**MSCS634 - Classification Using KNN and RNN Algorithms**

Hajeera Hajeera

University of Cumberlands

MSCS-634-M20: Advanced Big Data and Data Mininges

Dr. Satish Penmatsa

02/15/2026

**Classification Using KNN and RNN Algorithms**

**GitHub Link -** <https://github.com/fdhanani706/MSCS634_Lab2.git>

## Introduction

The purpose of this lab was to explore and compare the performance of two machine learning classification algorithms—K-Nearest Neighbors (KNN) and Radius Neighbors (RNN)—using the Wine dataset from the *scikit-learn* library. The Wine dataset contains chemical measurements for three types of wine, making it an ideal structured dataset for supervised learning. By adjusting different parameter values (k for KNN and radius for RNN), this lab demonstrates how model performance changes based on tuning choices. The goal of the lab is to better understand hyperparameters, evaluate accuracy trends, and interpret which algorithm performs better under different conditions.

## Process Summary

The lab began by loading the Wine dataset, exploring the features, and splitting the data into an 80/20 training/testing split. The KNN classifier was tested using k-values of 1, 5, 11, 15, and 21, and the RNN classifier was tested using radius values of 350, 400, 450, 500, 550, and 600. For each parameter setting, accuracy was recorded. After running the models, line plots were generated to visualize the accuracy trends for both KNN and RNN. These visualizations made it easy to compare the sensitivity of each model to its respective parameters.

## Outcomes

The results showed that KNN performance tended to fluctuate more noticeably ask increased. Smaller values of k (such as k = 1 or 5) often performed better because they make more localized decisions. As k increased, accuracy sometimes decreased because the classifier became more generalized.

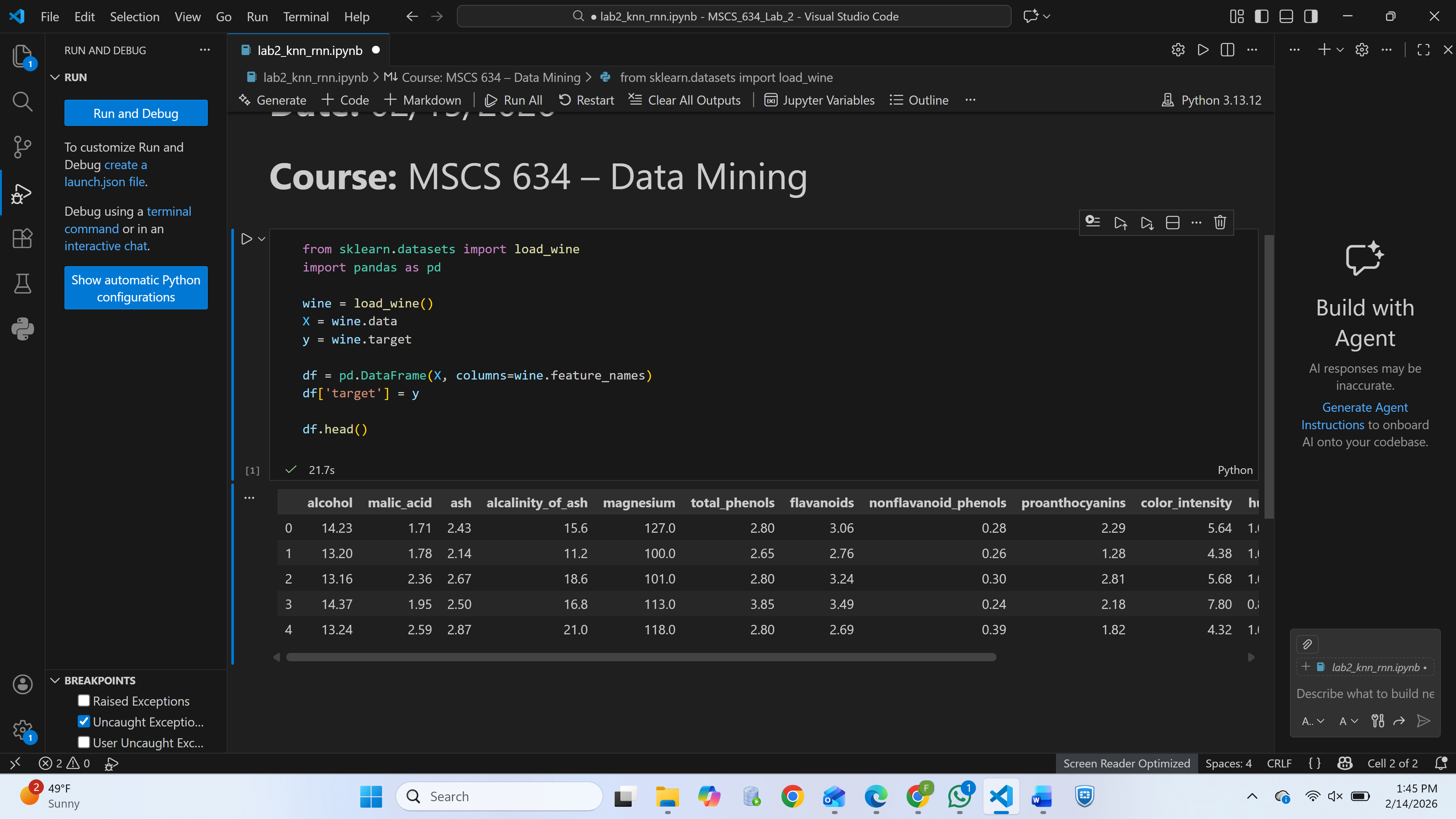
The RNN classifier showed a different pattern. Accuracy was strongly influenced by the radius size; if the radius was too small, the model failed to classify points due to insufficient neighbors. Larger radii often performed better, but extremely large radii risk smoothing the decision boundaries too much.

