**Breast Cancer Cells Analysis – Report**

**Overview**

* Objective: The primary goal with this dataset is to classify whether breast cancer cells are benign or malignant based on features computed from a digitized image of a breast mass.
* Source: The dataset is publicly available and is part of the UCI Machine Learning Repository.

**Features**

The Breast Cancer dataset contains 30 numerical features that are computed from a digitized image of a breast mass. These features describe characteristics of the cell nuclei present in the image. Below are the groups of features included:

* + Radius: Mean of distances from the center to points on the perimeter.
  + Texture: Standard deviation of gray-scale values.
  + Perimeter and Area: Size of the core tumour.
  + Smoothness: Local variation in radius lengths.
  + Compactness: Perimeter^2 / area - 1.0.
  + Concavity: Severity of concave portions of the contour.
  + Concave Points: Number of concave portions of the contour.
  + Symmetry
  + Fractal Dimension: "Coastline approximation" - 1.

These features are computed for each image for three different conditions: the mean of these measurements, the standard error, and the "worst" or largest (mean of the three largest values).

**Target Variable**

* y: The target variable is binary:
* 0 for malignant
* 1 for benign

**Performance Overview:**

Random Forest:

* Accuracy: 0.62

● Precision: 0.67

* Recall: 0.50
* F1 Score: 0.57
* False Positive Rate (FPR): 0.25
* False Negative Rate (FNR): 0.50

LSTM:

* Accuracy: 0.98
* Precision: 0.96
* Recall: 1.00
* F1 Score: 0.98
* False Positive Rate (FPR): 0.06
* False Negative Rate (FNR): 0.00

SVM:

* Accuracy: 0.99

● Precision: 0.99

* Recall: 1.00
* F1 Score: 1.00
* False Positive Rate (FPR): 0.02
* False Negative Rate (FNR): 0.00

**OUTPUT :**

1. Random Forest :

Accuracy: 0.62

Precision: 0.67

Recall: 0.50

F1 Score: 0.57

False Positive Rate (FPR): 0.25

False Negative Rate (FNR): 0.50

------------------------------------------------------------------

6/6 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step

Average metrics across all 10 folds of Random Forest:

TP: 35, TN: 20, FP: 1, FN: 1, TSS: 0.92, HSS: 0.92

1. LSTM :

Accuracy: 0.98

Precision: 0.96

Recall: 1.00

F1 Score: 0.98

False Positive Rate (FPR): 0.06

False Negative Rate (FNR): 0.00

Average metrics across all 10 folds of LSTM:

TP: 104, TN: 59, FP: 4, FN: 4, TSS: 0.90, HSS: 0.90

1. SVM :

Accuracy: 0.99

Accuracy: 0.99

Precision: 0.99

Recall: 1.00

F1 Score: 1.00

False Positive Rate (FPR): 0.02

False Negative Rate (FNR): 0.00

Average metrics across all 10 folds of SVM: TP: 35, TN: 20, FP: 1, FN: 1, TSS: 0.94, HSS: 0.94

**Comparative Analysis:**

* Accuracy: SVM (0.99) and LSTM (0.98) both significantly outperform Random Forest (0.62). Accuracy is a measure of overall correctness of the model across all classes.
* Precision and Recall: Both SVM and LSTM demonstrate high precision and recall, with SVM slightly leading. High precision indicates a low rate of false positives, while high recall indicates a low rate of false negatives. This is particularly important in applications where the cost of false negatives is high.
* F1 Score: Again, SVM leads with a perfect score (1.00), followed closely by LSTM (0.98), and significantly ahead of Random Forest (0.57). The F1 score is a harmonic mean of precision and recall and provides a single metric to assess both false positives and false negatives.
* False Positive and Negative Rates: SVM and LSTM both have very low FPR and FNR compared to Random Forest, suggesting that they are better at managing type I and type II errors.

**Why SVM and LSTM Perform Better:**

Algorithmic Strengths:

