Applications Development Laboratory (Machine Learning Applications) Course Code CS33002 (L-T-P-Cr: 0-0-4-2)

Name: Shubh Agnihotri

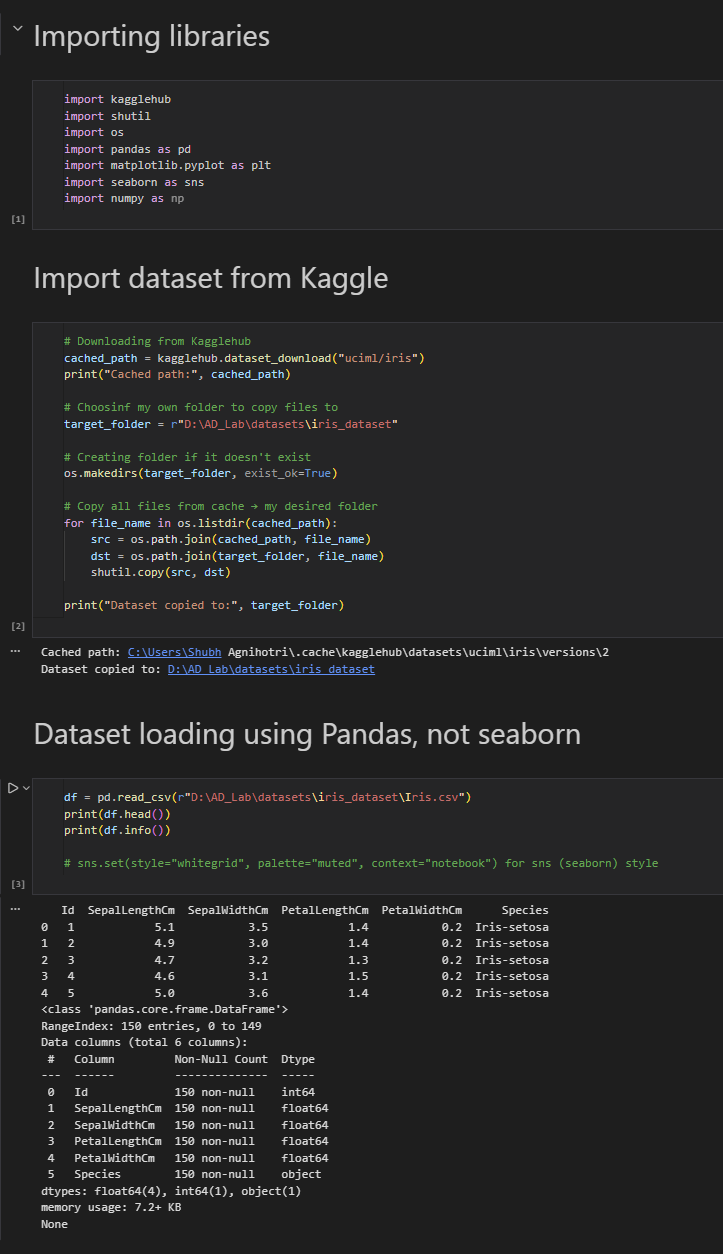
Section: CSE-6

Roll No: 2305336

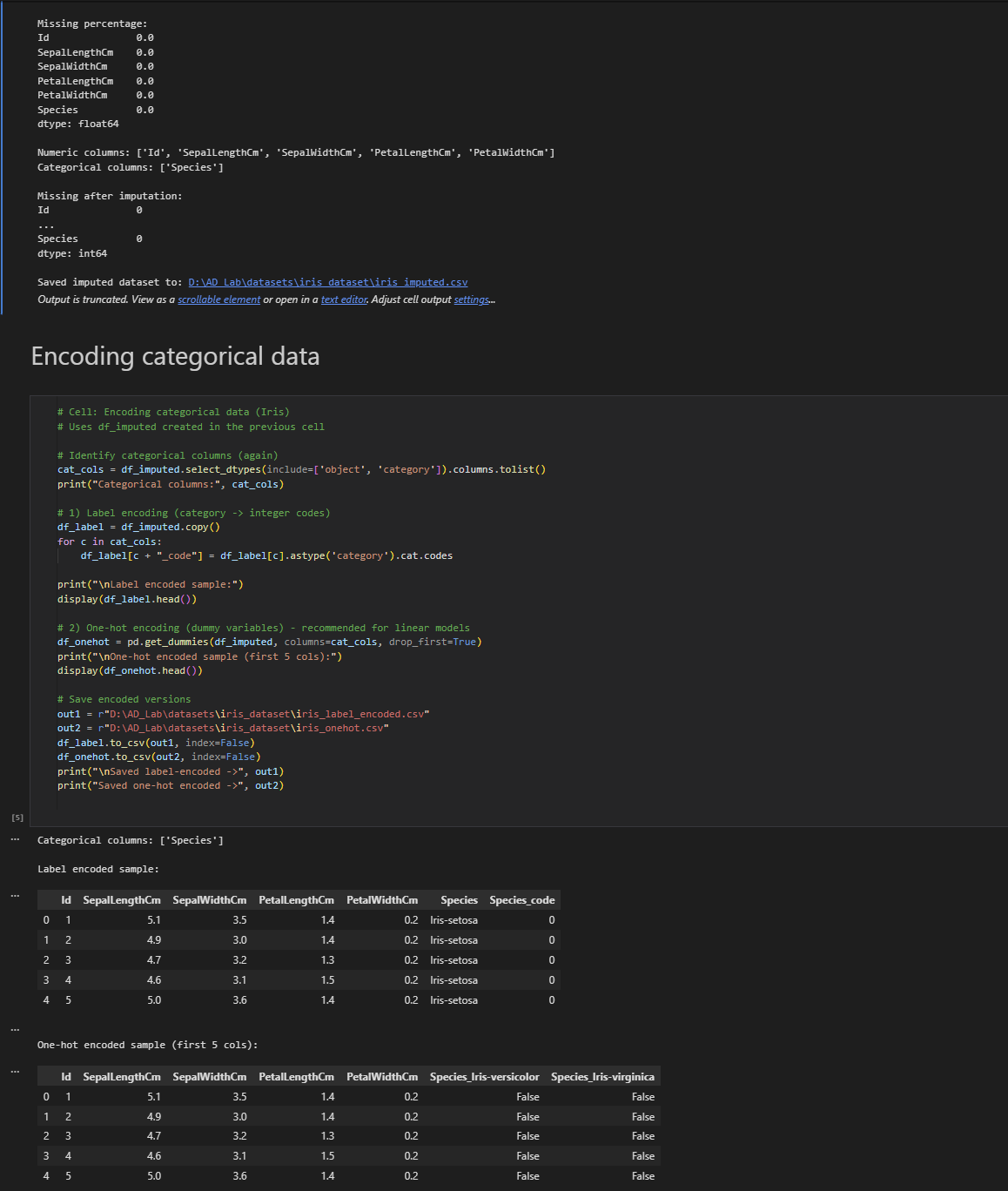
|  |  |  |
| --- | --- | --- |
| S.No | Experiment | Page No |
| 1 | Introduction to Machine Learning | 3-7 |
| 2 | Regression Analysis | 8-13 |
| 3 | Multiple Linear Regression | 14-18 |
| 4 | Classification with Logistic Regression | 19-23 |
| 5 | Decision Trees | 24-27 |
| 6 | Random Forests | 28-33 |
| 7 | Support Vector Machines (SVM) | 34-37 |
| 8 | K-Means Clustering | 38-42 |
| 9 | Principal Component Analysis (PCA) | 43-47 |
| 10 | Neural Networks with TensorFlow/Keras | 48-50 |
| 11 | Convolutional Neural Networks (CNN) | 51-55 |
| 12 | Capstone Application | 56-60 |

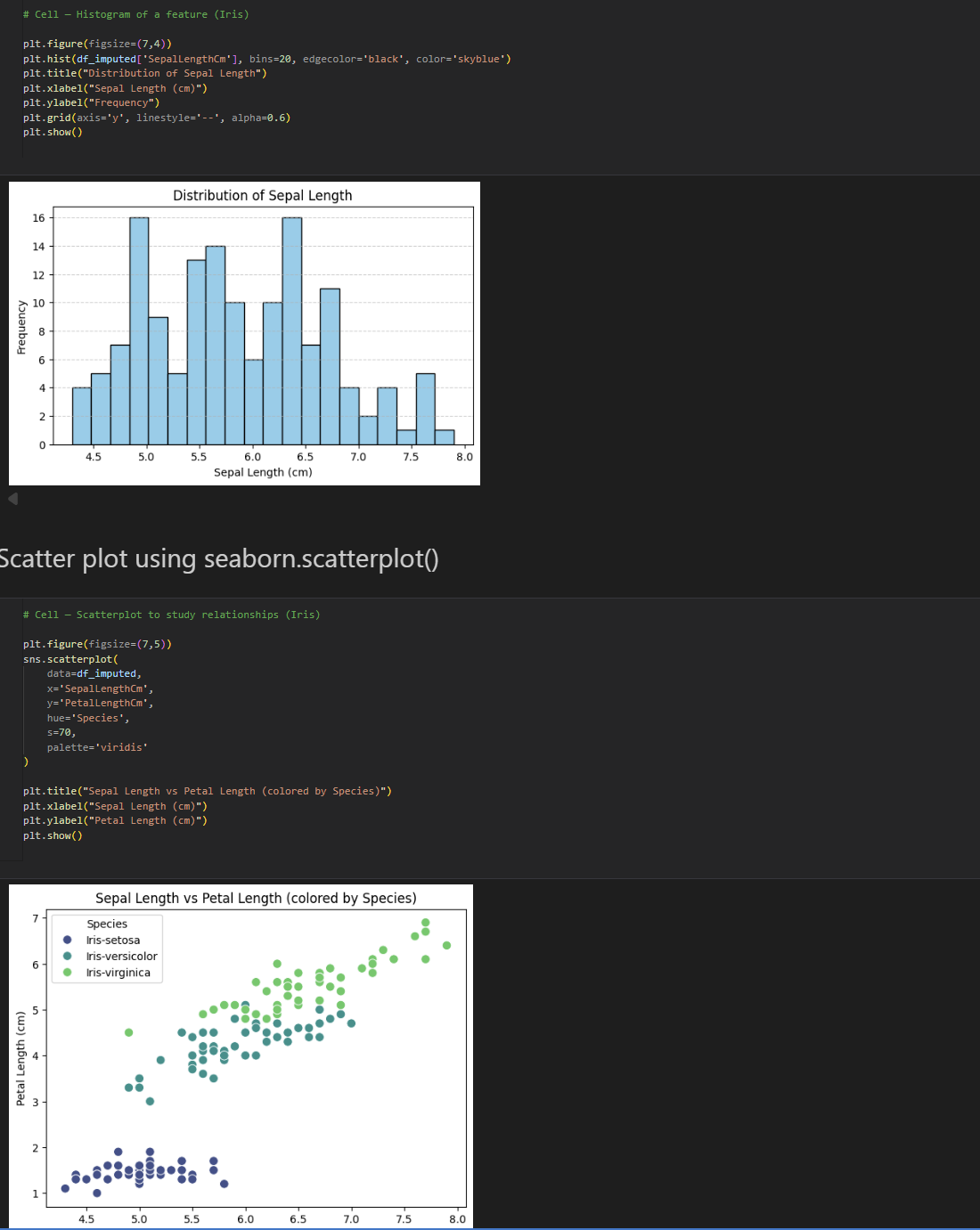
Week 1: Introduction to Machine

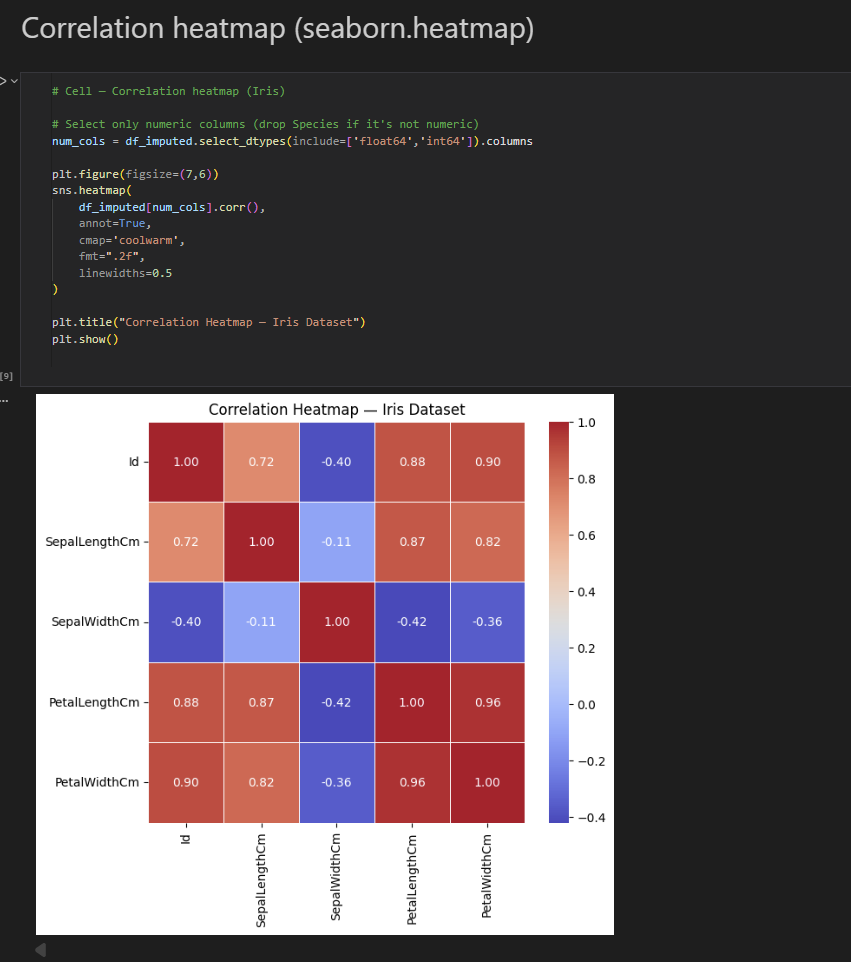
Learning Objective: Understand the basics of machine learning and Python / R, etc