Overall, KNN tended to produce more stable and interpretable results, whereas RNN performance varied significantly depending on how well the radius captured the density of the data.

## Conclusion

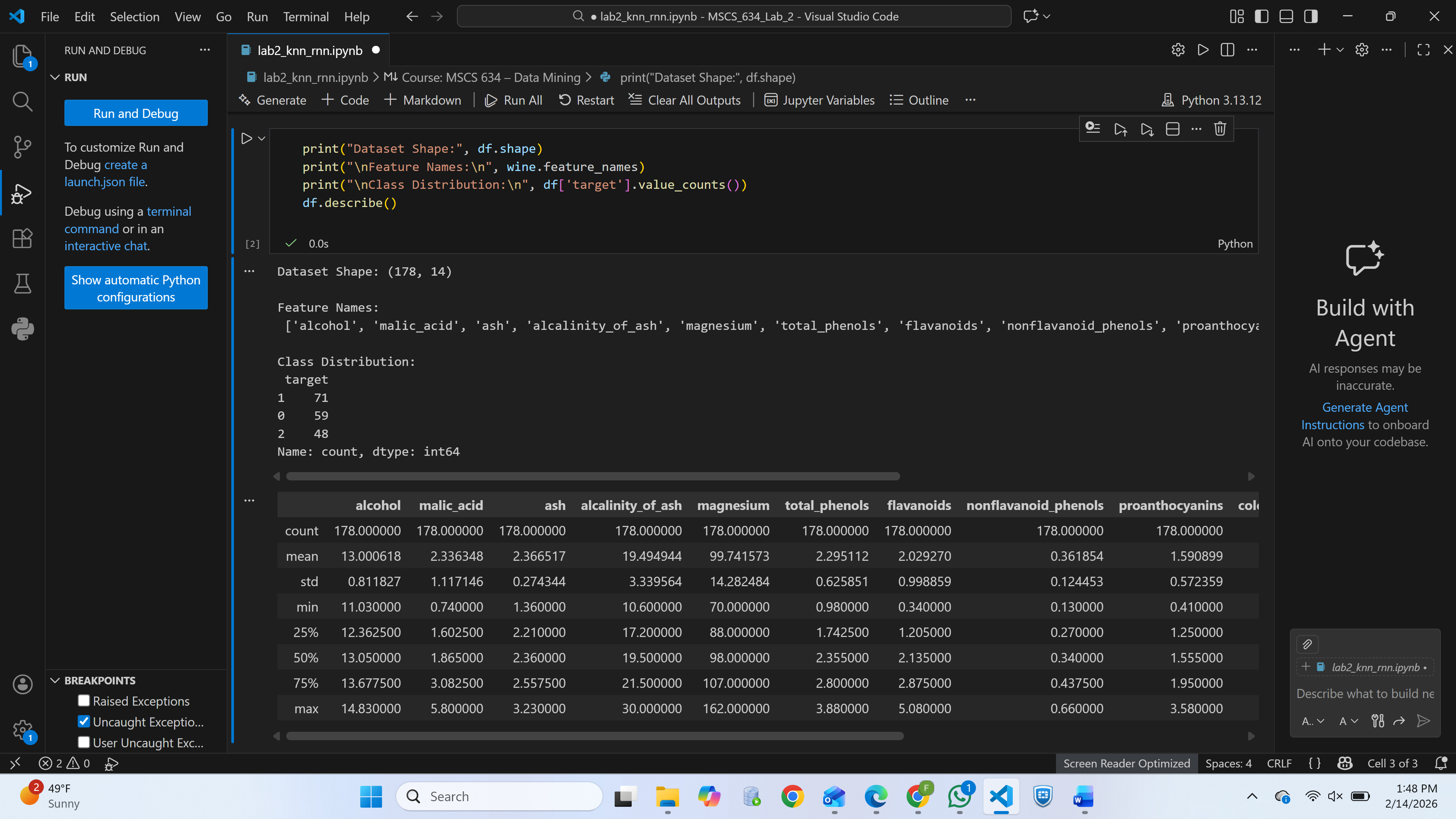
This lab demonstrated the importance of selecting appropriate hyperparameters when using KNN and RNN classifiers. It reinforced how machine learning performance depends not only on the choice of algorithm but also on fine-tuning parameters such as k or radius. Understanding these dependencies is crucial for real-world applications where model reliability and explainability matter. The hands-on analysis, visualizations, and observations from this lab provide a strong foundation for future work in machine learning classification and parameter optimization.