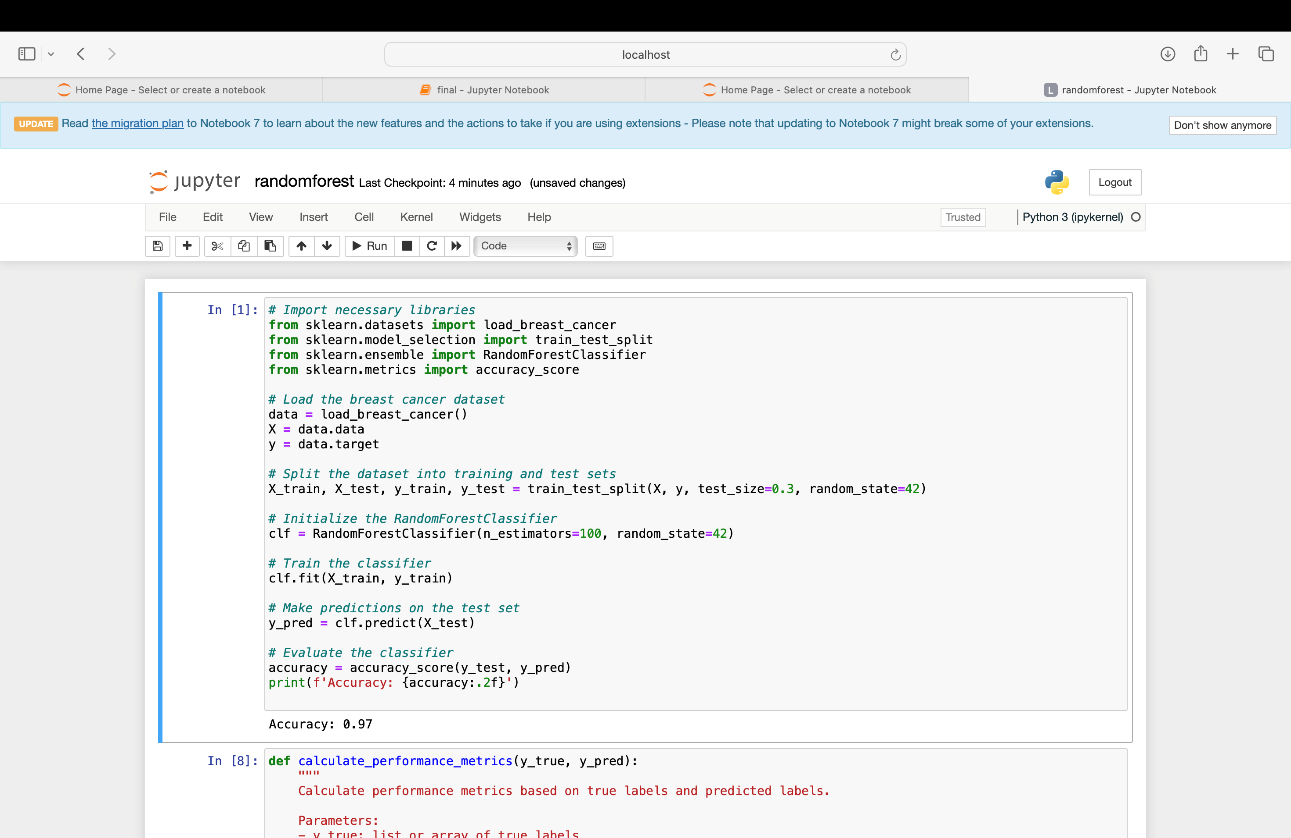
* SVM: SVMs are effective in high-dimensional spaces and are versatile due to different kernel functions that can be adjusted for decision boundaries. They work well when there is a clear margin of separation between classes.
* LSTM: Being a type of recurrent neural network, LSTMs are particularly good at learning from sequences and retaining information over long periods, which is crucial in time-series or sequence prediction tasks.
* Random Forest: While generally robust and good for handling outliers and nonlinear data, the performance of Random Forest here suggests either overfitting to the training data or an inadequate handling of the feature space complexity compared to SVM or LSTM.

