets, with stratification used to preserve the class balance in each set. This division avoided possible overfitting during training and enabled thorough model evaluation. Lastly, StandardScaler from scikit-learn was applied to the training and validation data to guarantee uniform scaling across all features.

Approach

The problem was approached using a systematic and comprehensive methodology, involving the following steps:

Firstly, four different classification algorithms were trained on the scaled training data: Logistic Regression, Decision Tree, K-Nearest Neighbors (KNN), Naïve Bayes, and Support Vector Machine (SVM). These were the models that were used as reference points, for tracking the efficiency of the techniques on the given data set.

The trained model was further tested on the validation set through the use of accuracy, precision, recall, and F1 scores in addition to confusion matrices. Thus, this evaluation allowed us to look at the strengths and weaknesses of each model and where we can see the opportunities for improvement, or where more research and development should be conducted.

According to the evaluation metrics used the model with the best results proved to be the Random Forest classifier this model was able to get the highest accuracy level on the test set. Therefore, leaving behind this model for the purpose of prediction on the unknown dataset as the final classifier.

Given this step-by-step approach that simply involved training, assessing and choosing the best fitting classifiers with high accuracy, you managed to find the apt classification algorithm to use based on the given problem and data set.

Classification Techniques and Results

The methods used in this project were Logistic Regression, Decision Tree, Support Vector Machine (SVM), and Random Forest for classification. All the models did exceptionally well on the validation set, with the Random Forest classifier achieving the top accuracy.

The Logistic Regression model obtained an accuracy of 99.34% in validation, with precision, recall, and F1-score of 0.99 for each class. The Decision Tree model demonstrated slightly better performance, achieving a validation accuracy of 99.60% and demonstrating perfect precision, recall, and F1-score for Class 0, while also attaining near-perfect scores for Class 1. The SVM model reached a validation accuracy of 99.57%, showing a precision of 0.99 for Class 0 and 1.00 for Class 1, and a recall of 1.00 for Class 0 and 0.99 for Class 1.

The Random Forest model, which merges several decision trees and uses their combined predictive ability, attained the top validation accuracy of 99.69%.

In order to further examine the effectiveness of each model, confusion matrices were created, visually displaying the accurate and inaccurate predictions of every classifier.

Best Classifier-Random Forest

Out of the models that were assessed, theThe Random Forest classifier was selected as the best model for addressing the specified classification issue because of its exceptional performance and numerous inherent advantages. Initially, it reached a test set accuracy of 99.75%, surpassing other classifiers by a slight but notable margin.

Random Forests are recognized for their ability to withstand noise and are less susceptible to overfitting in comparison to single decision trees. The strength of the model comes from combining predictions from various trees, lessening the influence of biases in each tree and leading to more reliable and precise predictions.

In addition, Random Forests can offer an understanding of the importance of different features, which can aid in feature selection or understanding the factors influencing classification outcomes. During the prediction process, they are able to deal with missing data by utilizing information from other trees and features.

Random Forests also offer the benefit of being able to parallelize the training and prediction tasks, which could result in quicker computation times, particularly for extensive datasets. The capability of scaling in Random Forests makes it a viable option for practical situations where computational effectiveness is important.

Even though the Random Forest model showed remarkable performance on the dataset, it is crucial to acknowledge that its effectiveness largely relies on the quality and relevance of the features supplied. In this instance, the astronomical data's measurements and parameters appear to be quite useful in differentiating between the spectral classes 'A' and 'B'.

Reflection

Overall, this data mining project gave lot of insights and experience during the phase of data mining and knowledge gained during the project from technical aspect as well as personality development point of view. From the technical point of view, one will agree that data preprocessing is a vital step that should never be performed haphazardly. Missing data, categorisation of variables, and normalisations are critical process steps that affect repeatedly the performance and accuracy of the models.

Data visualization was invaluable in learning more about the dataset, its general properties, data distributions, and relations between features when analyzing the dataset. It was helpful in deciding on when which preprocessing method was suitable and gave us insight into how to interpret the results of the model better.

The use of different algorithms in classifying car insurance risk level lets us compare the effectiveness and advantages of each algorithm, which shows that there is no all-purpose algorithm in data mining. It is also very important to underlining that choice of algorithm can depend on actual dataset, context of the problem and user’s requirements for the algorithm which can include interpretability or time complexity.

Some of the areas such as hyperparameters tuning and feature engineering were explained but not deeply discussed while they are important techniques that data miners must learn to perform during data analysis. It may also open the way for further enhancements to its performance, and adapt it to better identify more intricate patterns and dependencies within the data at hand.

In a sense of personal development, this project offered an opportunity to apply data mining approaches to a problem in field, which is an important step in turning a theory into reality. By performing data cleaning, feature selection and extraction, model construction, model assessment, cross-validation and implementation for real data, I gained further insights into those issues and factors which should be taken into account when conducting data mining tasks in practice.

As we look at this from a different perspective if one is to approach this problem again there are certain changes that could be made. Introducing different techniques such as gradient boosting or using a stacking method might provide even better results due to the utilization of various models' advantages. Algorithms could perhaps benefit from additional focus on techniques in feature engineering, including feature construction, feature subset selection, or feature extraction, as these practices could more effectively represent the data and enhance model performance.

REFERENCES

1. UM Students' Repository. http://studentsrepo.um.edu.my/8174/