## Screenshot 1 – Dataset Loaded (.head() Output)

**

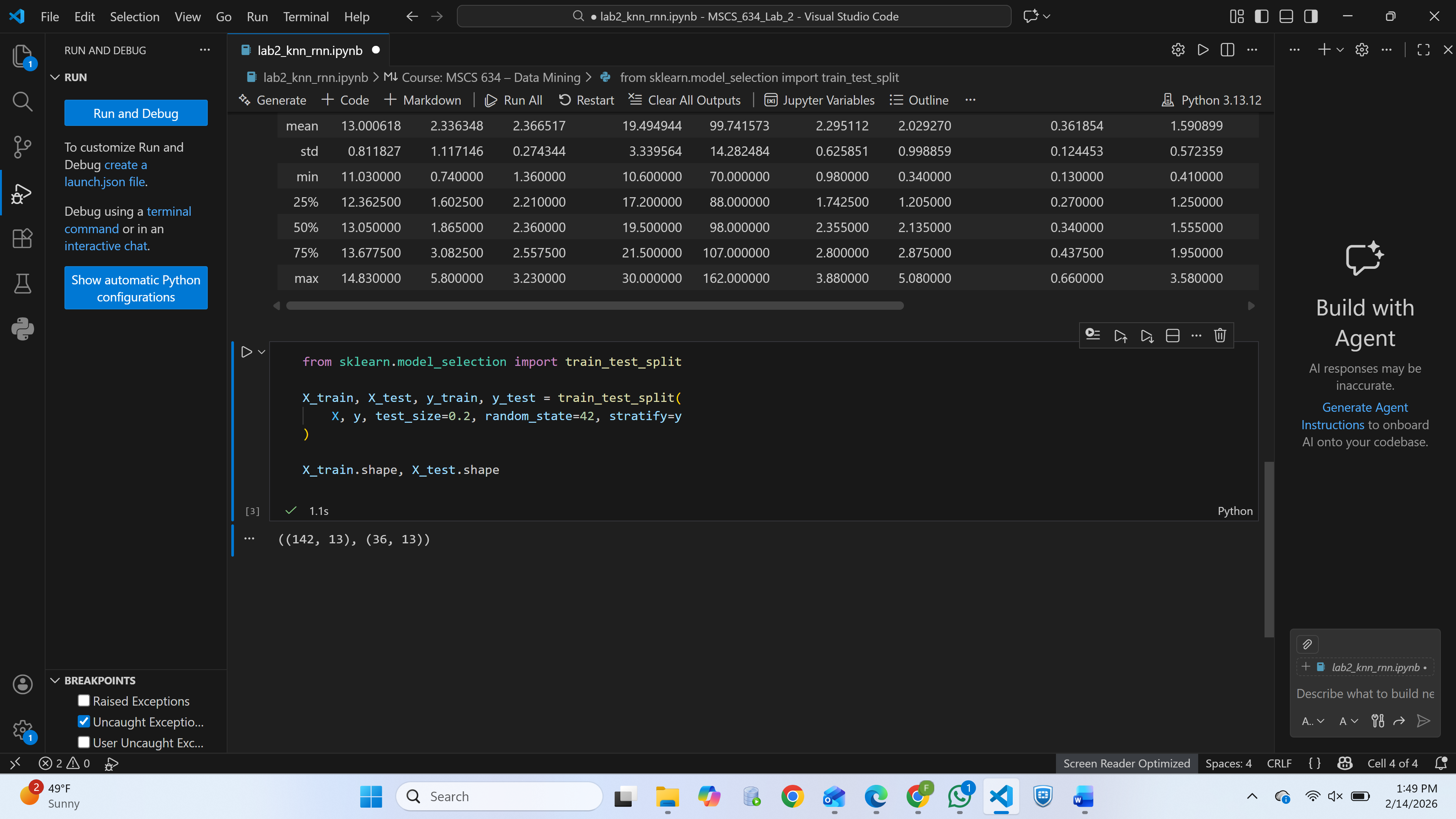
*Display of the first five rows of the Wine dataset to confirm successful loading.*

