

**ASSIGNMENT FRONT SHEET**

**Course Name: ALY6020 20906 Predictive Analytics**

**Professor Name:**  **Dr. Yevgeniy Frenkel**

**Student Name: Dong Quoc Tuong (Lukas)**

**Student Class: Fall 2019 CPS Term: B. 2020**

|  |
| --- |
| **Module 1: Classification Using Nearest Neighbors**  **Completion Date: April 12th Due Time:12:00am** |

**Statement of Authorship**

*I confirm that this work is my own. Additionally, I confirm that no part of this coursework, except where clearly quoted and referenced, has been copied from material belonging to any other person e.g. from a book, handout, another student. I am aware that it is a breach of Northeastern University’s regulations to copy the work of another without clear acknowledgement and that attempting to do so renders me liable to disciplinary procedures. To this effect, I have uploaded my work onto Turnitin and have ensured that I have made any relevant corrections to my work prior to submission.*

**Tick here** to confirm that your paper version is identical to the version submitted through Turnitin

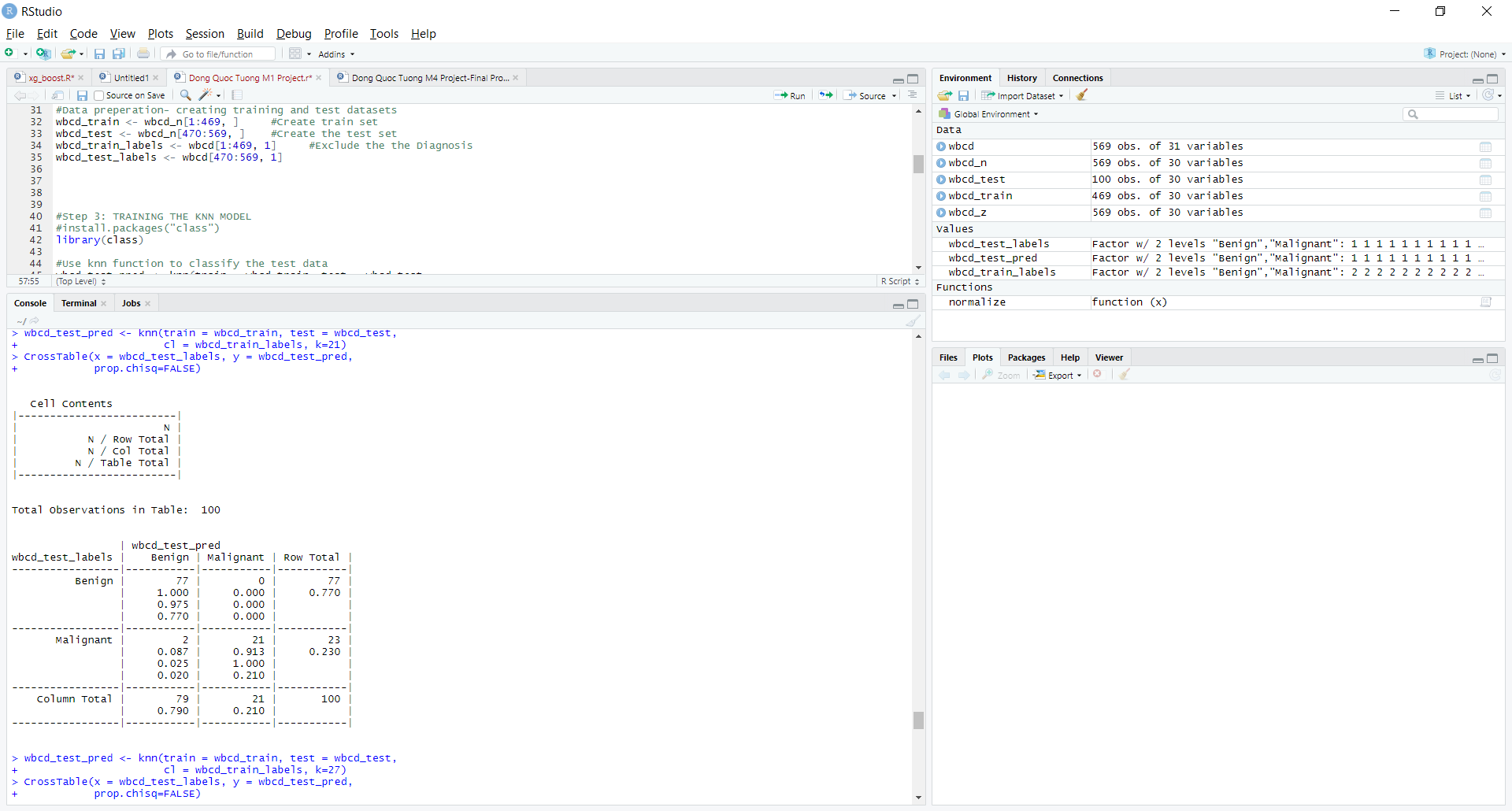