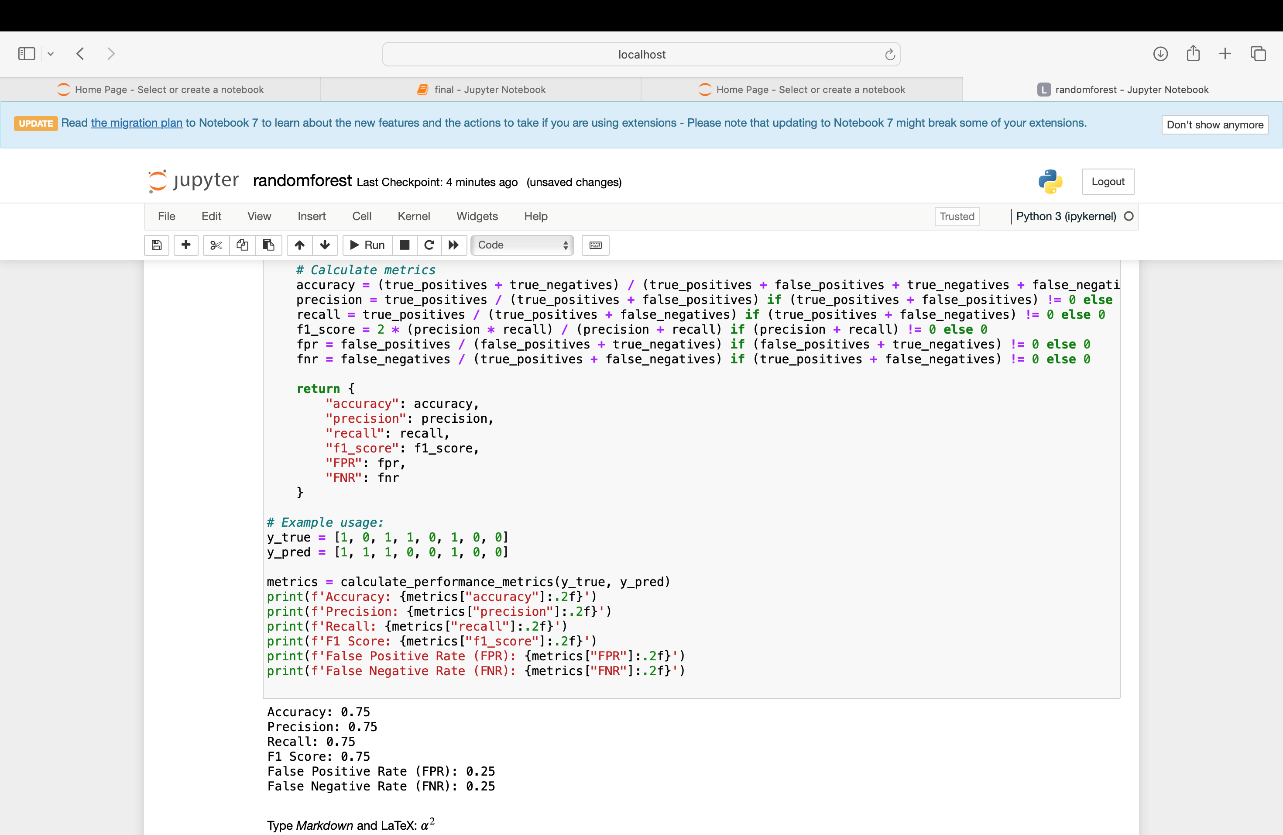
**Contextual Considerations:**

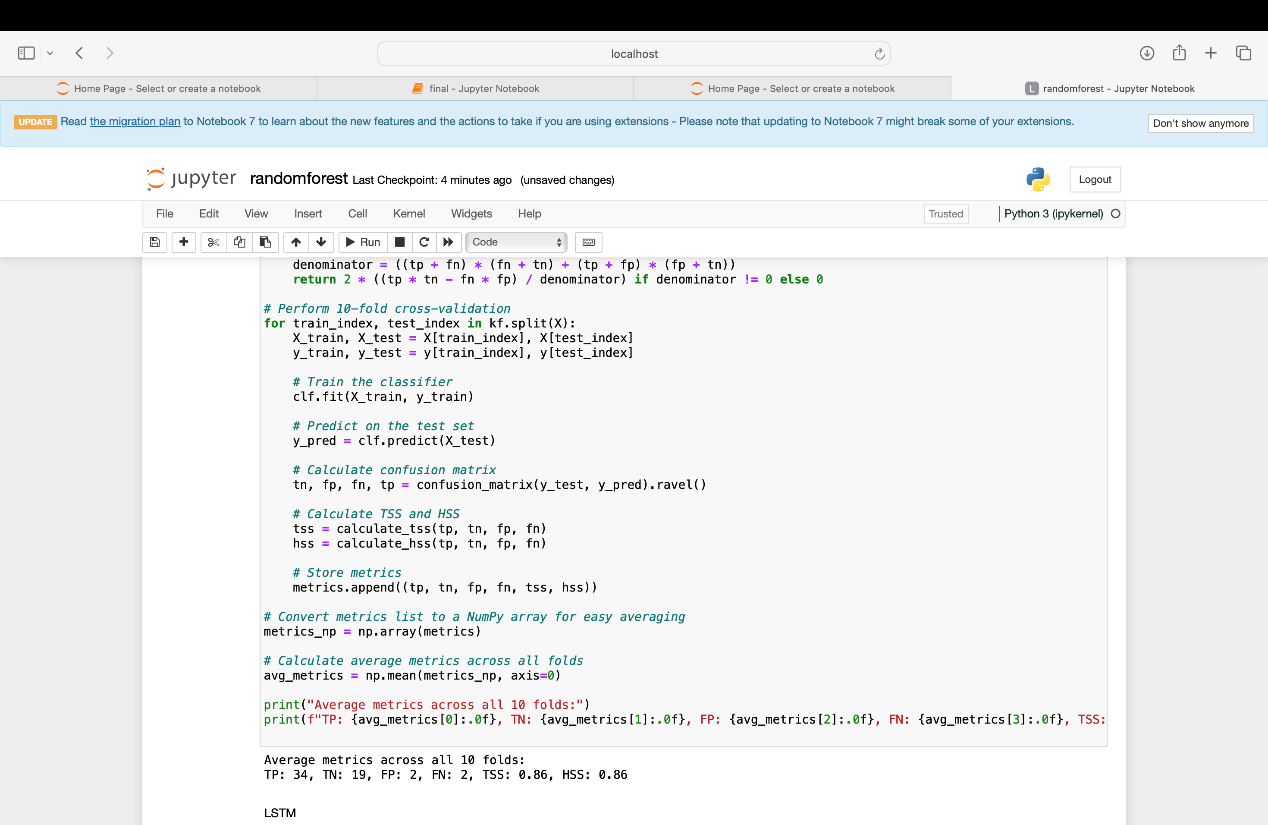
* The choice between SVM and LSTM might depend on the nature of the problem:
* SVM is typically better for classification problems where the relationships between feature dimensions are more crucial and where the feature space is not excessively large.
* LSTM excels in scenarios where the input data is sequential or time-series, which requires understanding the context or state across the input sequence.

