**IS-733 HOMEWORK 2 – UB01976**

**Part 1. Reflections on Homework 1**

**1.** From the feedback you received, what are the takeaways/lessons learned you could apply to future analysis?

**1 Ans: -** The Homework 1 was about understanding the data, creating dataset profile using the data profiling library to clearly understand the attributes and variables of the dataset, creation of temporal plots based on the data, solving the machine learning problem the data could face using the data mining techniques and finally building dashboard for users to explore the data. I have used the **data profiling on dataset related to water quality metrics which helped me understand what are categorical and numerical values, how much are the mean, median, and Standard deviation values, and missing values in the data**. This helped me to **learn about data**, what can be done with it, segregate the data, and how to use the data for analysis.

From the Temporal plots I was able to **understand the patterns in data, like on which day were the maximum readings, how it varied over months and through the complete yea**r. The machine learning problem taught me how to use the data mining techniques to **classify the data, sort or organize it, train a model to understand the data**, and give the required analyzed output which can be used for similar data in future.

**Distribution plots** have helped me in determining the potential application of machine learning models.The distribution plots gave me a better glance towards various water quality metrics in the form of **histograms and barplots, which helped me analyze the frequency of readings for various attributes**. From these, I was able to capture the required information and learned how to use **scatter plots, and heatmaps** to understand the correlation between parameters and how to apply them for any other use cases in future.

The machine learning problem which I found from my dataset was related to finding out whether water quality is good or bad. For that I have trained the model on 80 % of the dataset and tested on 20 % using the random forest model. In this process I have learnt **how to split, and train the dataset, how to choose the best model (i.e. which one to use whether it is classification, regression, or Naive bayes), and how to train the model on this dataset**. Later on check whether the accuracy, precision, recall, and F-1 score are as expected or not, to verify whether the model is performing correctly or not. At first, I wasn’t sure how to choose a modelling technique that is best suited for data, but later on I have verified based on AUC score and accuracy metrics which model is to be chosen. These helped me learn and revise the **data mining techniques** once again.

While creating the dashboard using the python libraries like dash, I have clearly understood how to use the features like **scaling over the plots, rolling mean, and how to interpret the data**. This has helped me create a dashboard with features of selecting a particular **point of time on the plot to look for readings, zooming in and out, and downloading the plot**. These have helped me improve my skills related to precision and looking at the minute details on the plots to analyze the plot even more clearly.

Overall, from Homework1, I have understood that apart from analyzing the data, it is equally important to interpret and **provide the insights of the data to the users** in the form of **visuals**. Once again I could revise the **data modelling** and have improved my knowledge related to how to use **data visualization tools and data mining techniques.** I can now apply thesein the upcoming data mining problems and future analysis.

**Part 2. Create a model card**

The model card contains information about the properties of the models. This is one way of organizing the knowledge about the model, which becomes handy in data science problem-solving. Prepare a table summarizing the properties of each base model we learned so far (Decision tree, Naive Bayes, K-nearest neighbor, logistic regression, SVM) with respect to the following properties:

1. parametric or non-parametric
2. Input (continuous or discrete or both or mixed)
3. Output (continuous or discrete or both)
4. Can the model handle missing value
5. Model representation
6. Model Parameters
7. How to make the model more complex
8. How to make the model less complex
9. Is the model interpretable or transparent

**2 Ans: -** Please find below the model card containing the information related to the properties of the models, this is a summarized table consisting of properties of each base model;

**Model Card**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Property** | **Decision Tree** | **Naive Bayes** | **K-Nearest Neighbor** | **Logistic Regression** | **Support Vector Machine** |
| **1. Parametric/Non-Parametric** | Non-Parametric | Parametric | Non-Parametric | Parametric | Non-Parametric |
| **2. Input** | Takes both Continuous and Discrete | Takes both Continuous and Discrete | Takes both Continuous and Discrete | Takes both Continuous and Discrete | Takes both Continuous and Discrete |
| **3. Output** | Discrete ( for Classification)  Continuous ( for Regression) | Discrete | Discrete | Discrete | Discrete ( for Classification) |
| **4. Handle Missing Value** | Yes | No | No | No | No |
| **5. Model Representation** | In Tree Structure | Probabilistic (Using Bayes Theorem) | Distance Based (Distance calculation between instances) | Linear Function (Logistic function) | Hyperplane (Kernel for non-linearity) |
| **6. Model Parameters** | Depth of the tree, Splitting Criteria | Probability estimates (Log\_Odds, Odds, Likelihood) | Distance, and Number of Neighbors (K values) | Coefficients (Weights), Intercepts | Gamma ( for RBF Kernel), Kernel type, C (Regularization) |
| **7. Make the model more complex** | Increase tree depth ( have more branches or divisions), reduce pruning | Add more features | Increase K value for more flexible distance metric | Reduce Regularization, and add polynomial features | Use a complex kernel (e.g., RBF), decrease Regularization |
| **8. Make the model less complex** | Reduce the depth of the decision tree and prune the tree | Reduce the number of features | Reduce K value, to make it simpler for distance metric calculation | Increase Regularization | Use linear kernel, increase regularization |
| **9. Interpretability/Transparency** | Yes (As it is easy to visualize the decision tree) | No (Difficult to interpret directly) | No (As predictions are based on instance comparisons) | Yes (Coefficients directly show the feature impact) | No |

**Part 3. Wine-Tasting Machine**

**1.** Read red-wine.csv into Python as a data frame, use a pandas profiling tool (<https://github.com/pandas-profiling/pandas-profiling>) to create an HTML file.