**Executive Summary**

KNN is one of the simplest yet most effective prediction models that can accurately foresee the outcome of a business. In this exercise, we will have a look at the Breast Cancer dataset and use some metrics to predict our model’s accuracy

**Analysis**



The dataset consists of 569 inputs with 32 features as we expect. After some data preparation process we can see that the Benign has 357 values, which accounts for 63% of the total dataset whereas only four out of ten cases are Malignant.

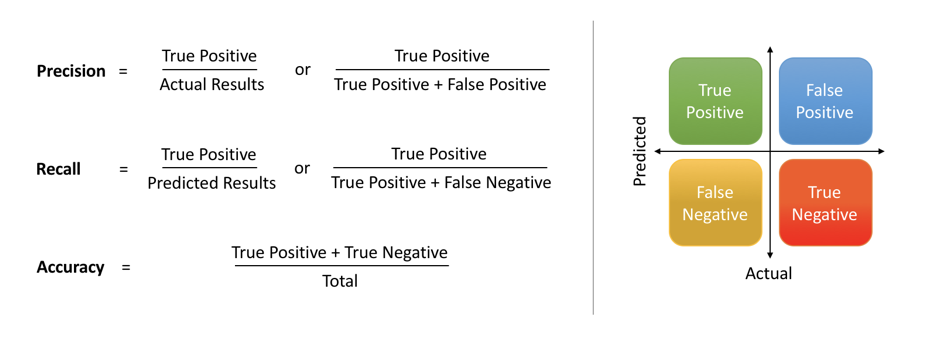


Since our training data includes 469 instances, we will can try k= 21, an odd number that approximately equals to the square root of 469. Bearing in mind that using an odd number can significantly lower the chance of ending with the tie vote. After building the KNN model, we created the matrix map below and saw that our estimated outcome committed 2 Type I errors and correctly predicted everything else. Because this is an Type I error, there might be some consequences that we need to take into consideration between choosing this model

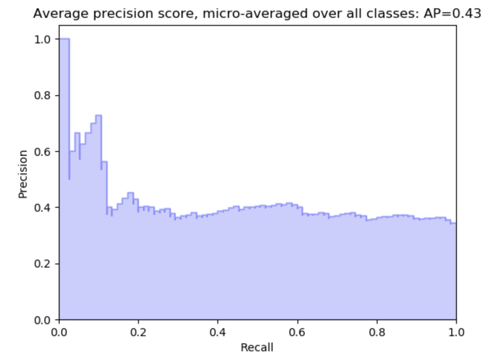
**Q1**

There are multiple metrics to judge the performances of the KNN models. The most popular two that people like to use is the “Precision & Recall” and “Trade off”.

