**Homework 2**

**Part 1. Reflections on Homework 1**

The feedback received from the TA/ Peers was really helpful the valuable insights helped me improve the aspects of the current homework. I was able to learn new things about how the results should be organized and the comments on the results for the visualizations should be written such that they are easy to understand. I also appreciate the constructive feedback provided by my fellow peers on the dashboard areas it helped me to redesign my dashboard and in particular helped me to learn about a few new libraries in Python for a more interactive approach to displaying data.

**Part 2. Create a model card**

The model card contains various information about the properties like Decision tree, Naive Bayes, K-nearest neighbor, logistic regression, SVM(Support Vector Machines)

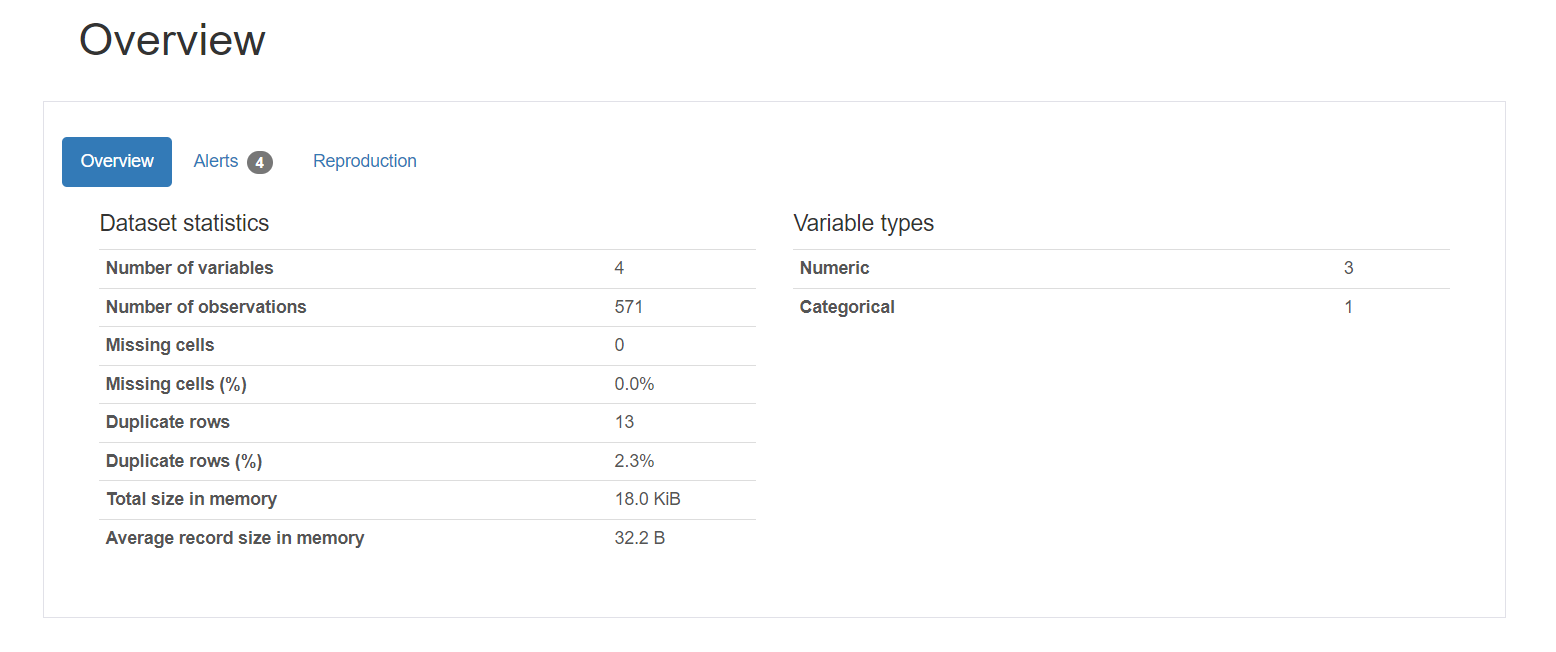
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Property** | **Decision Tree** | **Naïve Bayes** | **KNN** | **Logistic Regression** | **SVM** |
| **Parametric or Non-Parametric** | Non-parametric | Parametric | Non-parametric | Parametric | Non-parametric |
| **Input** | Mixed (continuous & Discrete) | Discrete | Mixed (continuous & Discrete) | continuous & Discrete | Mixed (Continuous & discrete ) |
| **Output** | Discrete | Discrete | Discrete | Discrete | Discrete |
| **Handle Missing Values** | Yes | Yes | Yes | Yes | Yes |
| **Model Representation** | Tree-like structure | Probability distribution | Proximity to K-neighbor | Linear equation | Hyperplane separation |
| **Model Parameters** | Splitting rules (e.g: Gini, Entropy) | Conditional probabilities | K value distance metric | Weights and bias | Support vectors, Margin |
| **Make the Model more Complex** | Increase tree depth, reduce leaf size | Add more features(independence assumption relaxed) | Increase K, use more distance metrics | Include more features, high-order | Use non-linear kernels, increase C parameters |
| **Make the Model Less Complex** | Decrease tree depth, increase leaf size | Reduce features, simplify independence assumption | Decrease K, use a simpler distance metric | Feature selection, regularization | Use a linear kernel, decrease the C parameter |
| **Interpretability** | Moderate | Highly interpretable | Moderate | Moderate | Moderate |