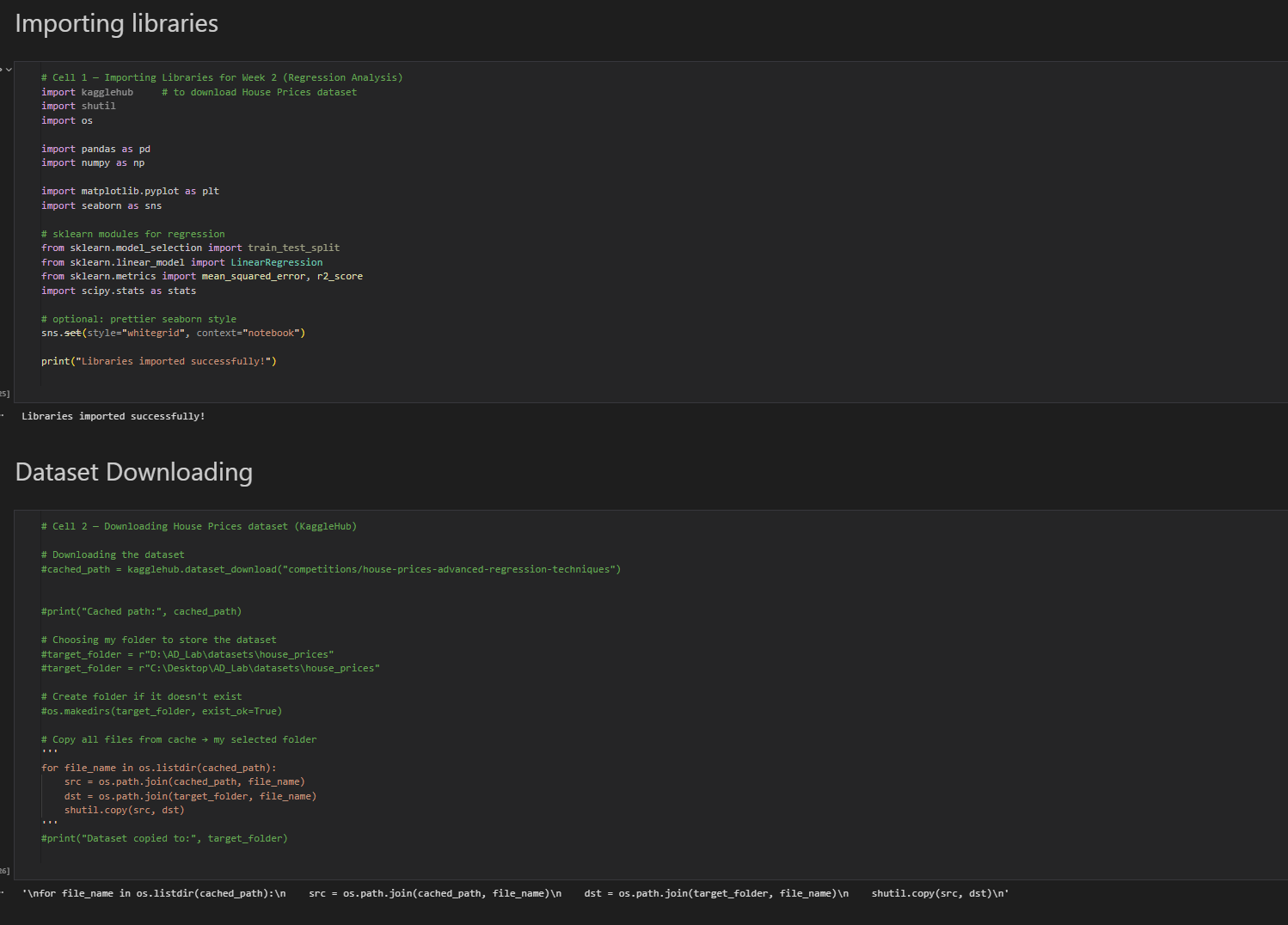


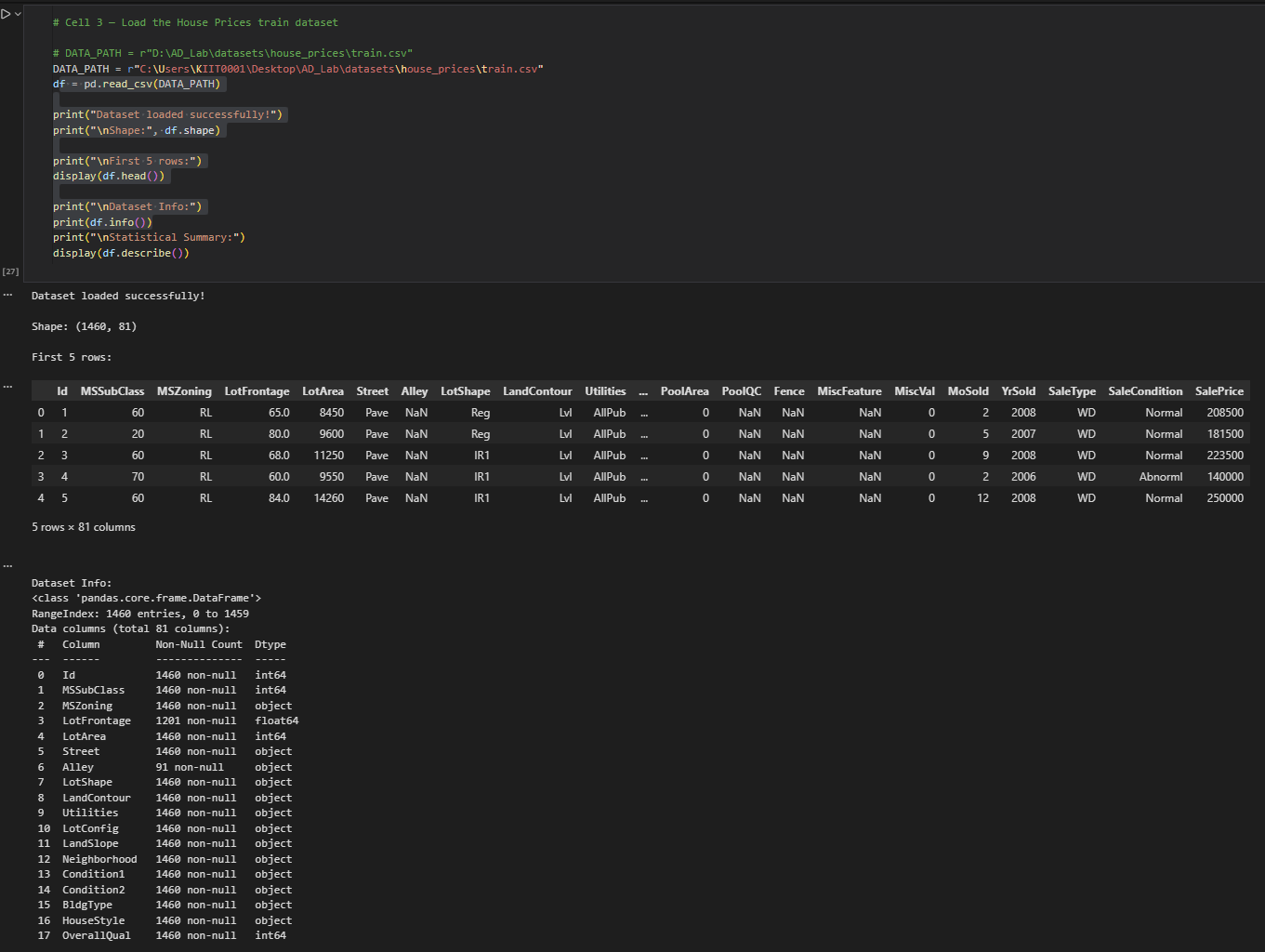


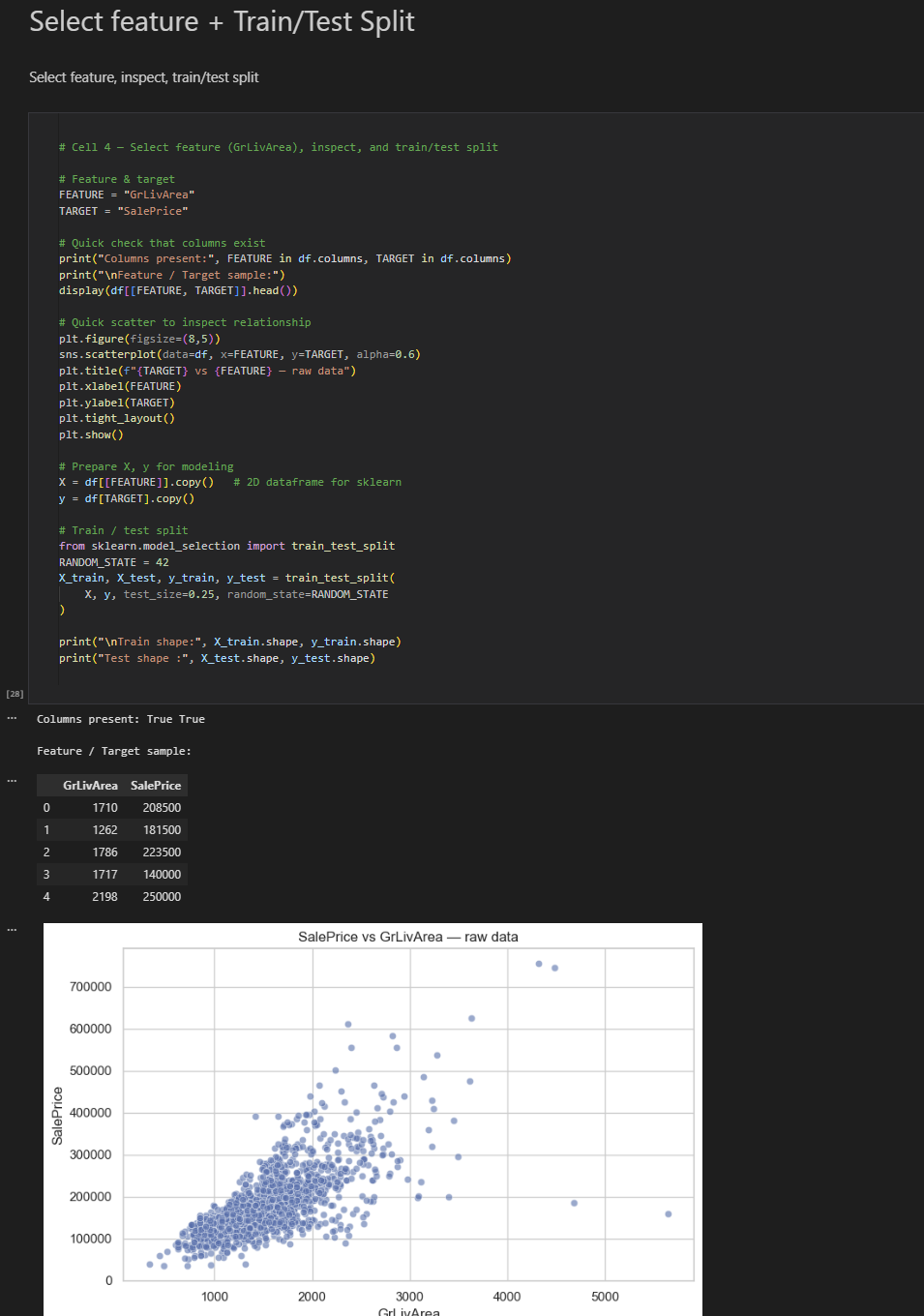


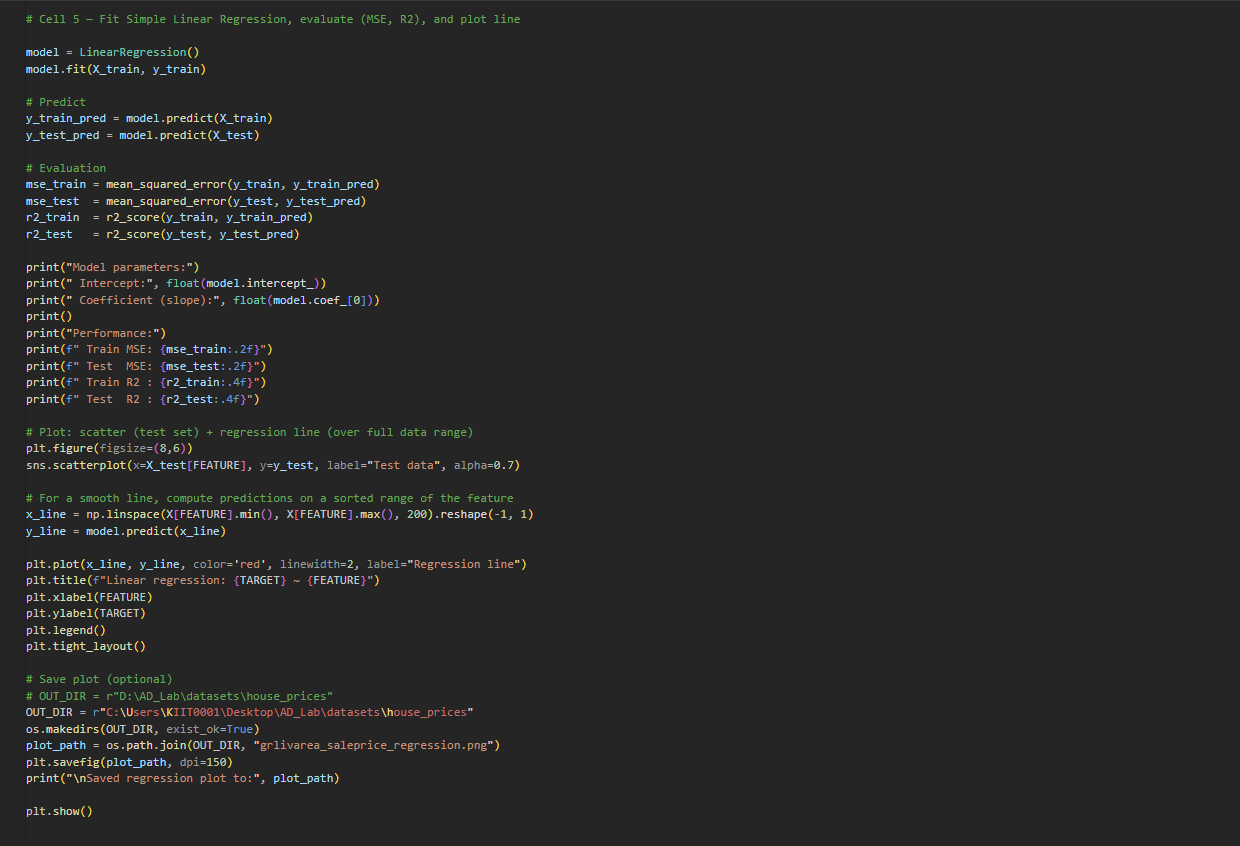
Week 2: Regression Analysis

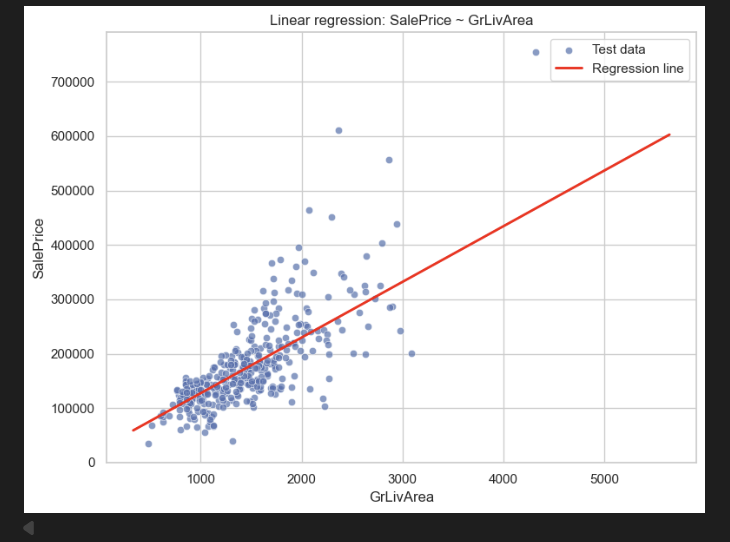
# Objective: Learn simple linear regression for predictive analysis