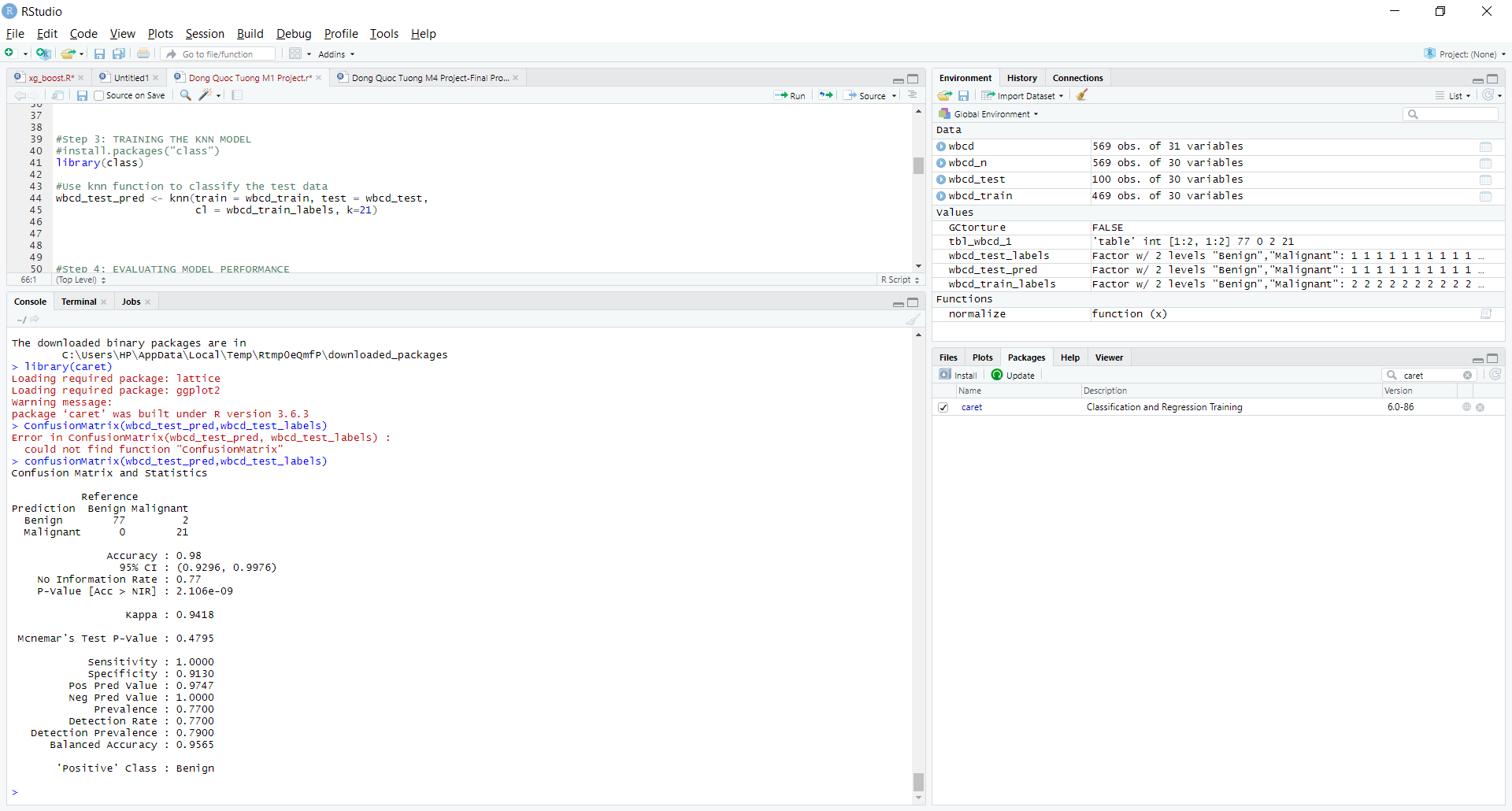
* Precision is defined as the percentage of your model’s results that is similar to the test set. On the other hand, recall means the percentage of total relevant results that are correctly categorized by the model. Therefore, both numbers have the range from 0 to 1, the closer they are to 1, the better. Sometimes, the Precision number is not enough to determine the usefulness of your model. For example, when predicting a terminal illness like breast cancer in this exercise, Recall is around 0.9 percent, which means that we have a high Type I error. Our model has mistakenly predicted 1 in every 10 positive cases as negative, thus, our customers will die because of our carelessness