Part 3. Wine-Tasting Machine

**Python Tasks (50 points)**

1. **Read red-wine.csv into Python as a data frame, use a pandas profiling tool (**[**https://github.com/pandas-profiling/pandas-profiling**](https://github.com/pandas-profiling/pandas-profiling)**) to create an HTML file, and paste a screenshot of the HTML file here (10 points)**

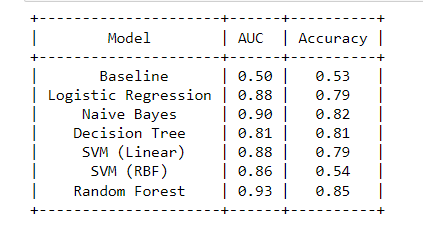
**Answer:** For this question I have loaded red-wine.csv as a data frame into the Python notebook then I have created a pandas profiling report to get an overview of the data that is being used, attached are the snippets of the HTML file of the dataset



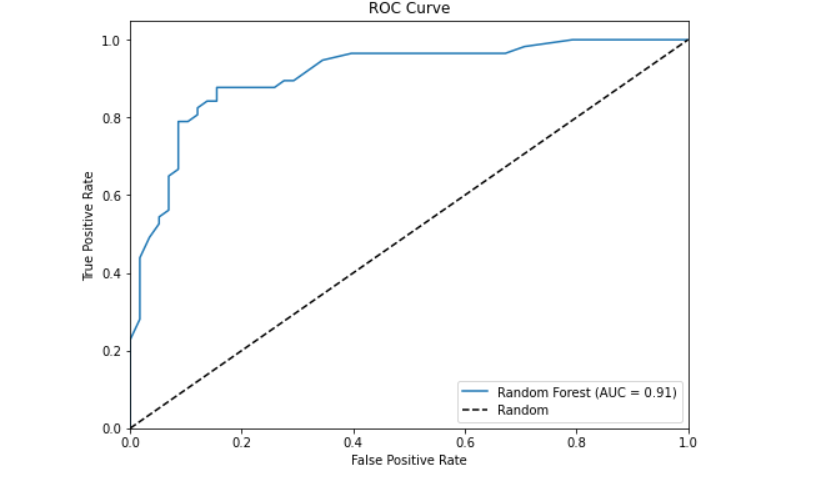
1. **Fit a model using each of the following methods and report the performance metrics of 10-fold cross-validation using red-wine.csv as the training set (25 points).**