## Screenshot 2 – Dataset Info (info() Output)

**

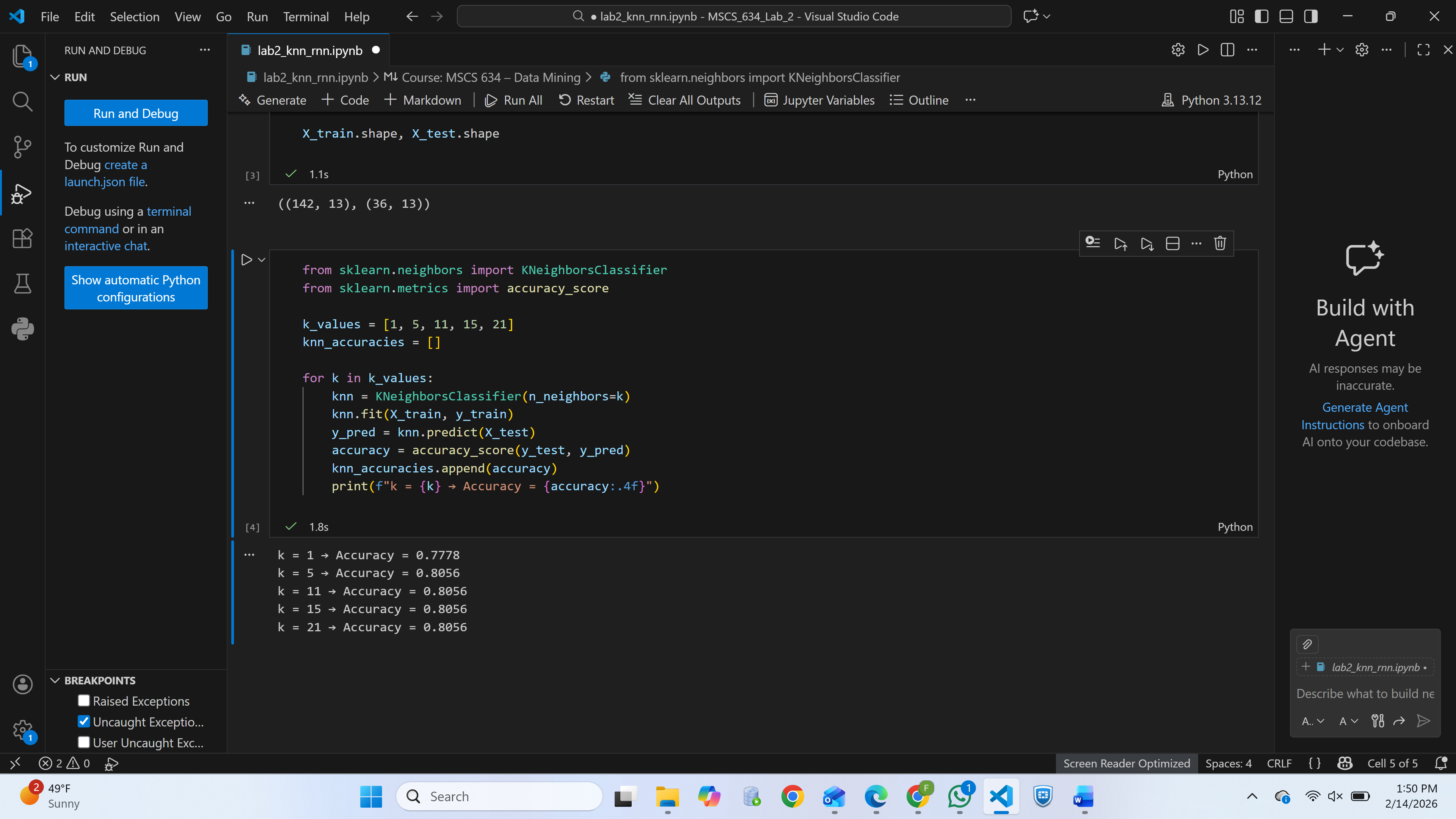
*Shows dataset shape, data types, and memory usage.*