* Trade off: When we are using the KNN models, if we neglect some features and build a simple model, we will have high bias and low variance (underfitting). On the contrary, if the KNN model has a higher number of features and get more complex (high k) we will have high variance and low bias (overfitting). It also affects the Precision and Recall as well, the higher the Precision is, the lower Recall is and vice versa. Hence, we need to make sure to choose the right number of K to reach a good balance between Precision and Recall (Saxena, 2018)



**Q2**

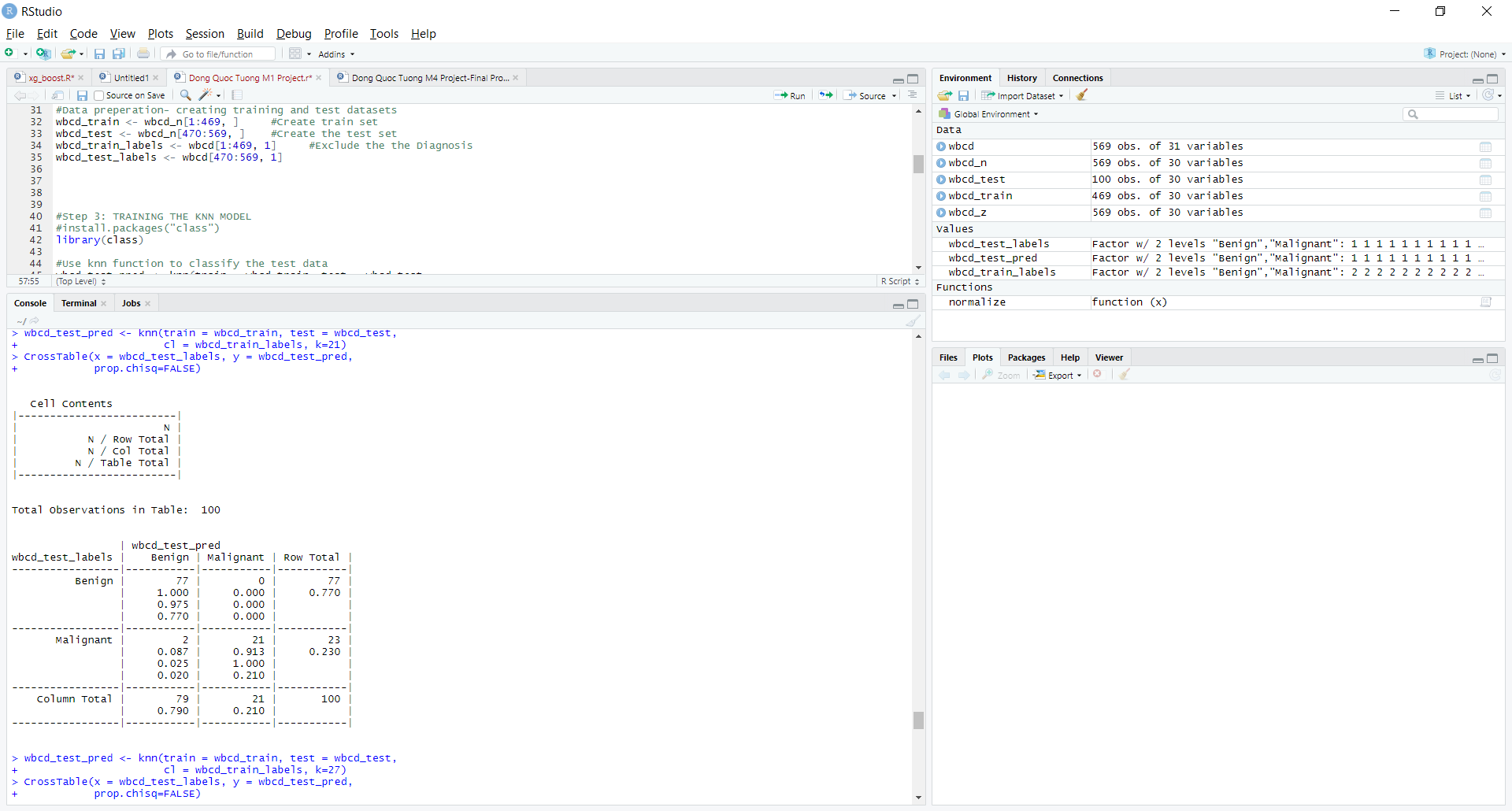


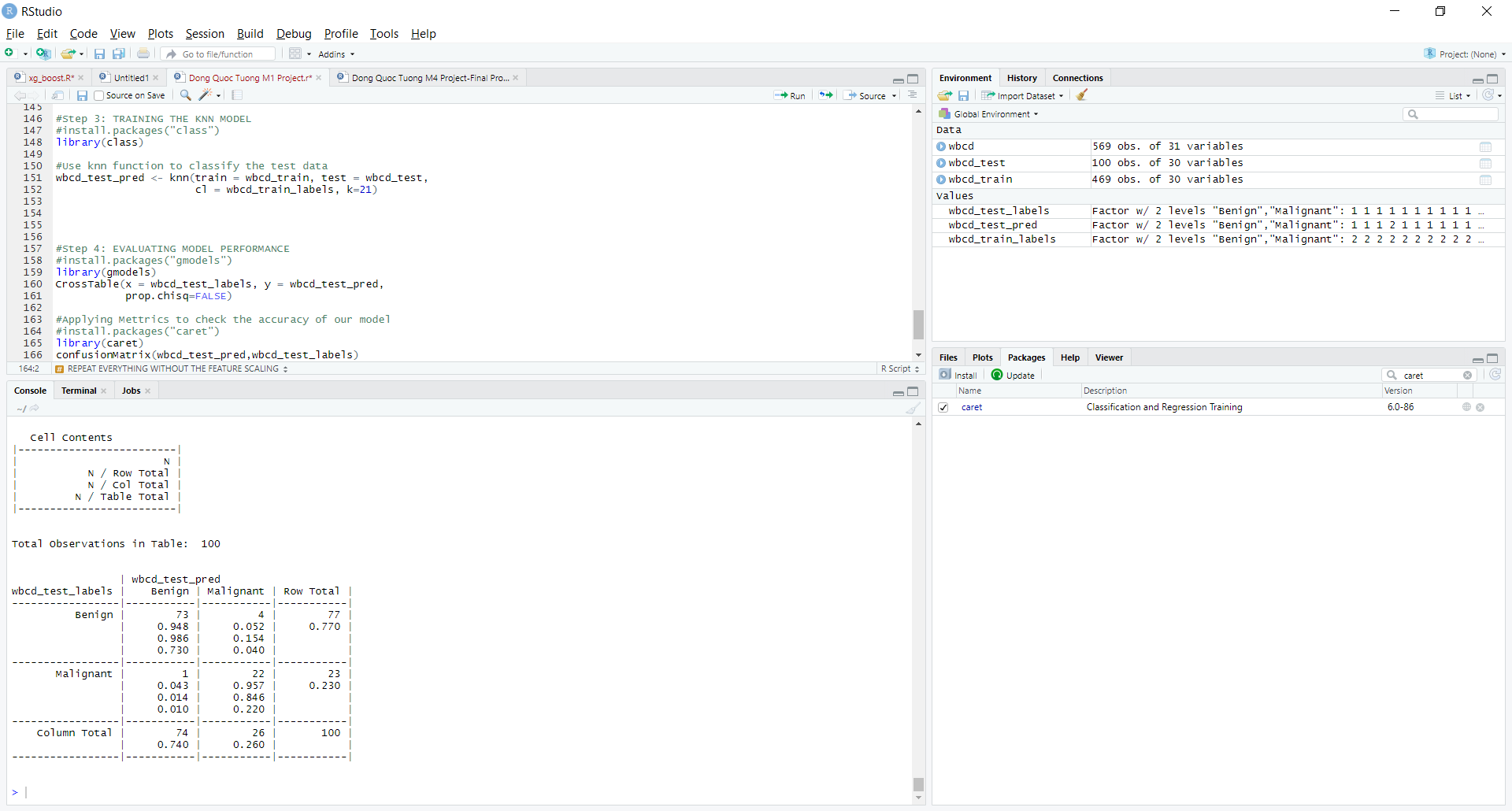
The Recall number (Sensitivity), which is 1, indicates that in general all the there is almost none faulty in our prediction. Equally important, Then Precision value, which is 0.9747, indicates that our model’s outcomes are perfectly in sync with the test set. After running the KNN model through different K, we saw that the precision and Recall values do not change. Thus, we can conclude that safely say that our Model withstands any criticism without conducting any trade off.