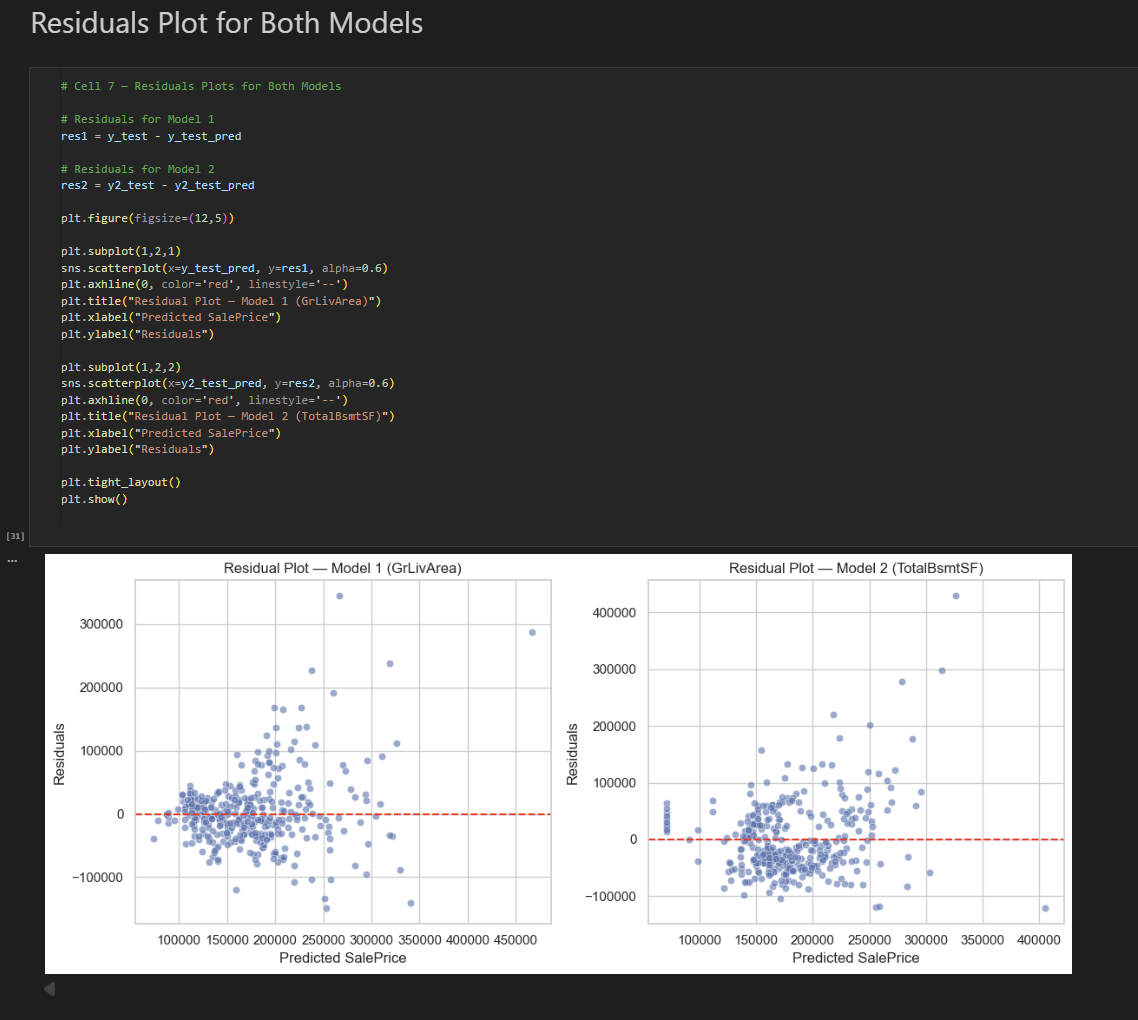


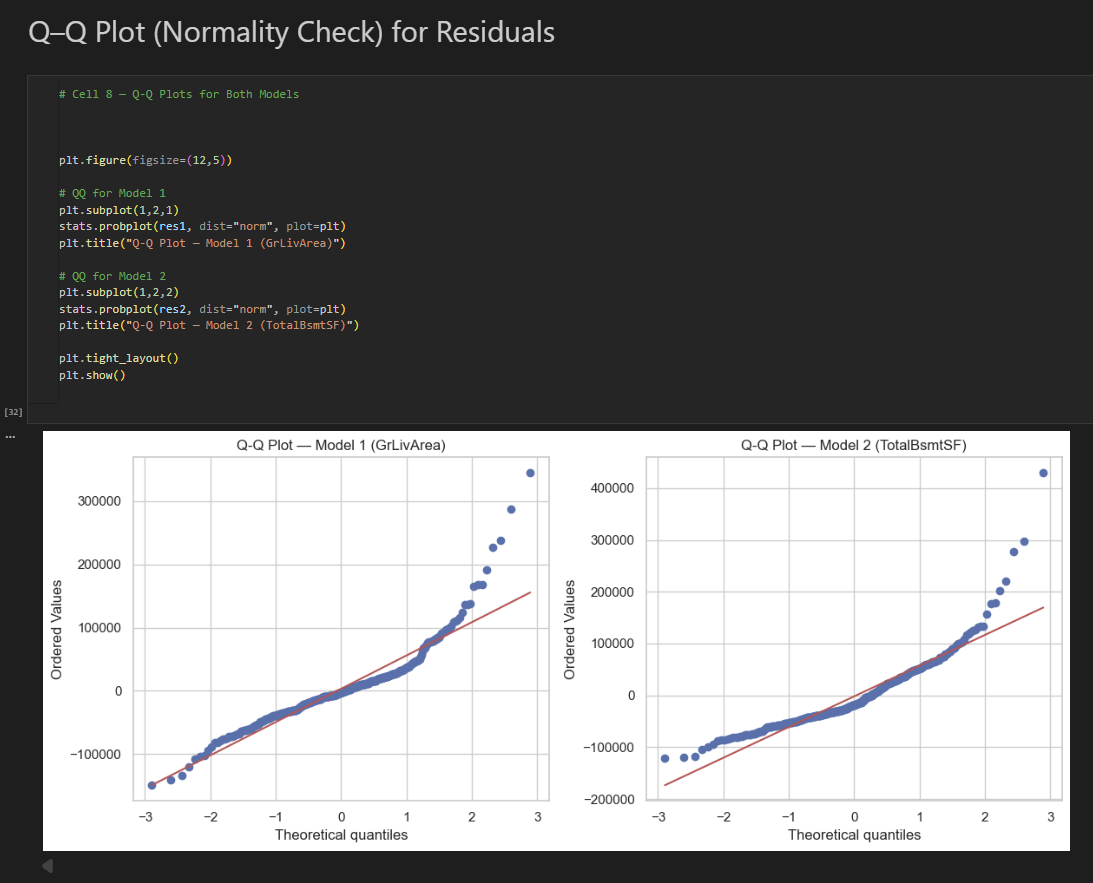






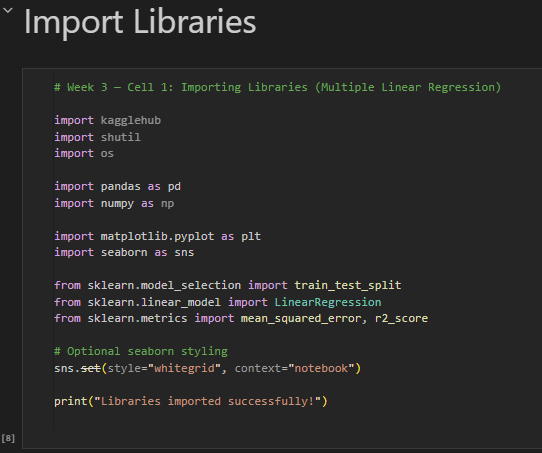


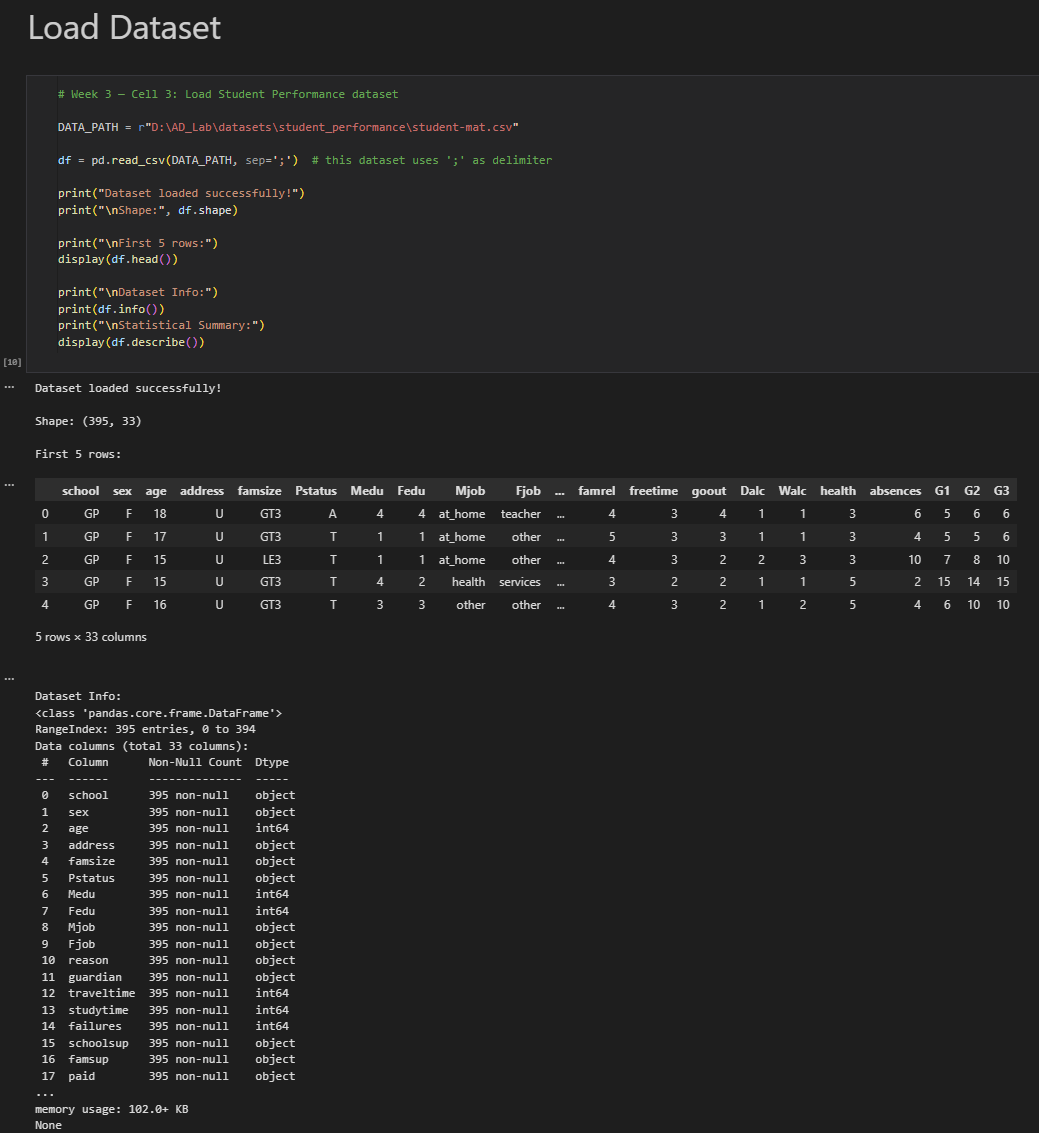


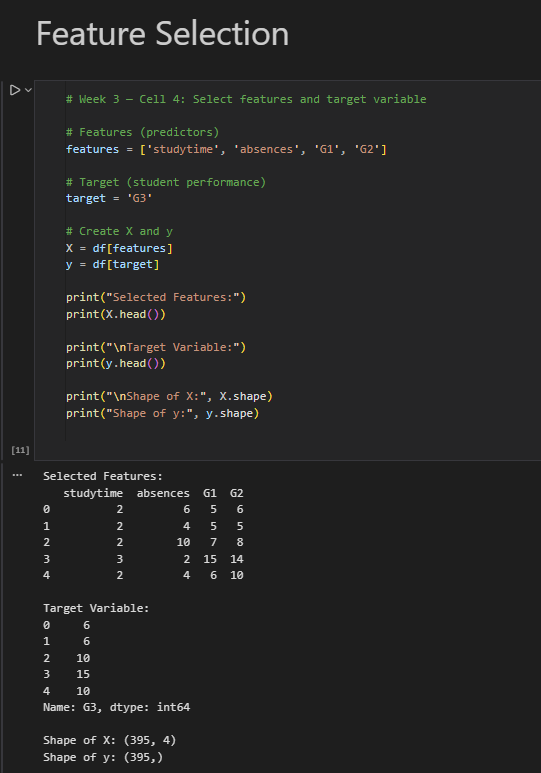


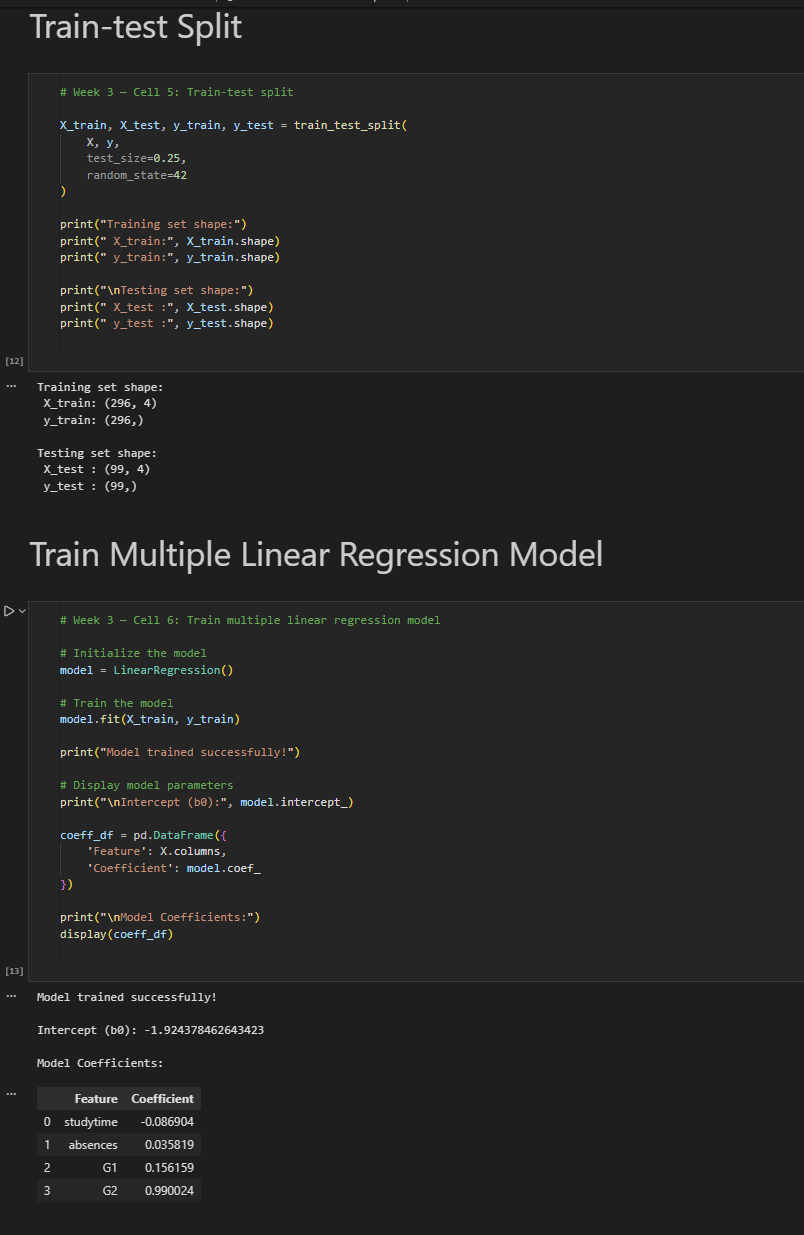
Week 3: Multiple Linear Regression

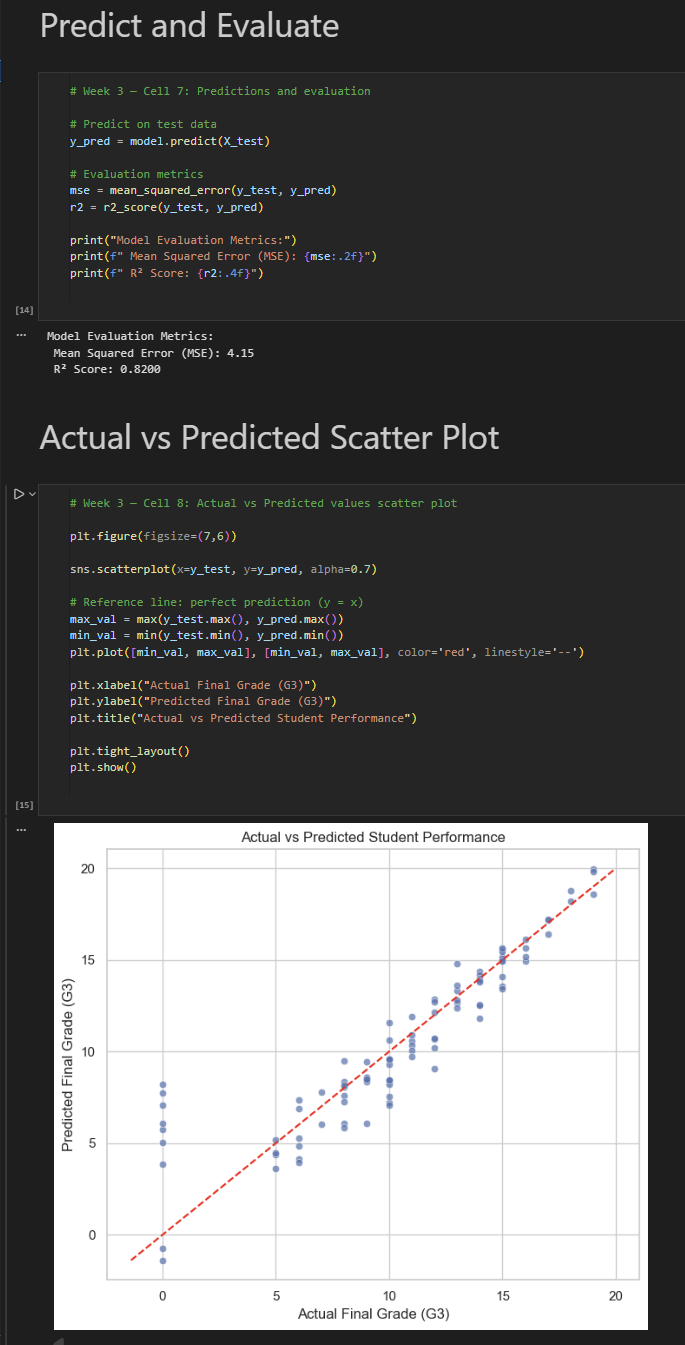
# Objective: Extend regression analysis to handle multiple predictors.