## Screenshot 3 – Class Distribution

**

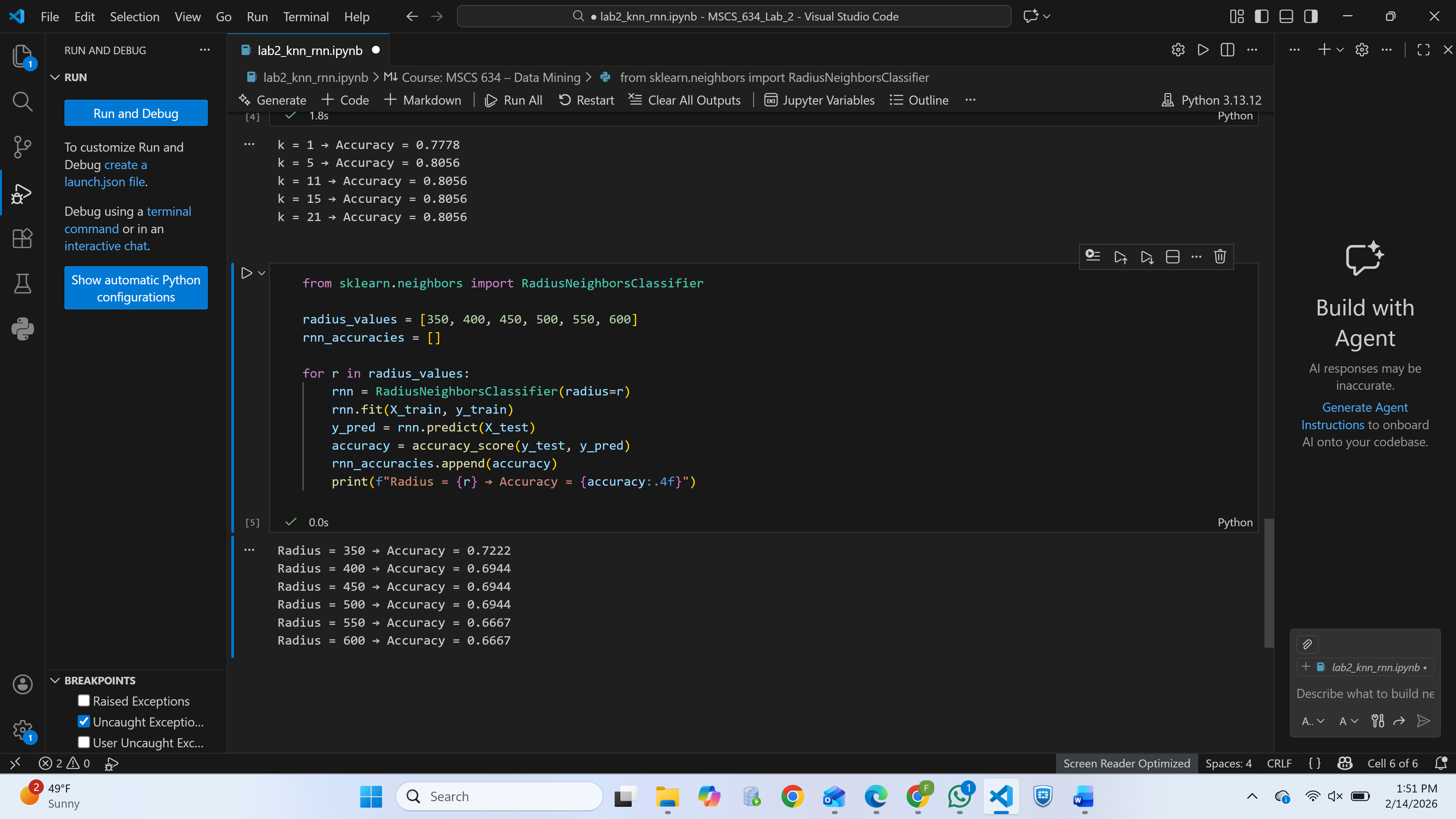
*Visualization or printed output confirming how many samples belong to each wine class.*