**Answer:**

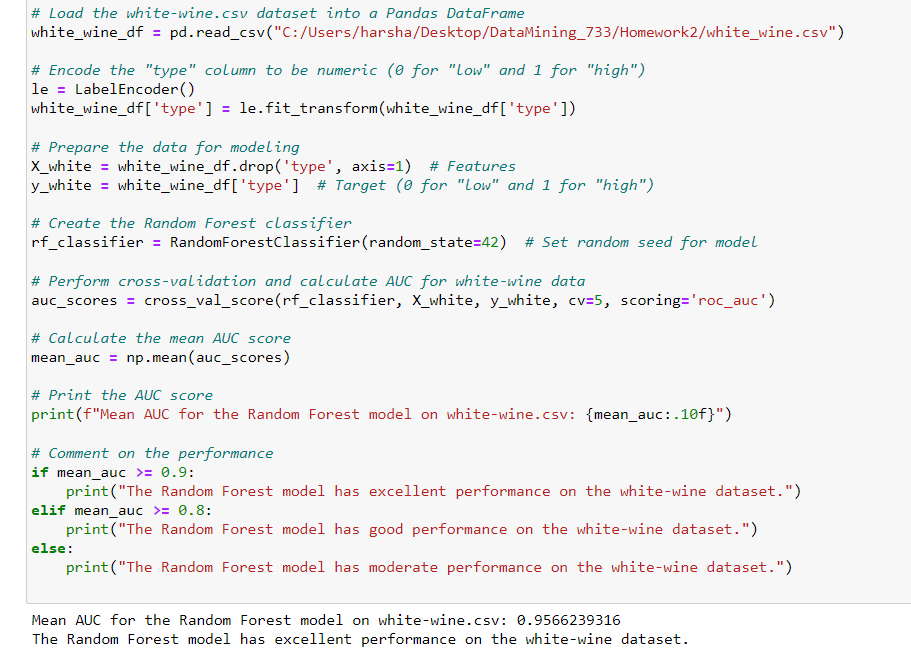
|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Model** | **Baseline** | **Logistic Regression** | **Naive Bayes** | **Decision Tree** | **SVM-Linear** | **SVM-RBF** | **Random Forest** |
| **AUC** | 0.50 | 0.88 | 0.90 | 0.81 | 0.88 | 0.86 | 0.93 |
| **Accuracy** | 0.53 | 0.79 | 0.82 | 0.81 | 0.79 | 0.54 | 0.85 |



1. **Plot the ROC curve of the Random Forest classifier from the Python package, and paste a screenshot of your ROC curve here (10 points)**



1. **Using the best model obtained above in Q2 (according to AUC), running the model on white-wine.csv, and reporting the AUC score, comment on the performance. (5 points)**

**Answer: **

1. **Suppose all the models have comparable performance. Which model would you prefer if the wine-tasting experts would like to gain some insights into the model? Note: there could be multiple model types fitting this criterion. (5 points)**

**Answer:**

If all the models perform similarly and the purpose is to acquire insights into the model, I would choose a model that gives greater interpretability and insights into the decision-making process. Models such as Logistic Regression and Naïve Bayes might be preferred. It is because in general, these models are more interpretable than advanced models like Random Forest, which yield greater accuracy but are less clear in terms of how predictions are made.

**Logistic Regression**: I prefer this model because it is easy to understand and shows how each factor affects the result. The coefficients assigned to each attribute are simply examined to determine their influence on the prediction.

**Naïve Bayes**: It is a simple model, it also provides insights into feature relevance via conditional probabilities. I can analyze how the model calculates the likelihood of a certain class based on the observed values of features.

**GPT**: I have used ChatGPT as a tool that helps me in writing the areas of Python code. I used this chatGPT to complete the task I felt it was an easy task but later understood that we can use chatGPT only to a limited extent and the remaining we have to go through the concepts of Python, understand the time complexity while running the code, design the plots in a way that they are easily understandable.

**ChatGPT Log:** [**https://chat.openai.com/share/19f2f6b2-da5b-4929-86f4-67712abfd321**](https://chat.openai.com/share/19f2f6b2-da5b-4929-86f4-67712abfd321)

**GitHub Link:**