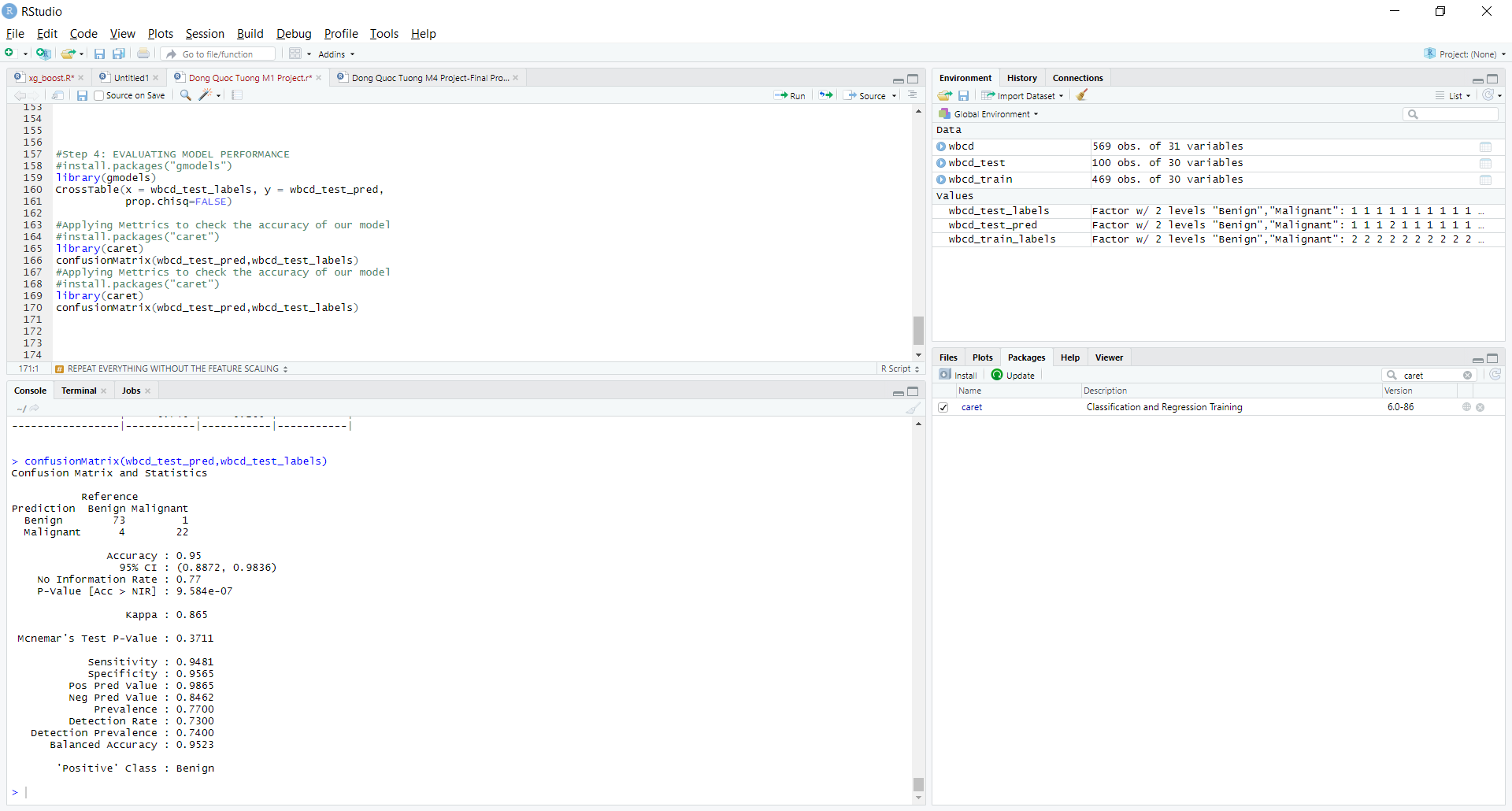
Nevertheless, if we have a closer look at the Type I and Type II map below, we would go for K value of 5 rather than 11 or 21. It is due to the fact despite having higher error rate; K value 5 produced no False Negatives. This means that it does not mistakenly predict any Malign cases as Benign, which is the ultimate aim of our prediction model.