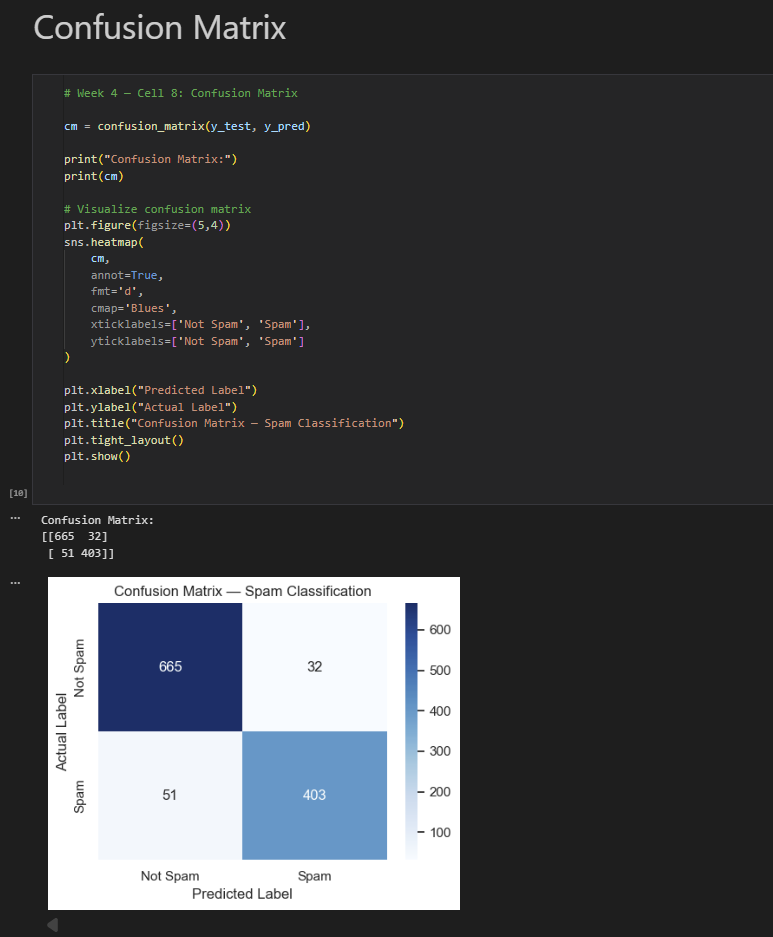
Week 4: Classification with Logistic Regression

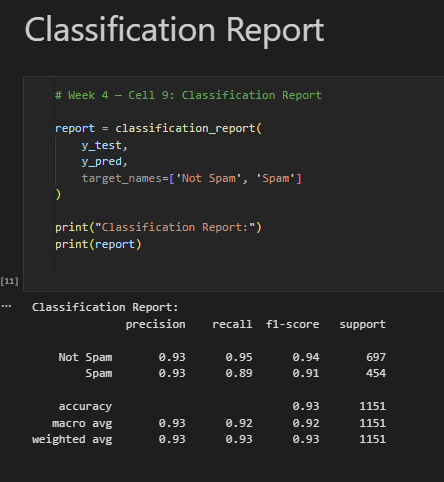
# Objective:Understand binary classification using logistic regression.





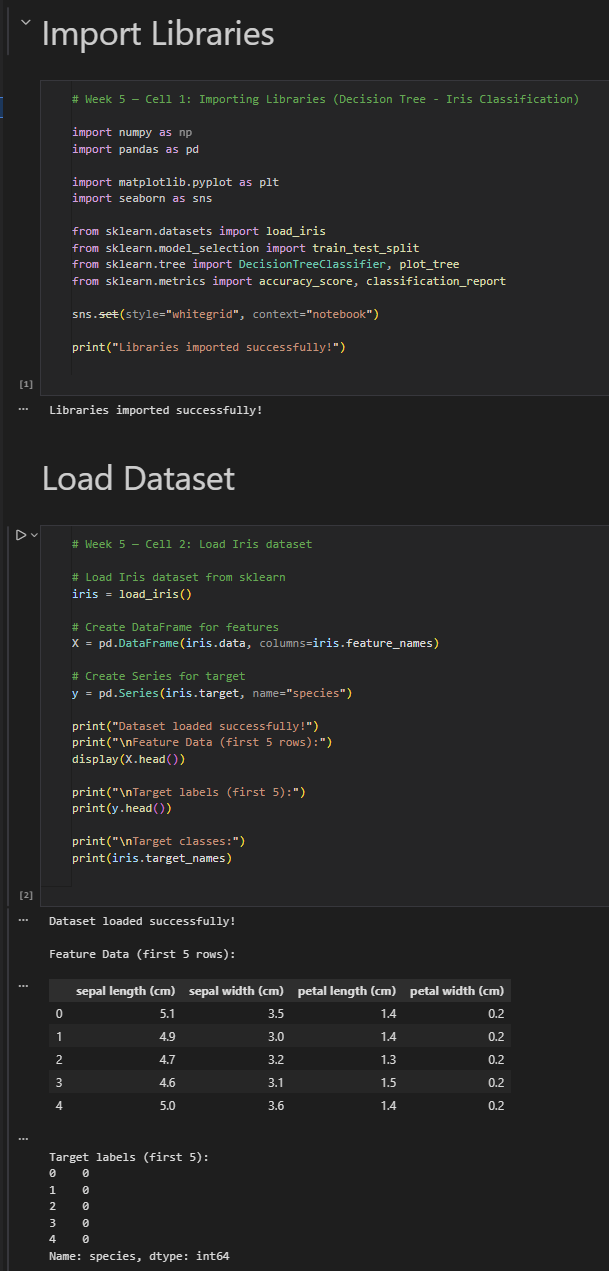


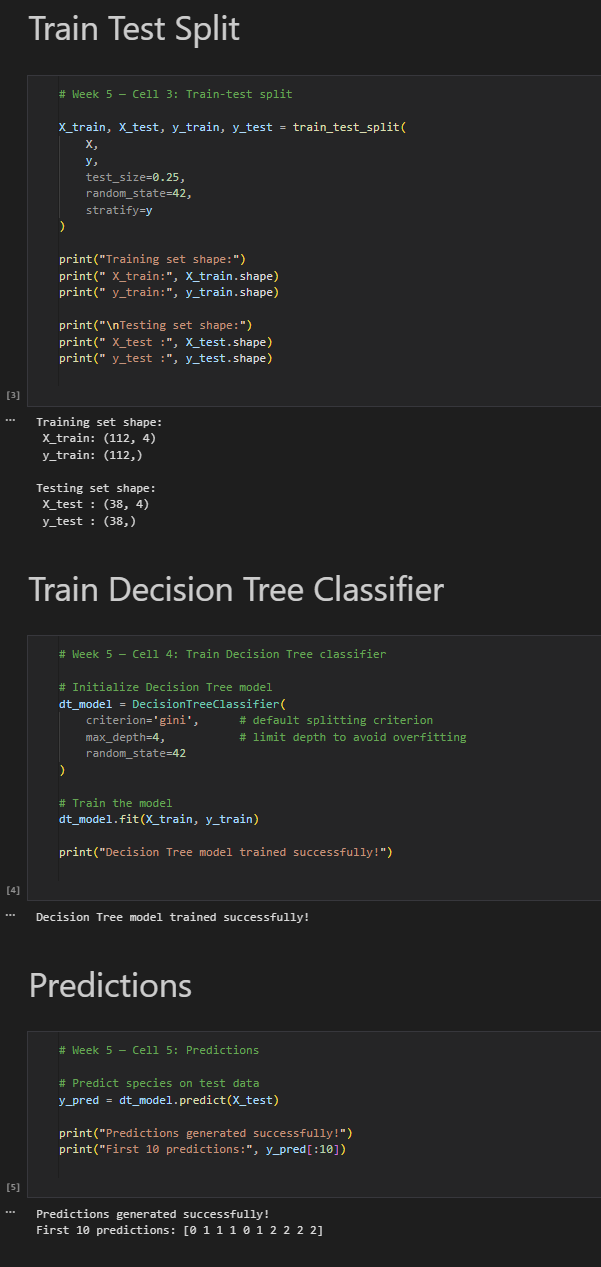


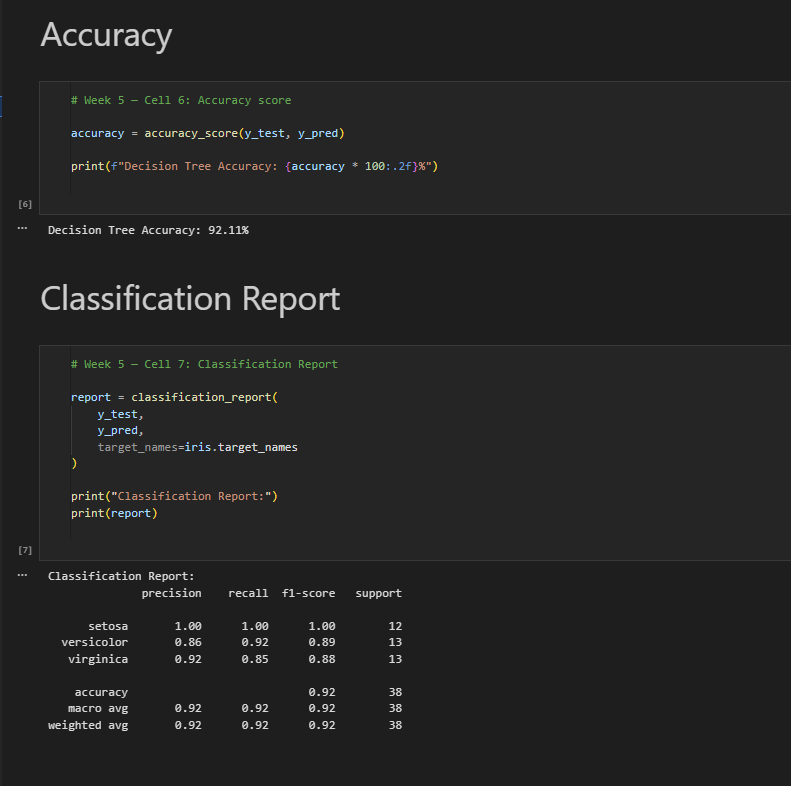


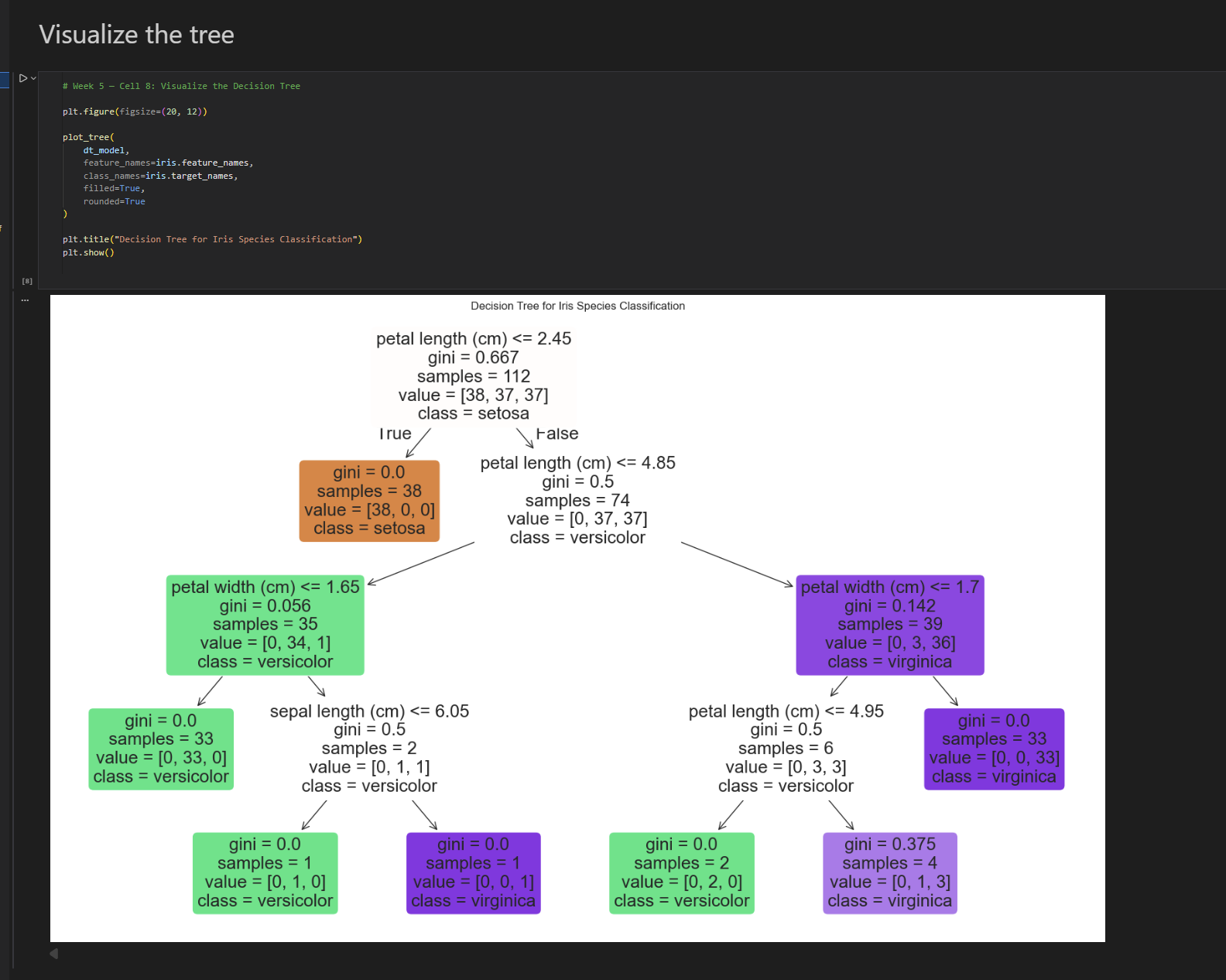
Week 5: Decision Trees

Objective: Understand binary classification using logistic regression.