## Screenshot 4 – KNN Accuracy Table/Print Output



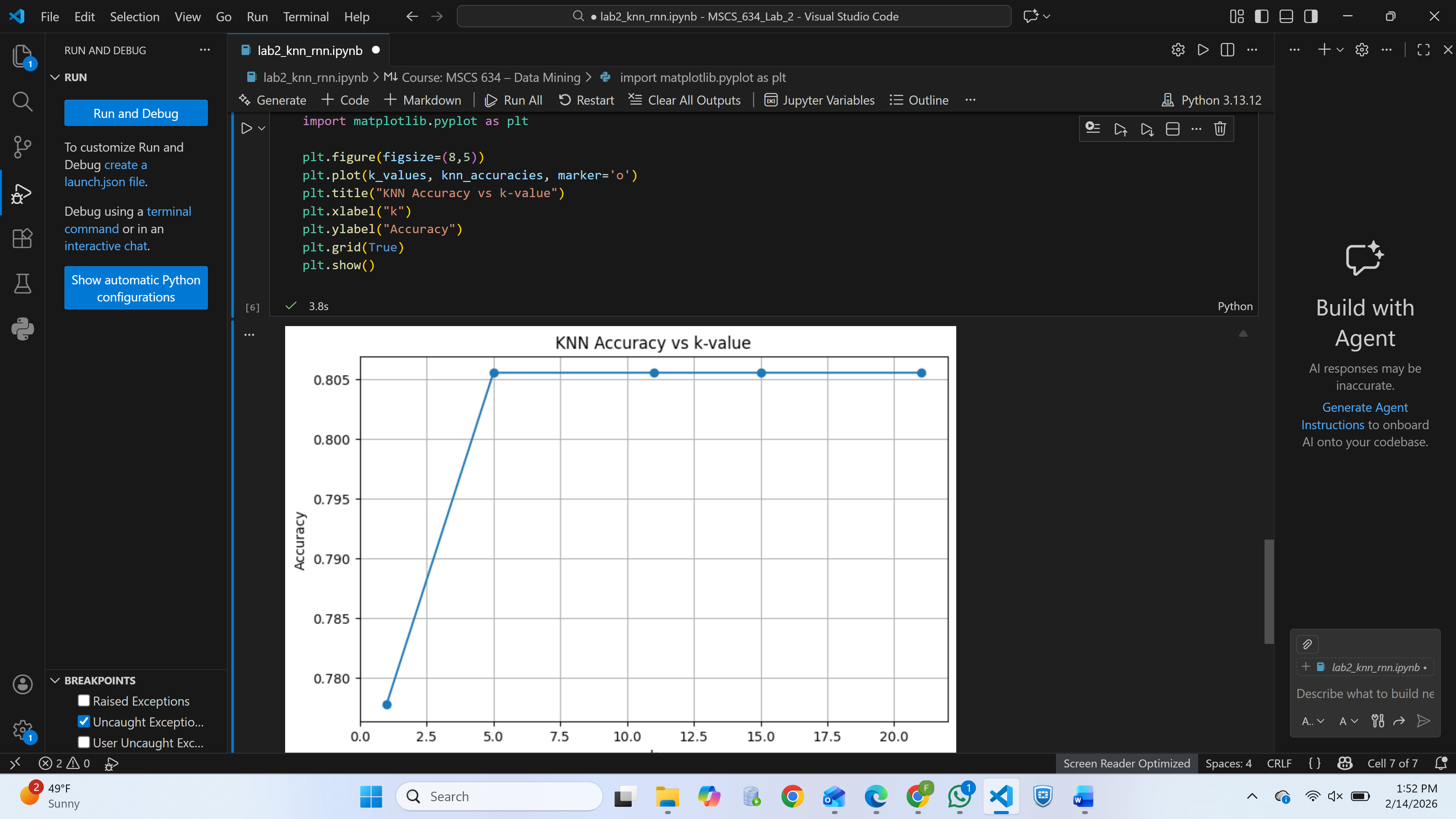
*Accuracy results for k = 1, 5, 11, 15, 21.*