**Q3**



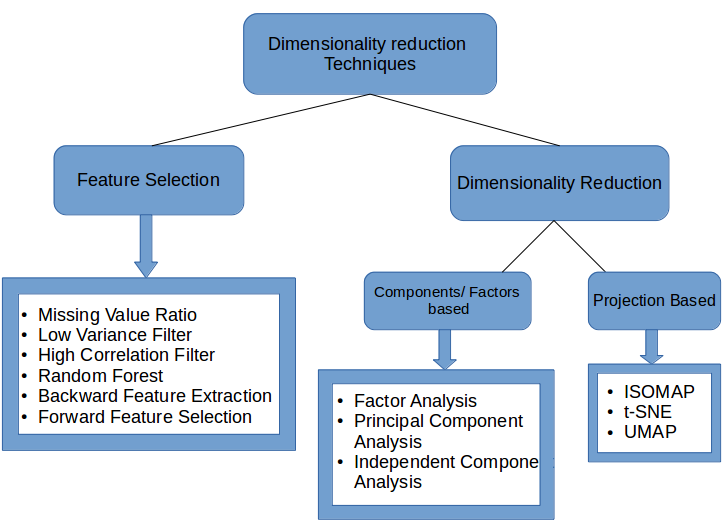
After eliminated the Scaling process in our code, we see that there is a slight increase in the wrong prediction, both Type I and Types II. Before, there were 79 Benign cases and now it dropped to 74, whereas Malign cases rose from 21 to 26.



The Recall rate is 0.94 instead of 1 while the precision rate increases to 0.9865. This is a trade off between Recall and Precision. Nonetheless, we know that even the precision rate increases, the model is not correct.

**Q4**

In general, ranking features based on their impact on the algorisms performance is considered to be Dimensionality reduction. Dimensionality reduction is an encompassing definition with two factors: Feature Selection and Feature Extraction. Feature Selection, where feature ranking methods belong, is the easier methods between two of them. In order to rank the features, the researchers can employ multiple approaches: Missing Value ratio, Low Variabance Filter, Forward feature elimination, etc. The most common one is Backward feature elimination but it depends on the analysis and dataset to choose the best Feature ranking approaches (Sharma, 2019)



**Conclusion**

In spite of its simplicity and efficiency, KNN still requires careful considerations before the researcher decided to use them. As this case has demonstrated that, it is better to have no avoidable consequences than to be wholeheartedly accurate.

**References**

Saxena, S. (2018). Precision vs Recall. Retrieved from <https://towardsdatascience.com/precision-vs-recall-386cf9f89488>

Sharma, P. (2019, September 5). The Ultimate Guide to 12 Dimensionality Reduction Techniques (with Python codes). Retrieved from <https://www.analyticsvidhya.com/blog/2018/08/dimensionality-reduction-techniques-python/>