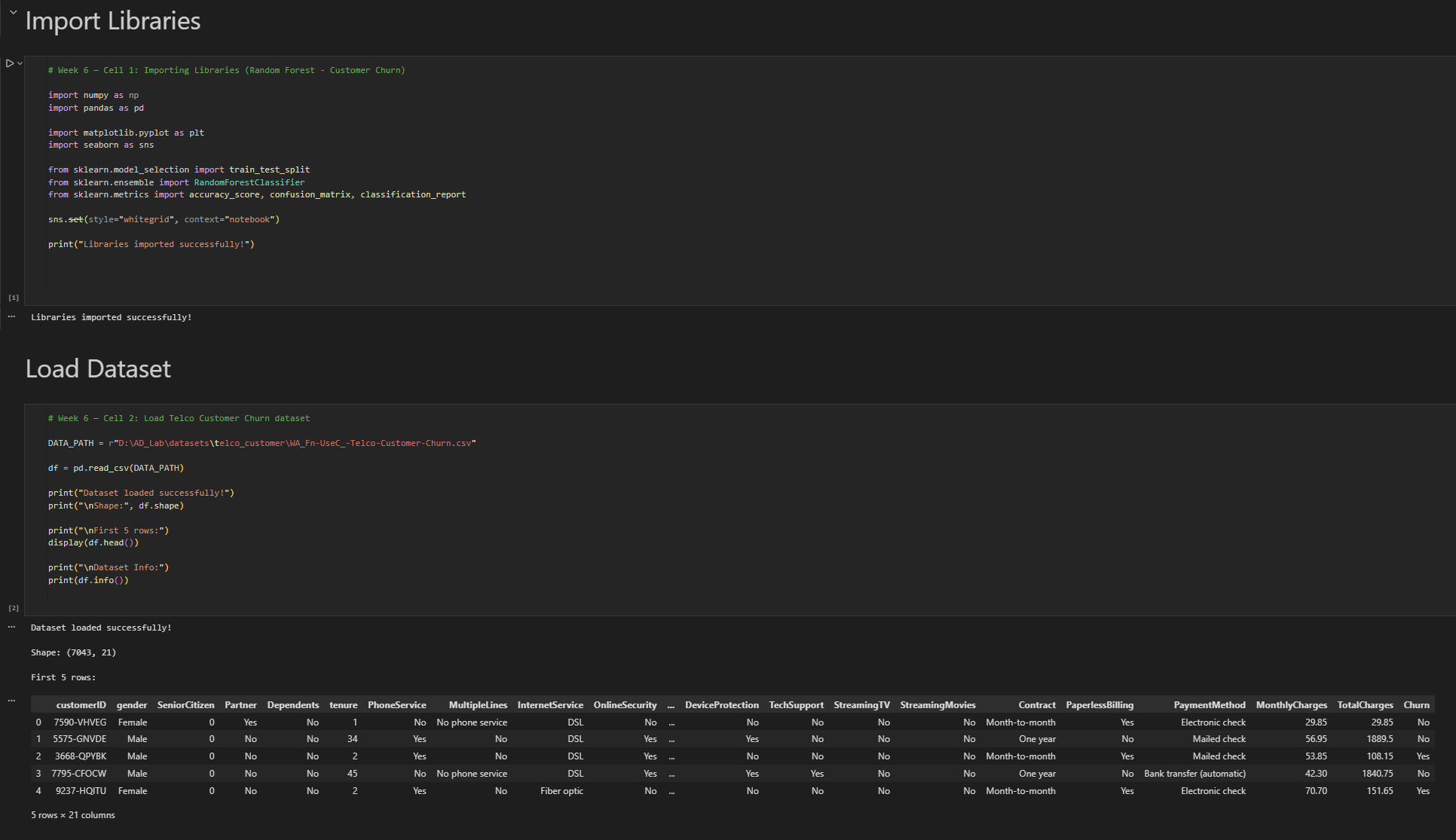


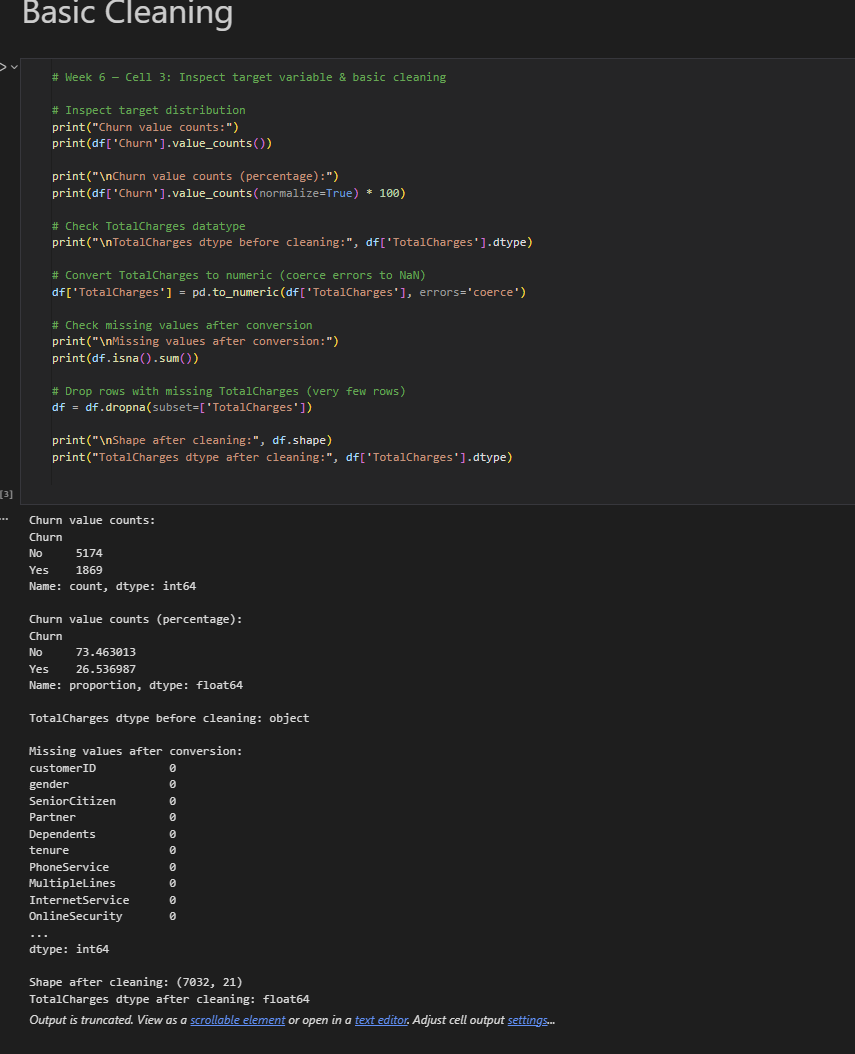


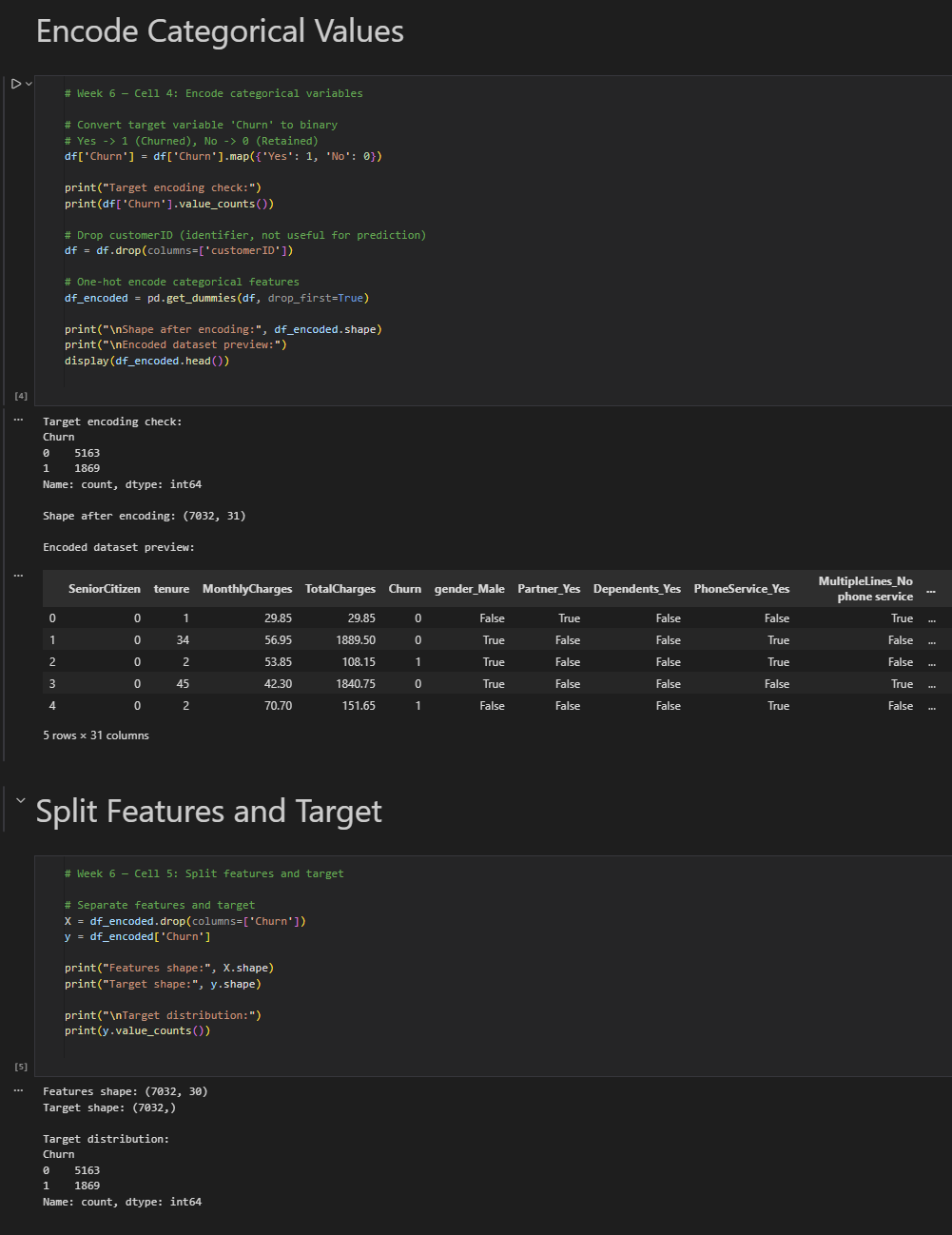


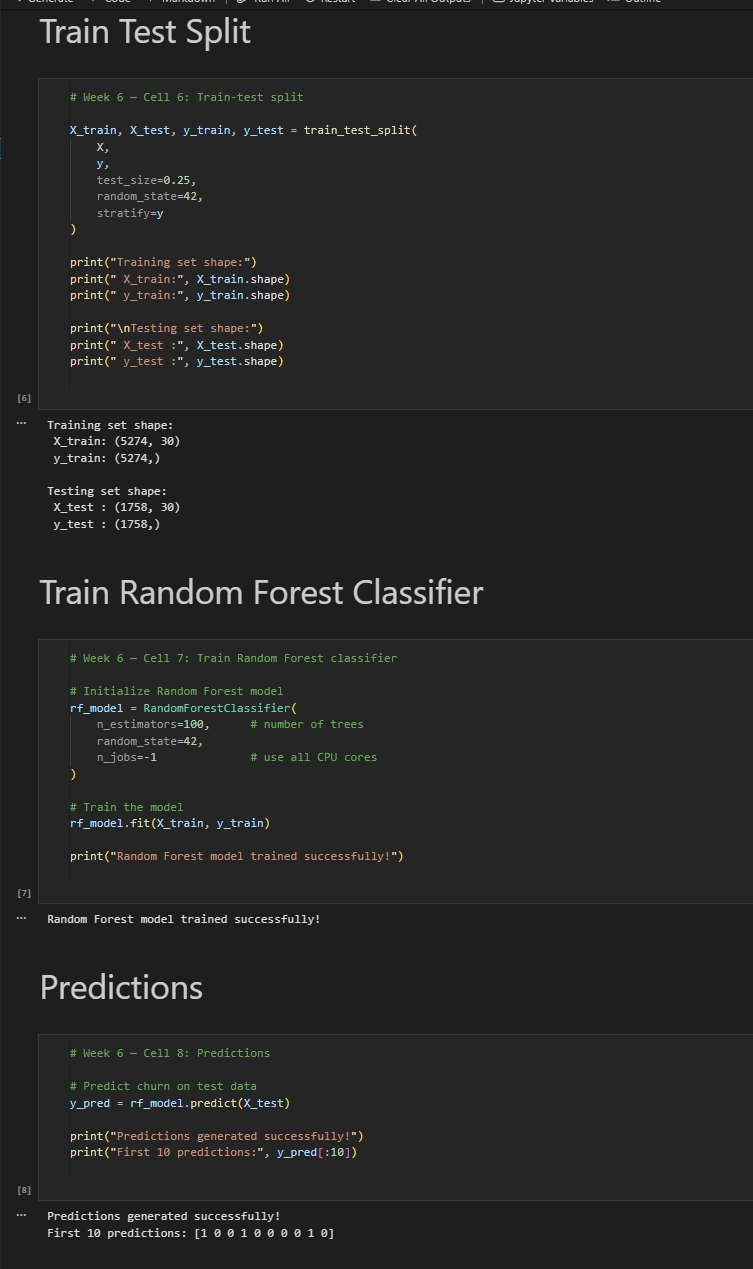
Week 6: Random Forests

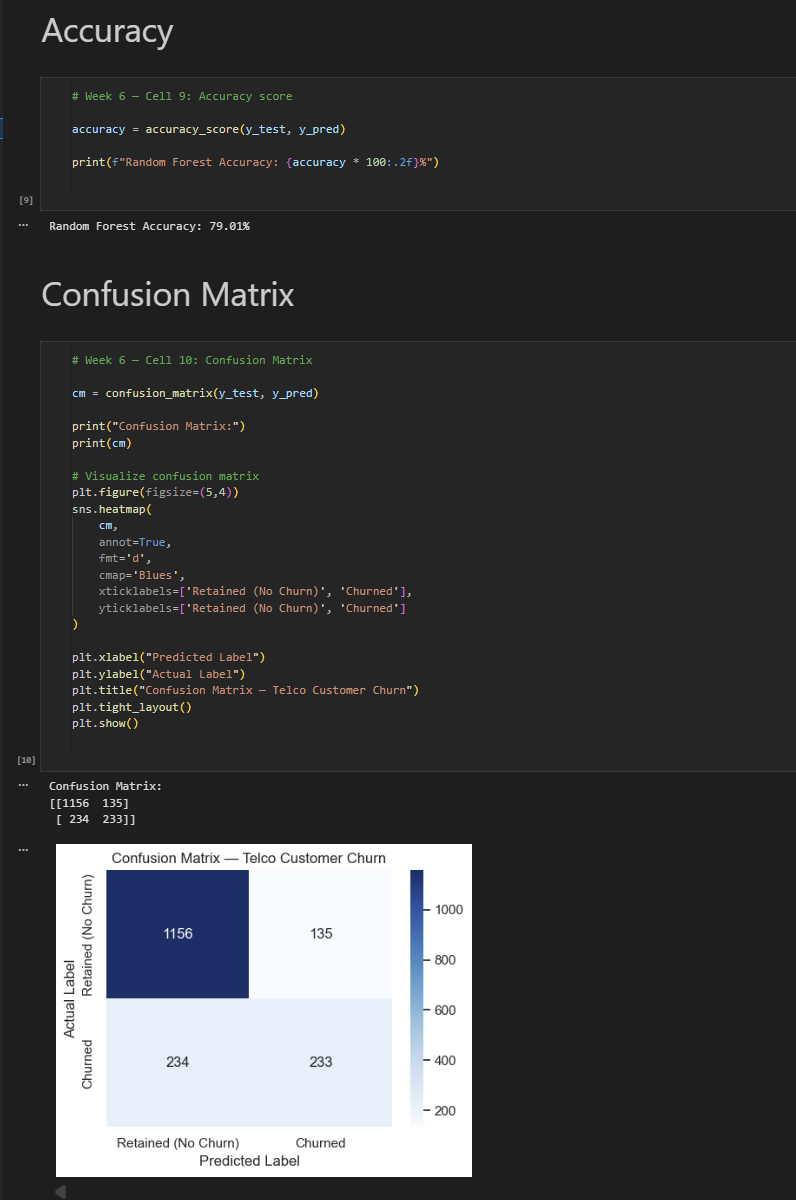
# Objective: Understand ensemble methods using randomforests.