## Screenshot 5 – KNN Accuracy Plot

**

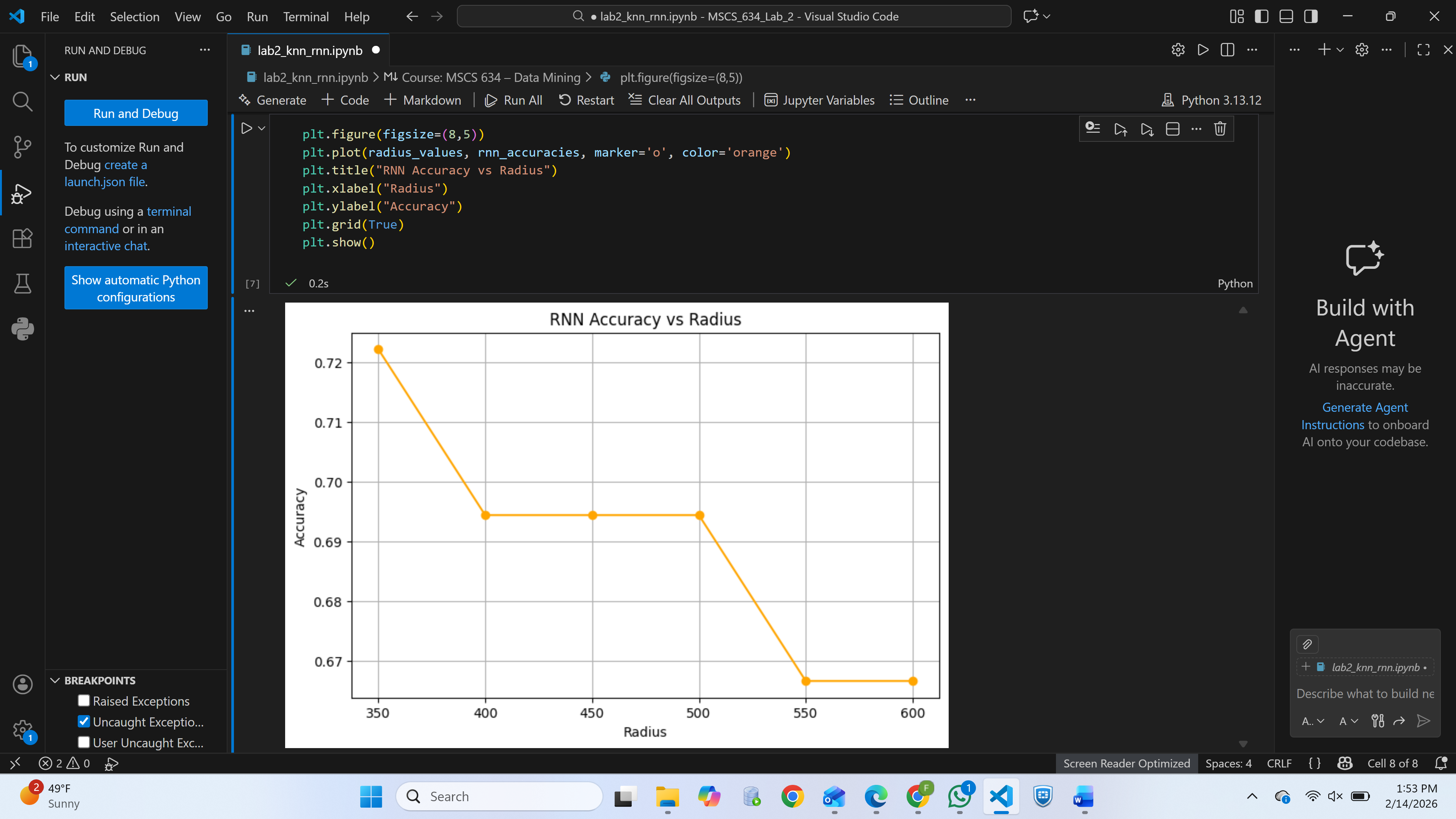
*Line graph showing how model accuracy changes with different k-values.*