**Part 3 1 Ans: -** Please find the Github link below, where there is python code for Data profiling and HTML file for the red-wine.csv file;

<https://github.com/UB01976/is7332025/blob/main/data-mining-project-repo/hw2/UB01976_Homework2.ipynb>

**2.** 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.

**Part 3 2 Ans: -** Below is the table showing the values of Accuracy and AUC of various models with 10-fold cross-validation;

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Model** | Baseline | Logistic Regression | Naive Bayes | Decision Tree | SVM - Linear | SVM - RBF | Random Forest |
| **AUC** | 0.5 | 0.87094 | 0.88248 | 0.76021 | 0.87130 | 0.91059 | 0.86945 |
| **Accuracy** | 52.89 % | 79.51 % | 82.14% | 75.83 % | 78.98 % | **82.31 %** | 80.04 % |

Please find the Github link below, where there is python code for understanding performance of various model with 10-fold cross-validation using the red-wine.csv file as training data;

<https://github.com/UB01976/is7332025/blob/main/data-mining-project-repo/hw2/UB01976_Homework2.ipynb>

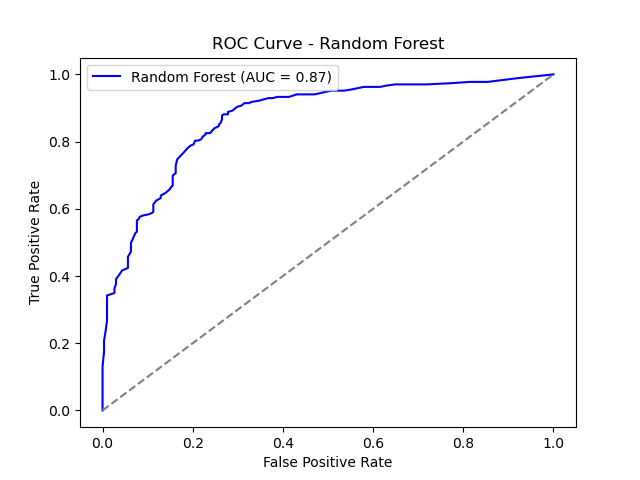
**3**. Plot the ROC curve of the Random Forest classifier from the Python package, and paste a screenshot of your ROC.

**Part 3 3 Ans: -** Please find the Github link below, where there is python code for generating the ROC curve for the Random Forest classifier;

<https://github.com/UB01976/is7332025/blob/main/data-mining-project-repo/hw2/UB01976_Homework2.ipynb>

The ROC curve of Random Forest classifier is as shown below;

(*ROC curve is continued on the next page…)*

****

**4**. 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.

**Part 3 4 Ans: -** The best model obtained from the Q2 (according to AUC) would be the **SVM-RBF** (with AUC Score = **0.9105**). So, running the model on white-wine.csv file would give us the following results;

SVM-RBF model trained on white-wine dataset - Accuracy - **81.25 %**

AUC - **0.9455**

As we can see the accuracy is **81.25 %** which is pretty high and denotes that the model is making 81 correct out of 100 predictions. And looking at the AUC score of **0.9455 which** is very high, gives us the information that the model can distinguish between the low and high values for type attribute very accurately.

So, this high AUC score suggests that the model will be able to predict accurately even in case of some class imbalance.

Please find the Github link below, where there is python code for generating the AUC score and Accuracy of white-wine dataset using SVM-RBF;

<https://github.com/UB01976/is7332025/blob/main/data-mining-project-repo/hw2/UB01976_Homework2.ipynb>

**5**. 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?

**Part 3 5 Ans: -** Assuming the case where all the models have comparable performance, we can go ahead with models like **Decision tree** and **Logistic Regression** for explaining or providing the insights of the model to wine-tasting experts.

The reason for selecting these models is that they are easy to interpret and explain to others. Decision trees provide the **visual representation** or structure of the data in the form of a tree, which is easy for someone from non-technical background to understand. Coming to Logistic Regression, it provides clear **feature importance through coefficients**. By doing so, wine-tasting experts can clearly see how each variable affects the probability of classifying the wine type into “High” or “Low”.

We can use other methods as well like **SVM and Random Forest**, but the problem is that these models are **complex and difficult to intercept** as they rely on high dimensional space. For example, Random Forest is simply a black box model with several decision trees in it, it would be really difficult for someone to **explain these individual predictions** and people from non-technical background may not understand these predictions easily.

Hence, we can use models like **Decision tree and Logistic regression** if the wine tasting experts want to understand the insights of the model.