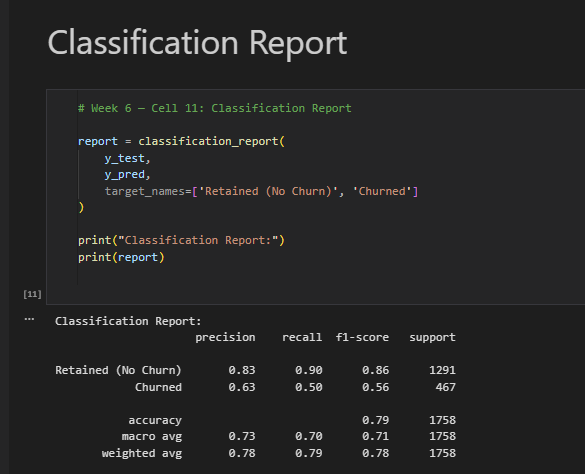






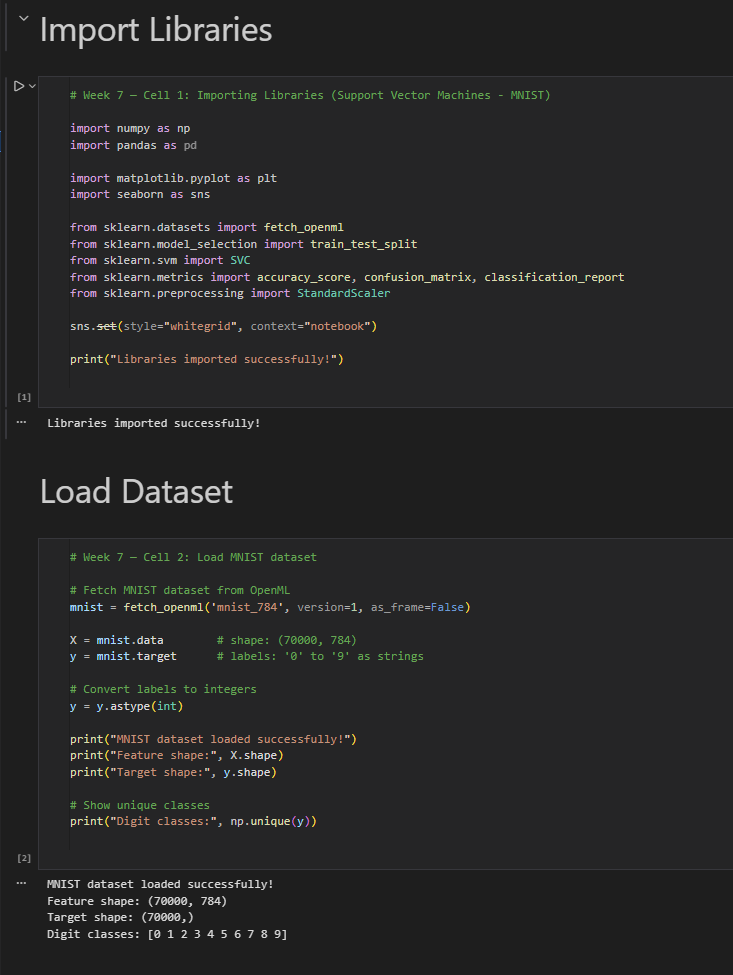




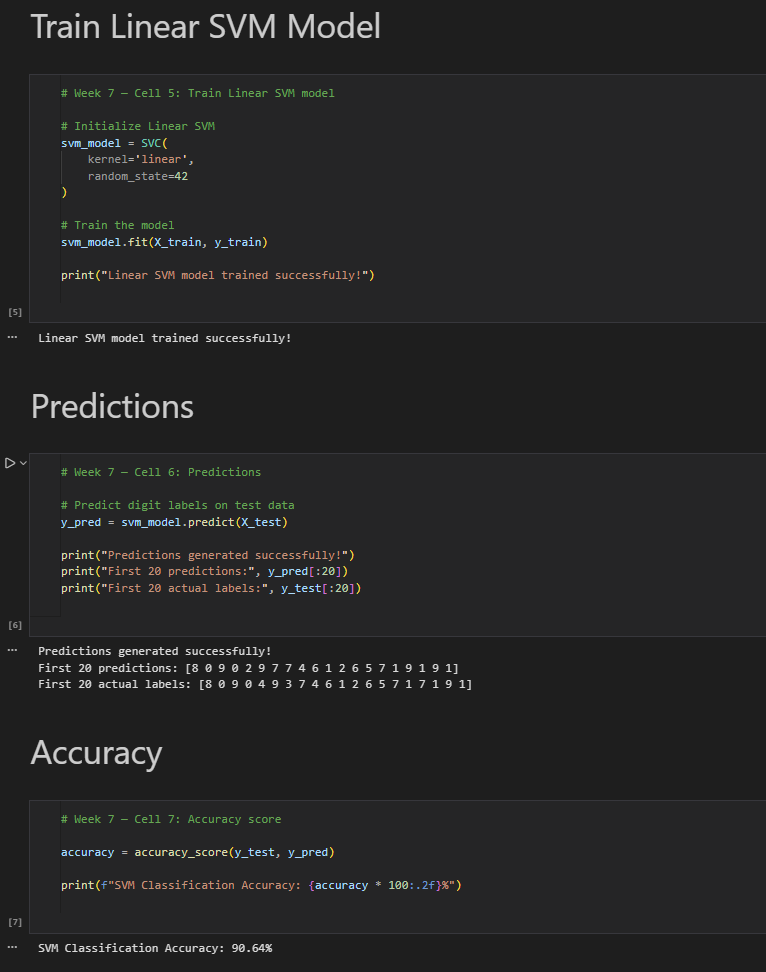


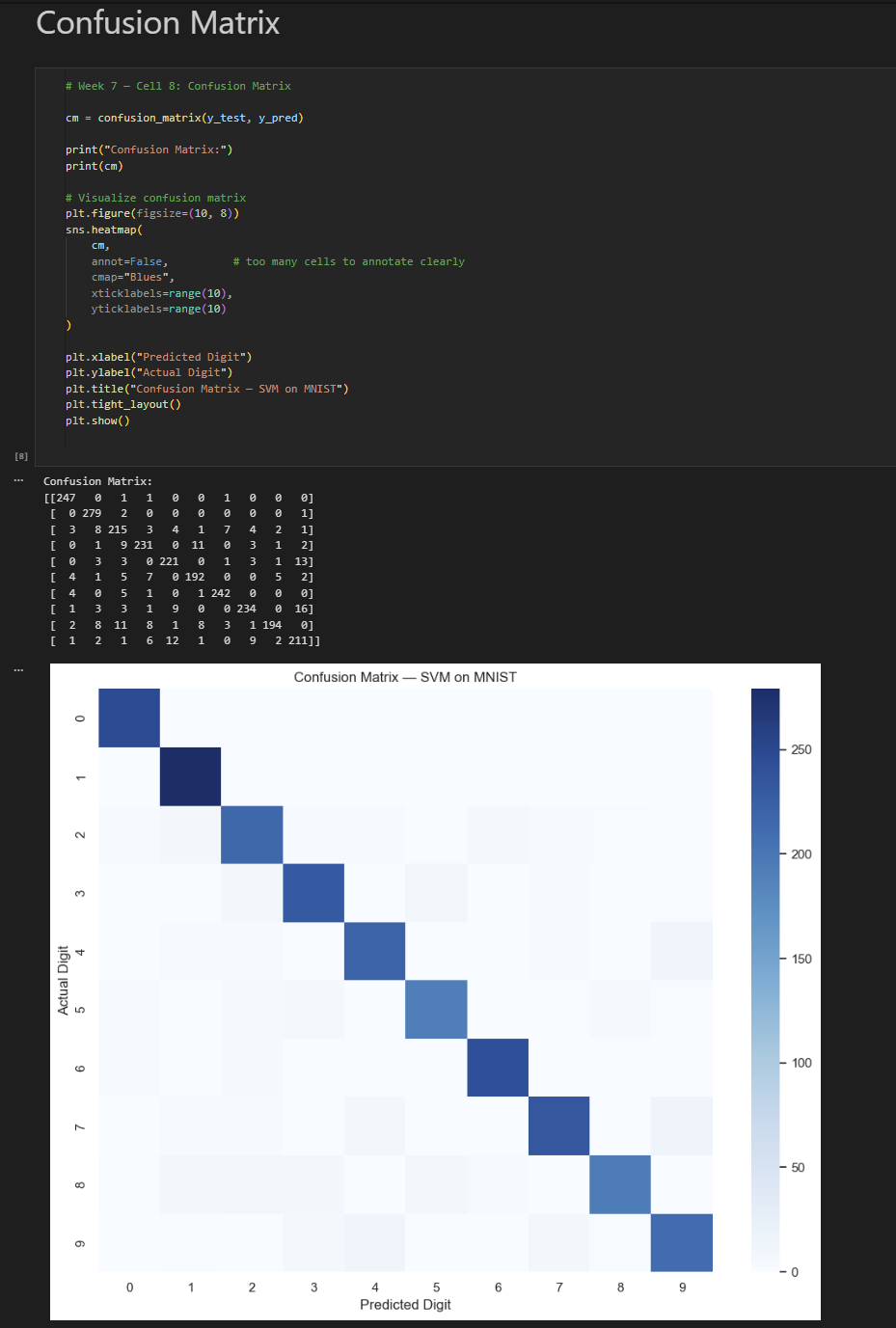
Week 7: Support Vector Machines (SVM)

# Objective: Explore SVM for linear and non-linear classification



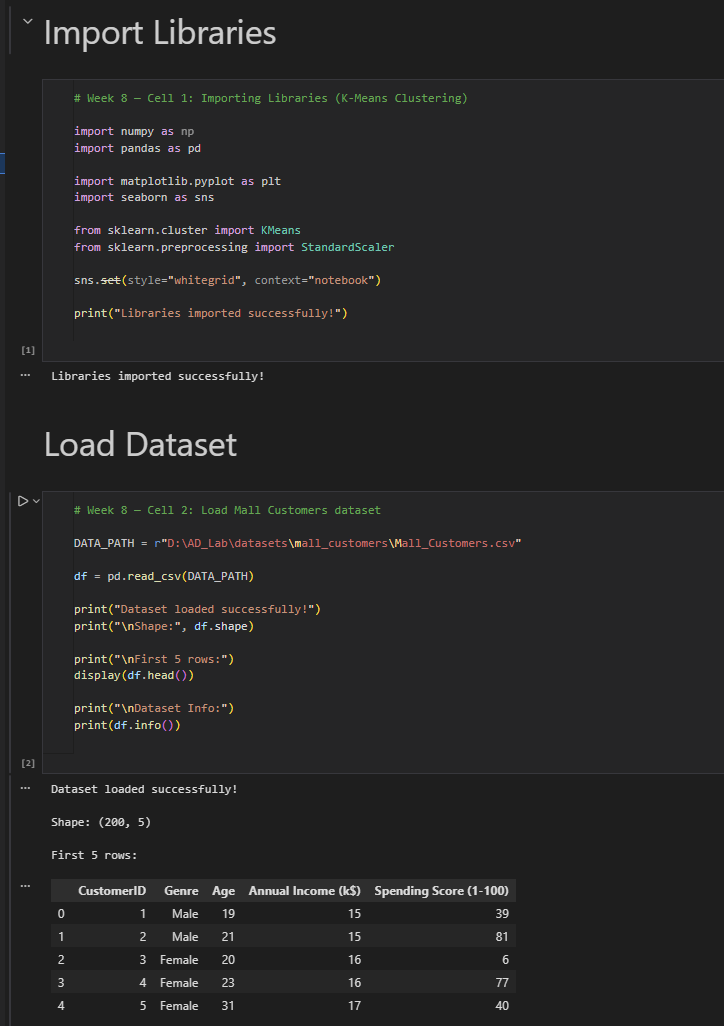


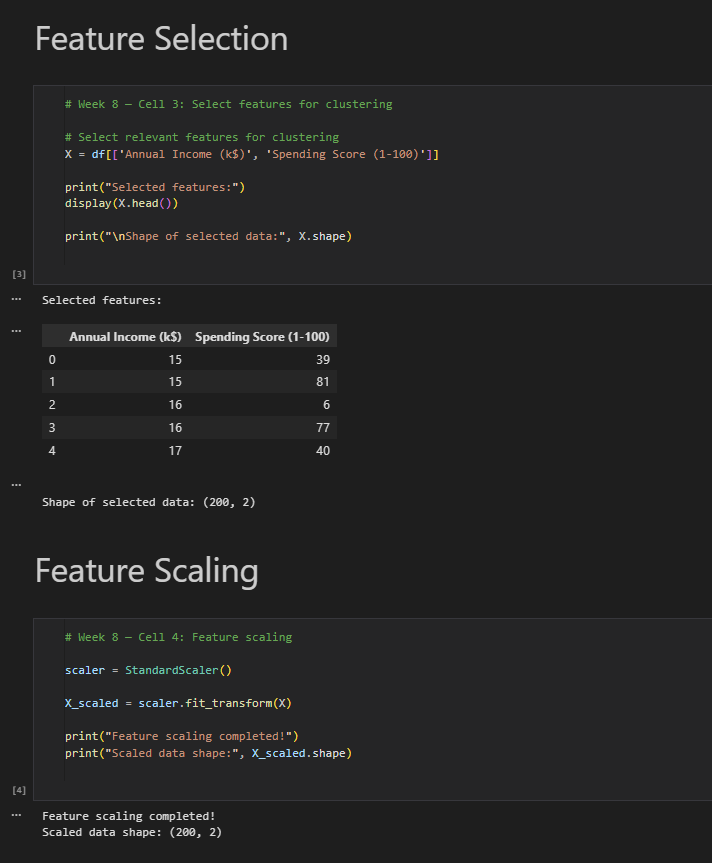


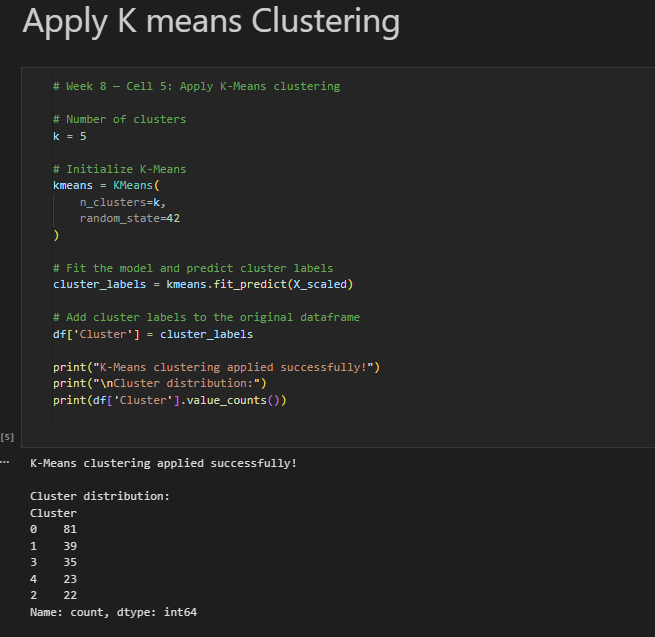


Week 8: K-Means Clustering

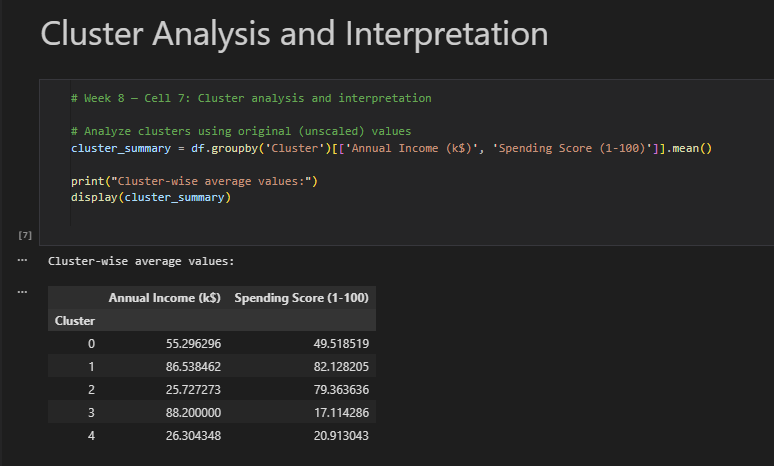
# Objective: Learn unsupervised learning through clustering.