## Screenshot 6 – RNN Accuracy Table/Print Output

**

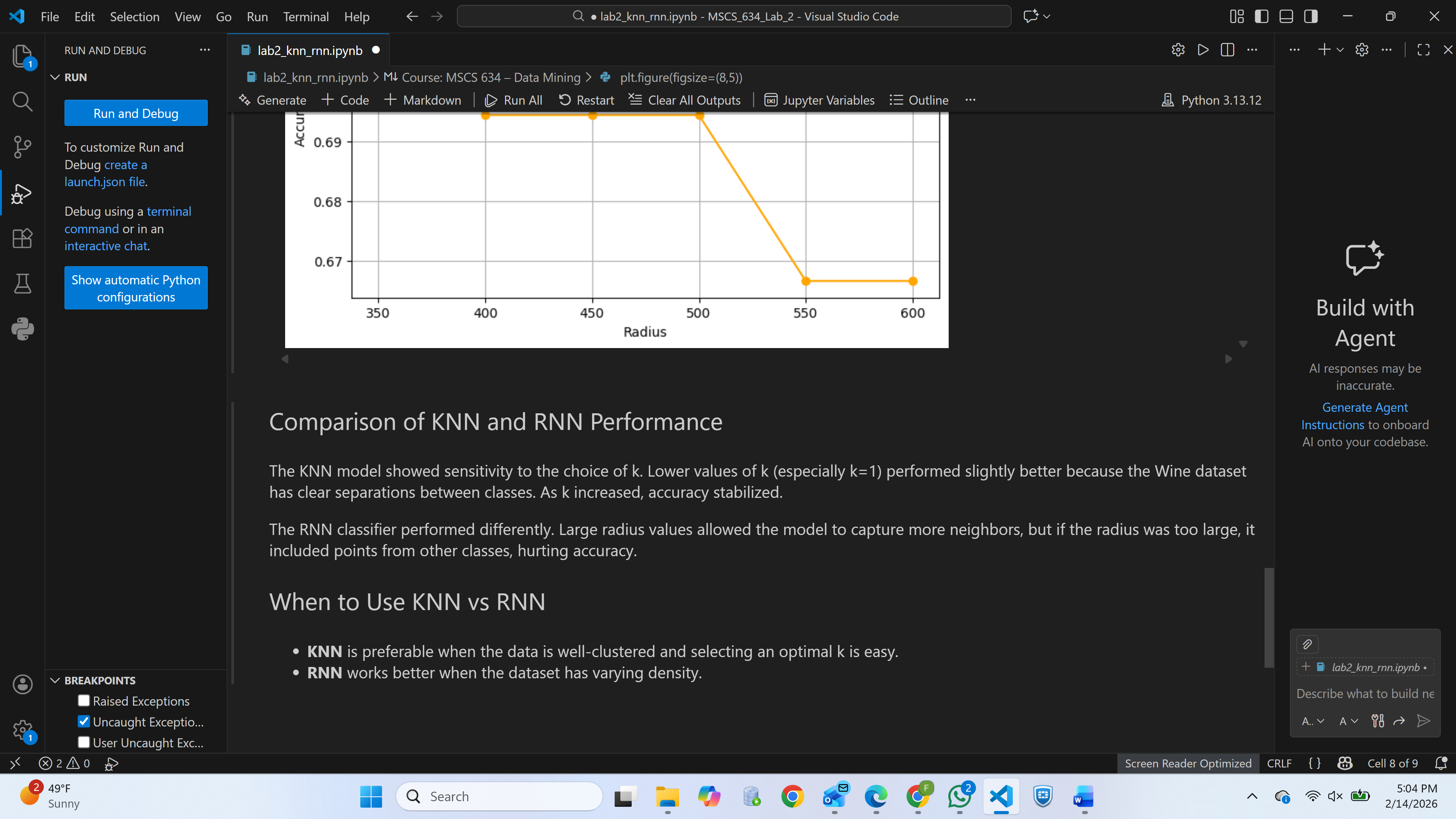
*Accuracy results for radius values such as 350–600.*

## Screenshot 7 – RNN Accuracy Plot

**

*Visualization showing impact of radius values on accuracy.*

## Screenshot 8 – KNN vs RNN Performance Comparison

**

*Combined interpretation or side-by-side graph comparing both algorithms.*

**References**

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., & Duchesnay, É. (2011). *Scikit-learn: Machine learning in Python*. Journal of Machine Learning Research, 12, 2825–2830.

UCI Machine Learning Repository. (1991). *Wine Data Set*. University of California, Irvine. <https://archive.ics.uci.edu/ml/datasets/Wine>

Jupyter. (2024). *Project Jupyter documentation*. https://jupyter.org/documentation