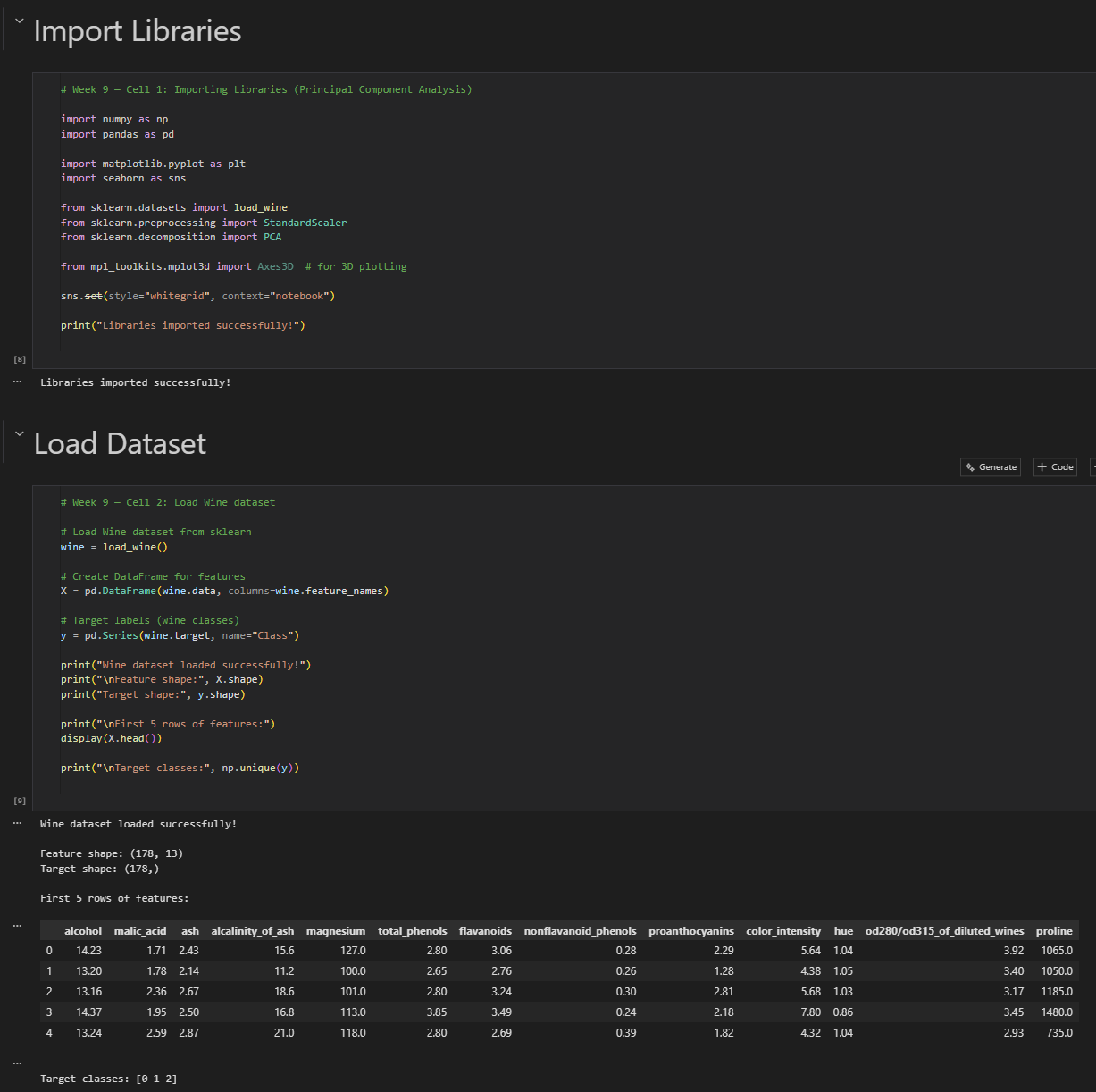


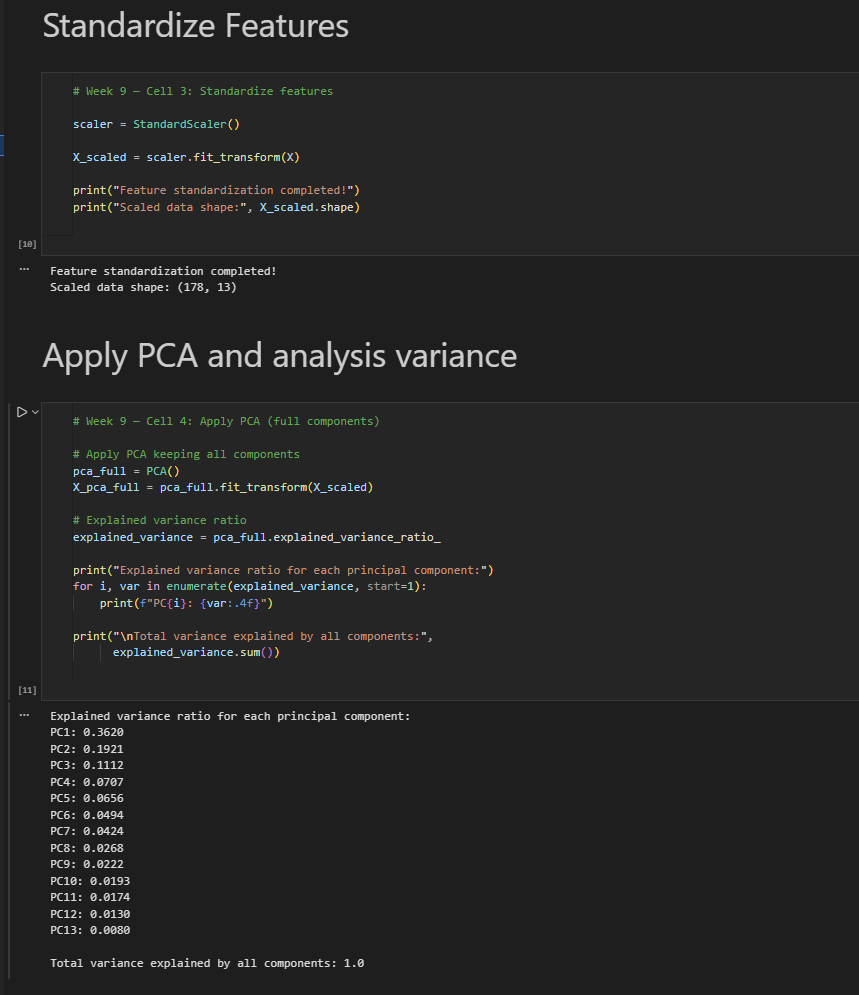


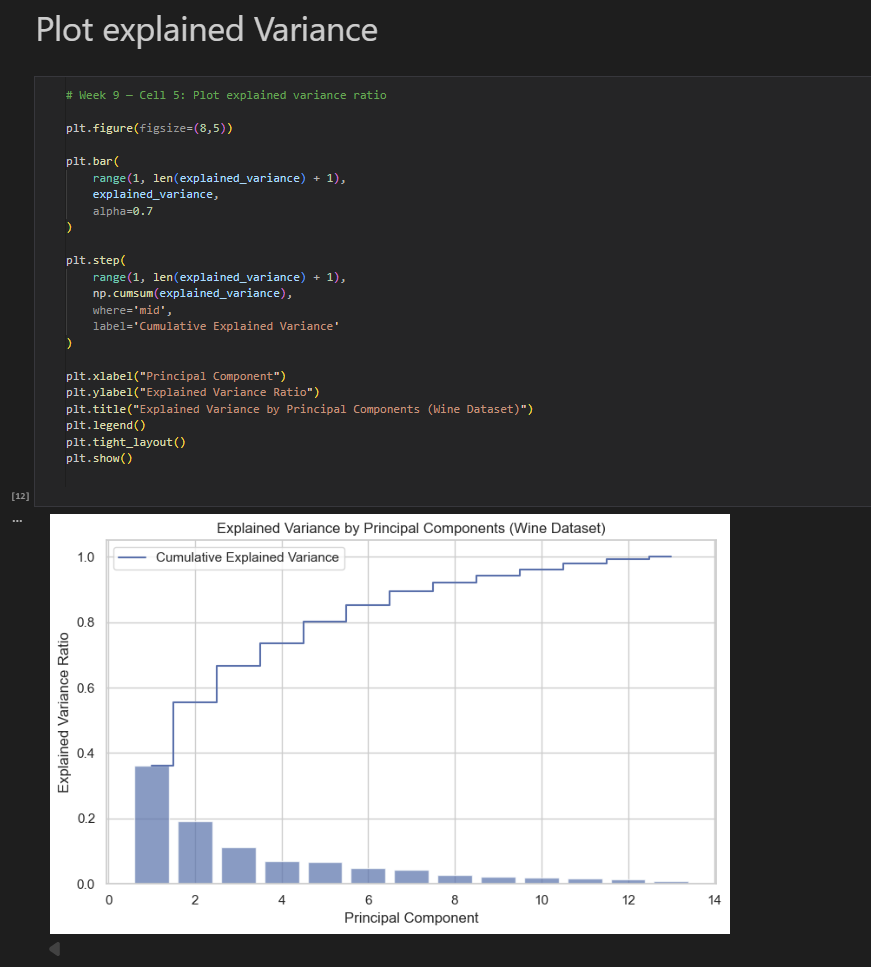


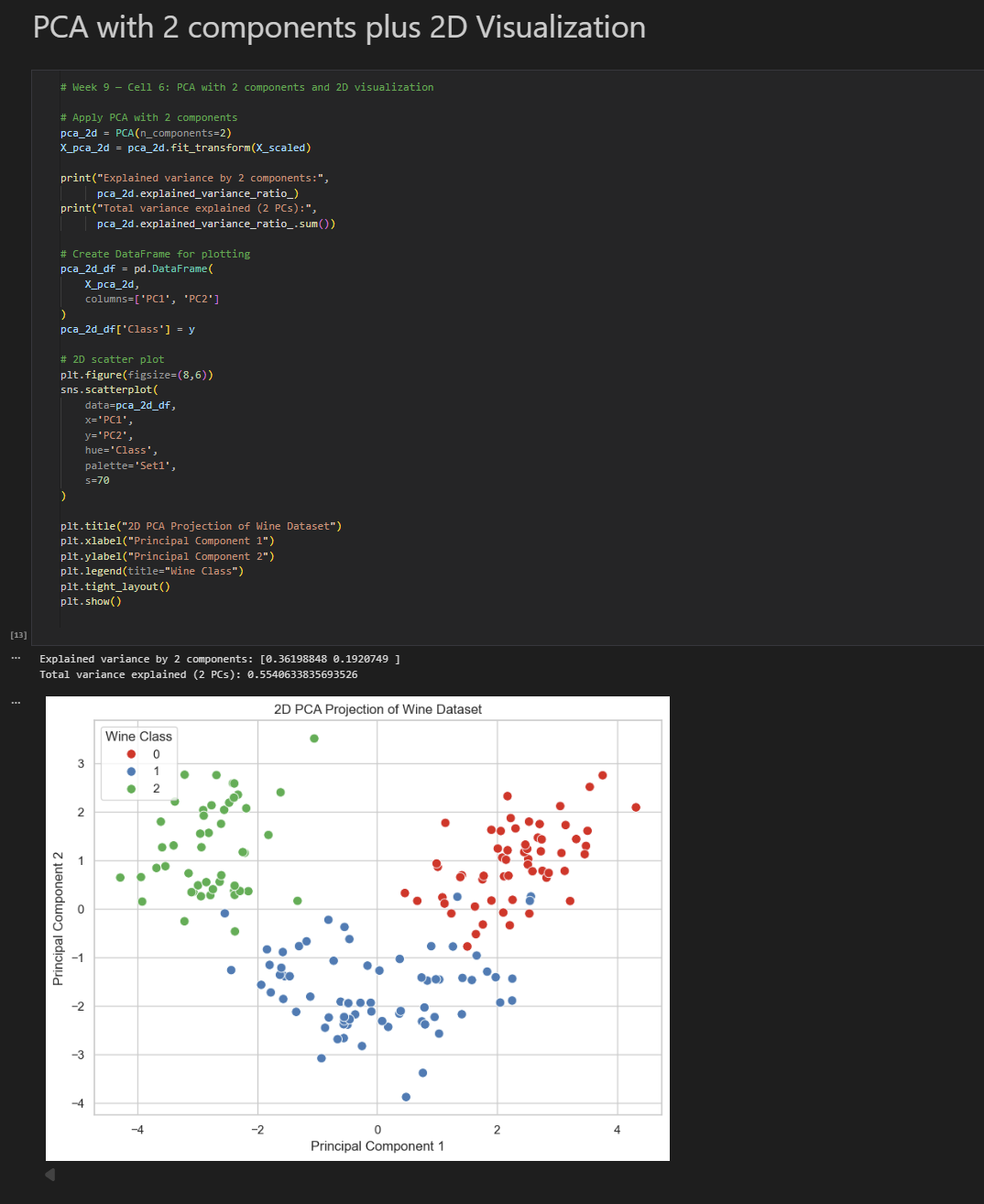
Week 9: Principal Component Analysis (PCA)

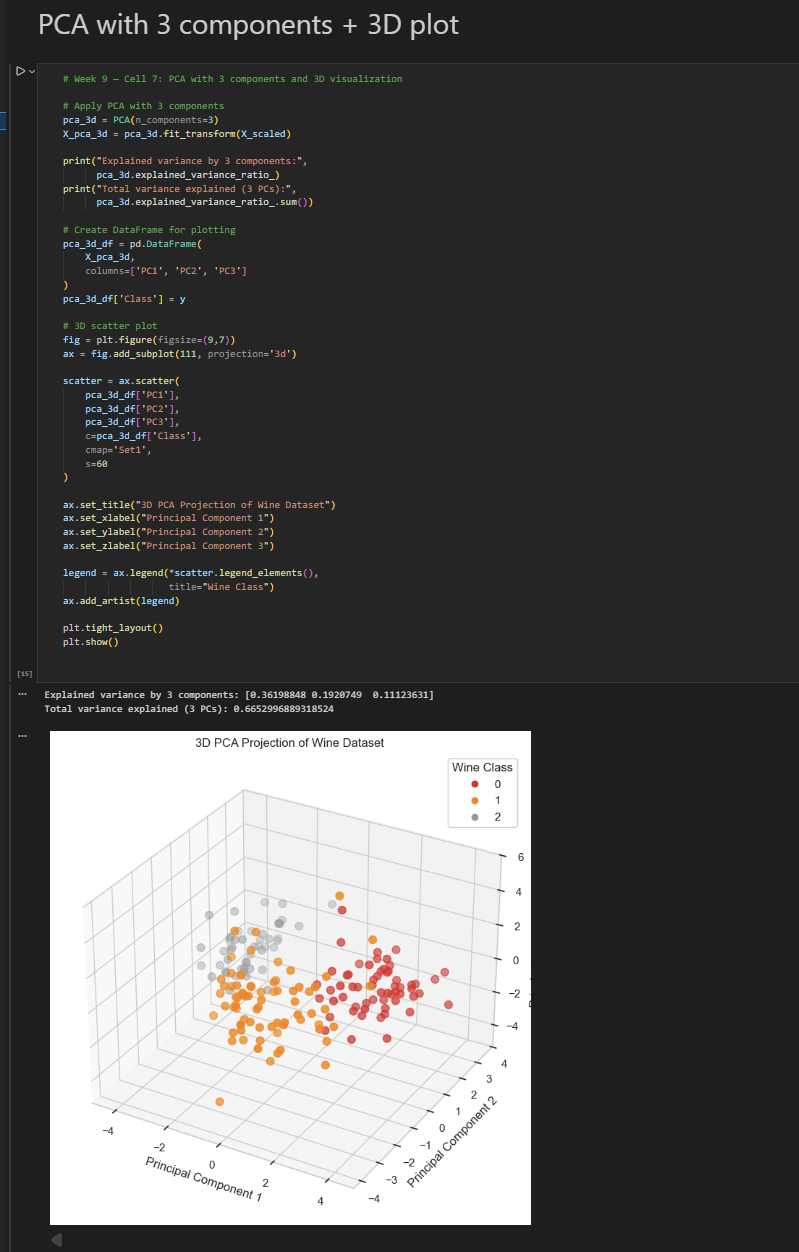
# Objective: Reduce dimensionality using PCA.





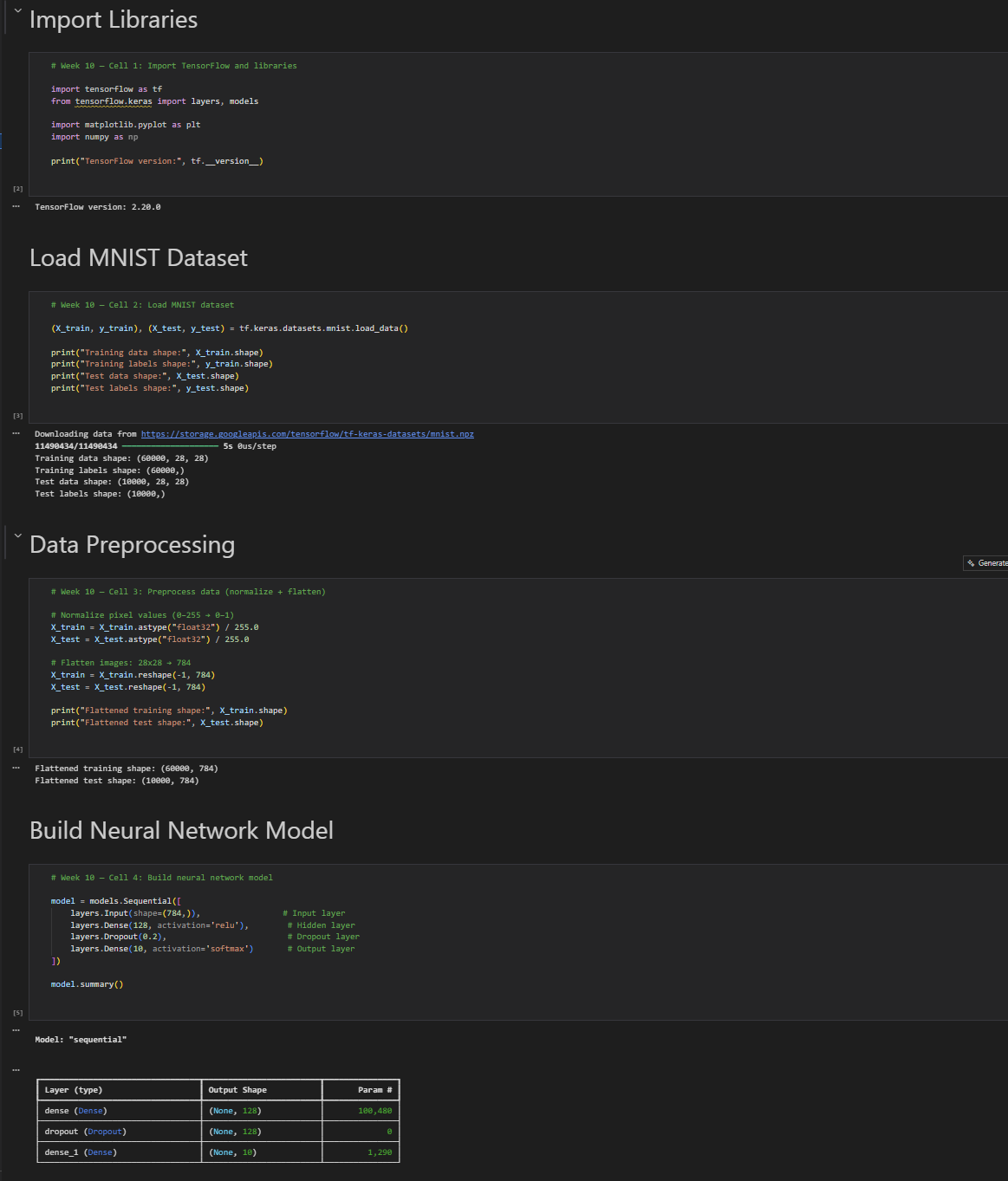


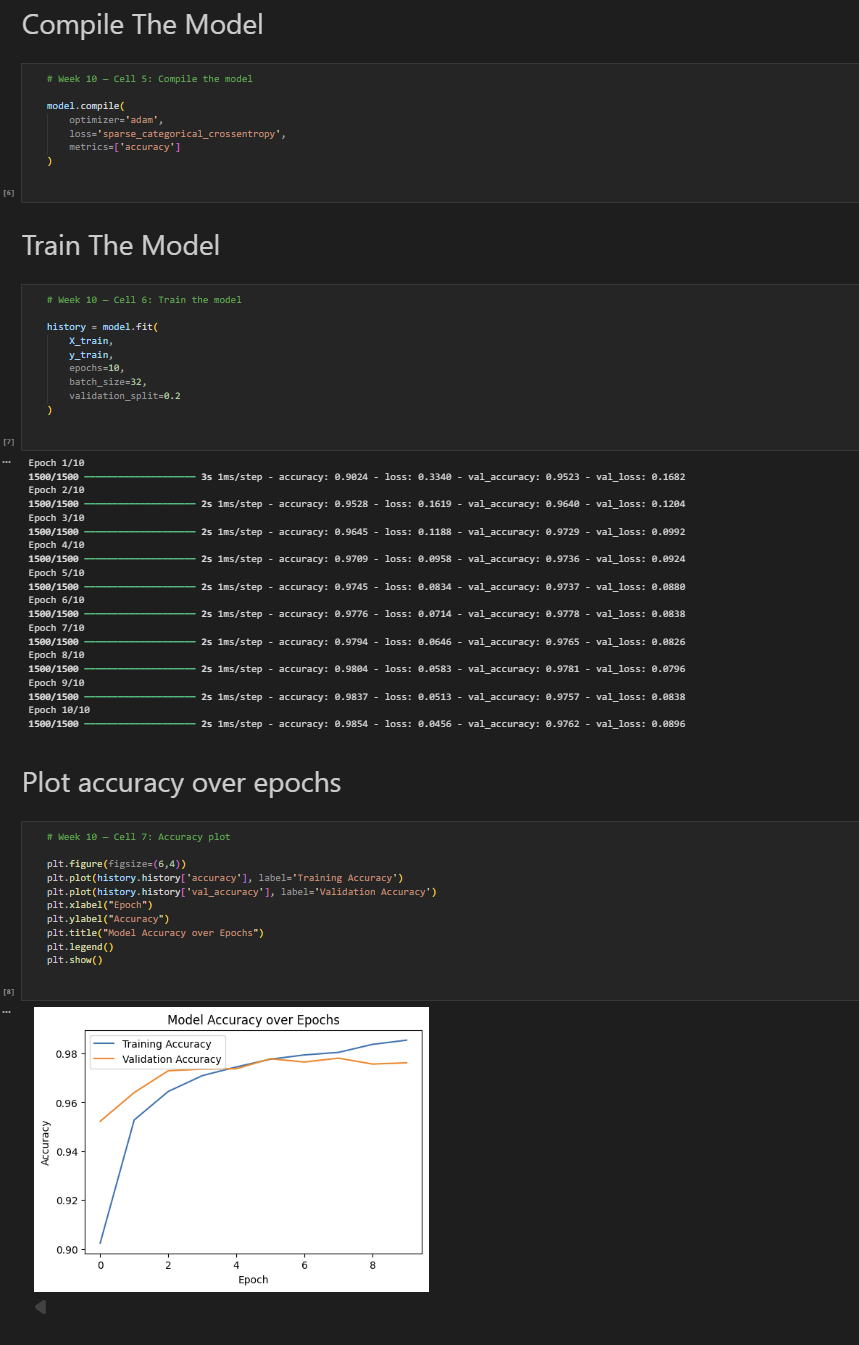


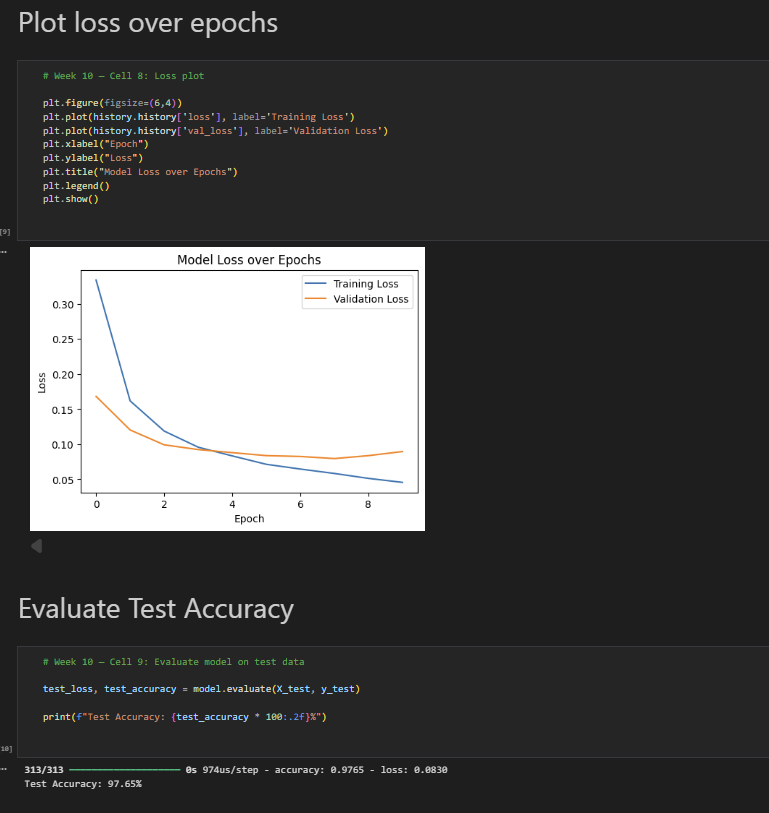


Week 10: Neural Networks with TensorFlow/Keras

# Objective: Introduce deep learning basics with neural networks.



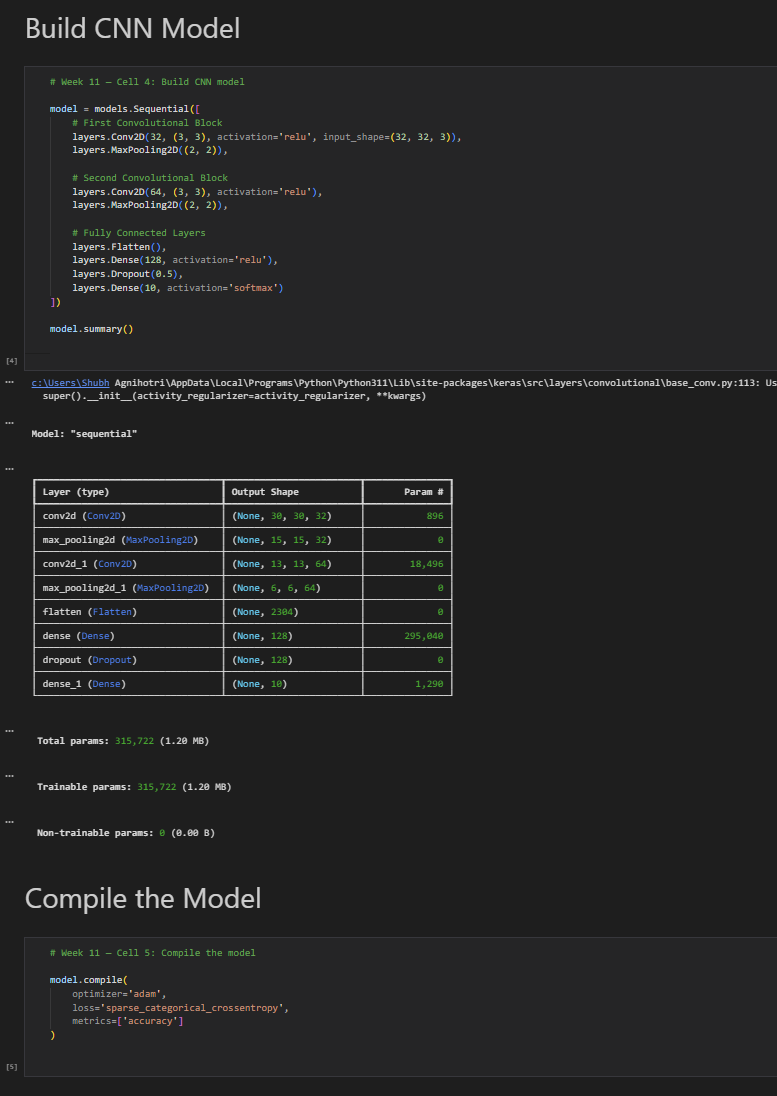


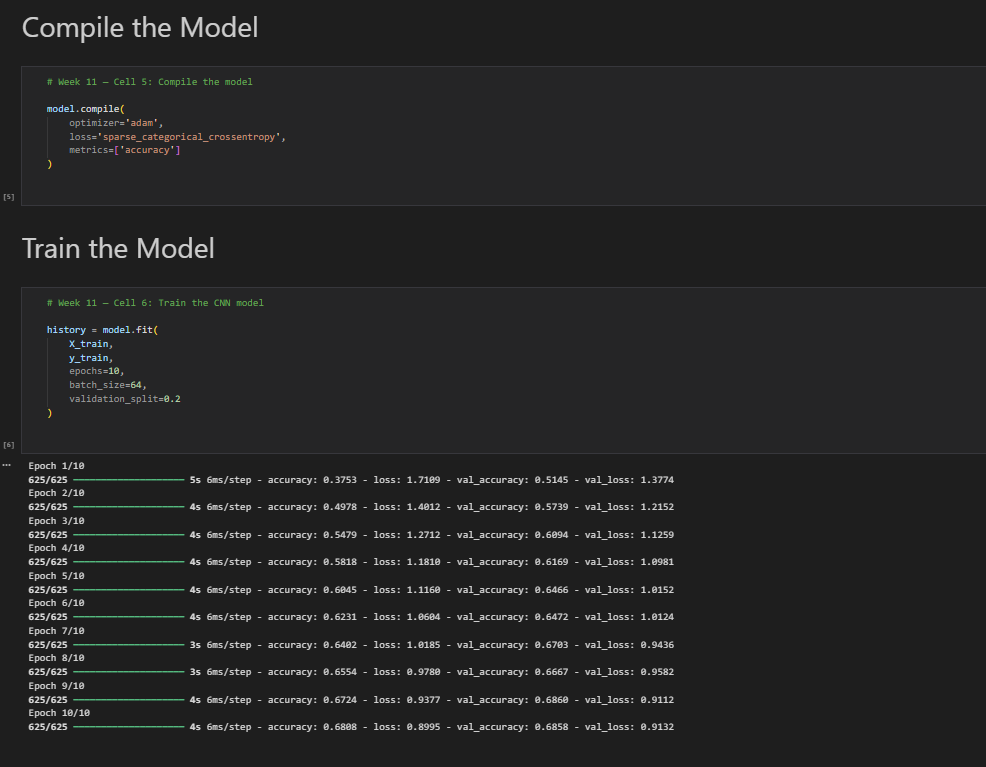


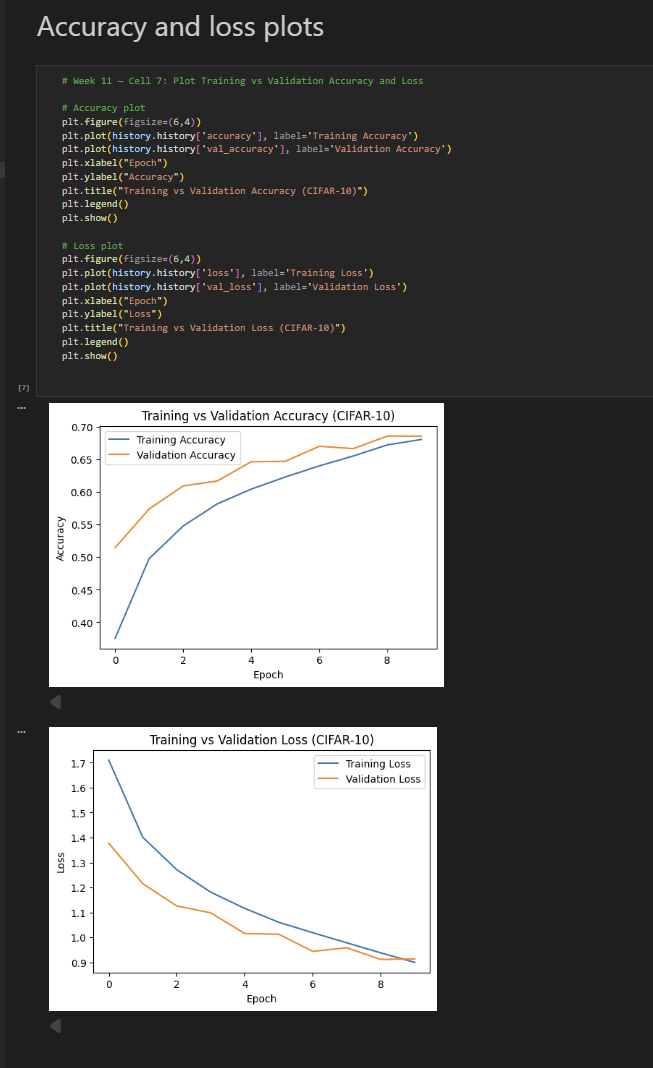
Week 11: Convolutional Neural Networks (CNN)

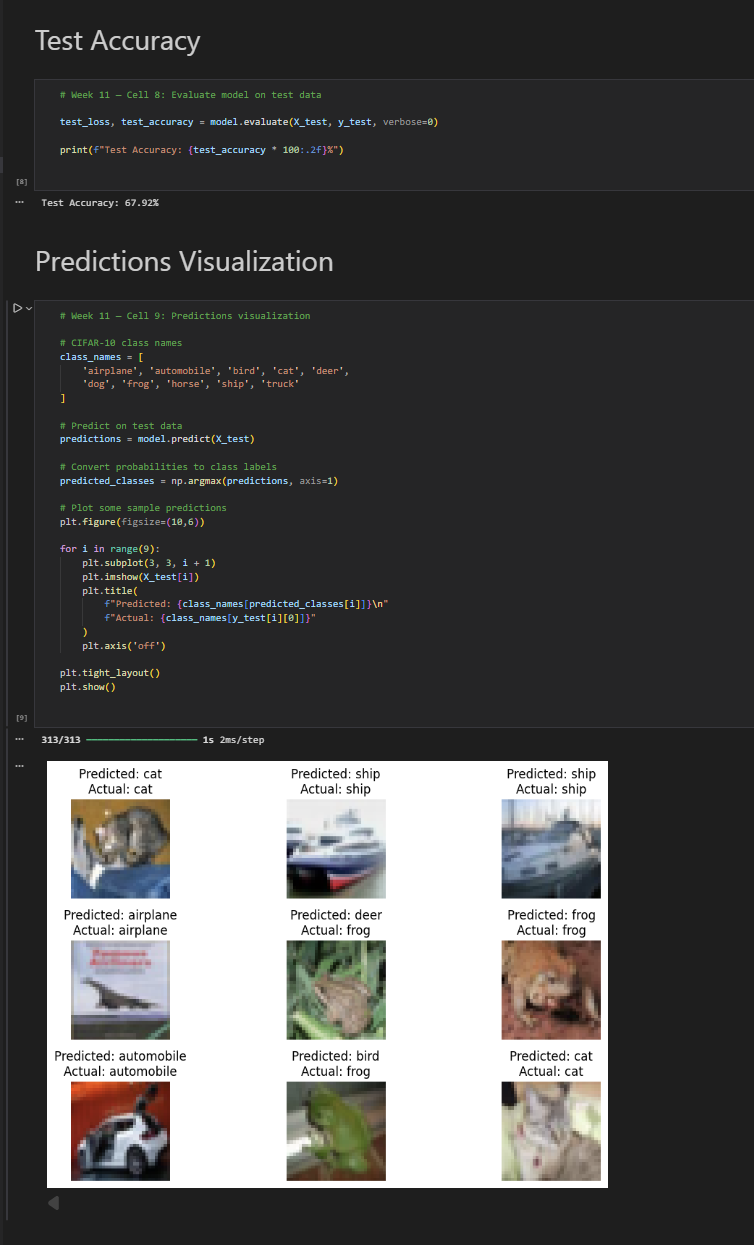
Objective: Learn image classification using CNNs











**Capstone Application: MNIST Digit Recognition Web App**

**Objective:**  
To develop and deploy a machine learning–based web application that classifies handwritten digits using a trained neural network model.

Description:  
In this project, a neural network model was trained on the MNIST dataset using TensorFlow/Keras. The trained model was saved and integrated into a Flask-based backend server. A web-based frontend was developed using HTML, CSS, and JavaScript, allowing users to upload handwritten digit images through a browser. The backend processes the image, performs prediction using the trained model, and returns the result to the frontend.

## Technologies Used:

* Python
* TensorFlow / Keras
* Flask
* HTML, CSS, JavaScript
* NumPy, Pillow

# Working:

1. User uploads an image through the web interface.
2. The image is sent to the Flask backend using the Fetch API.
3. The backend preprocesses the image and feeds it to the trained model.
4. The model predicts the digit and returns the result as JSON.
5. The prediction is displayed on the browser.

Outcome:  
The application successfully classifies handwritten digits and demonstrates the integration of machine learning models with web technologies.