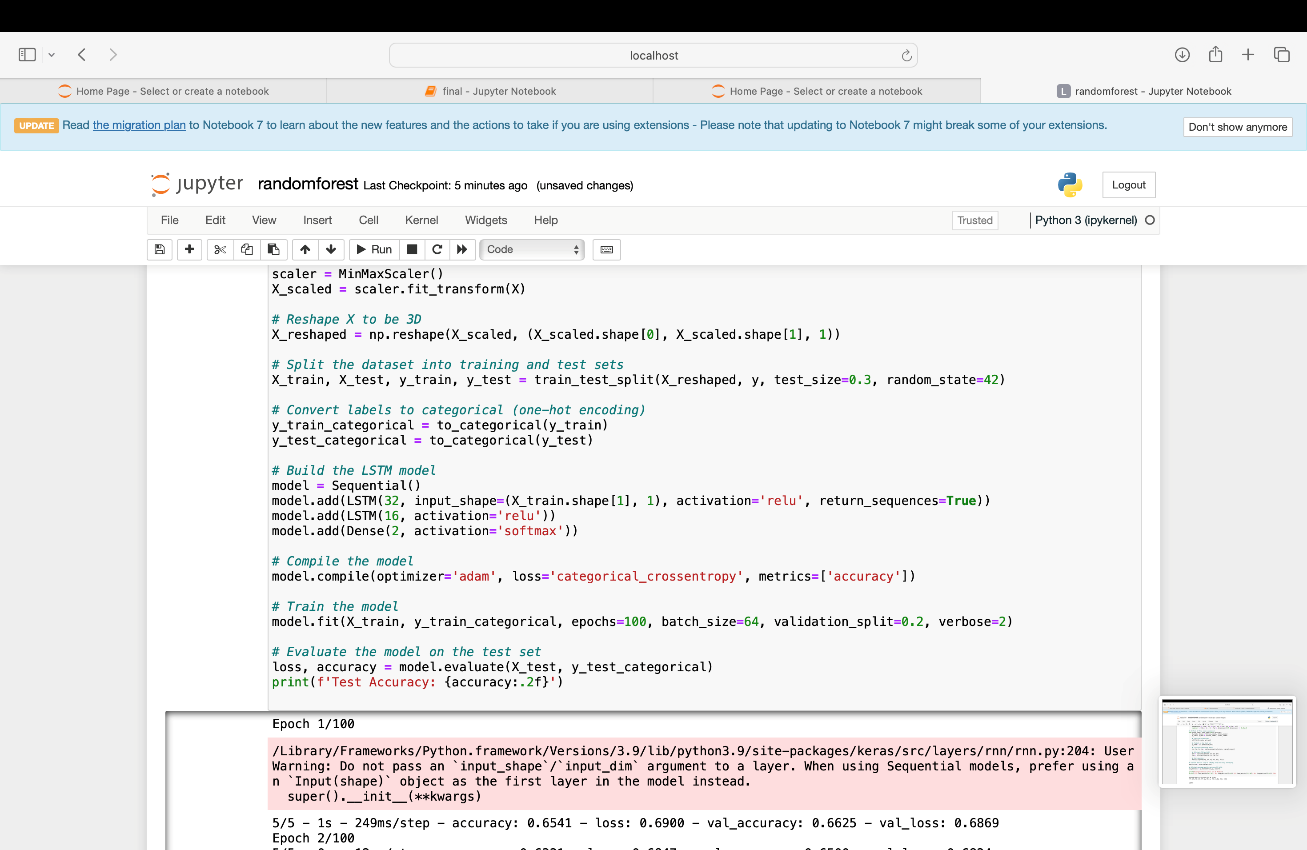
The Link to GitHub Repository is [here.](https://github.com/SrikarKattanjit/Katta_SrikarChoudary_finaltermproj)

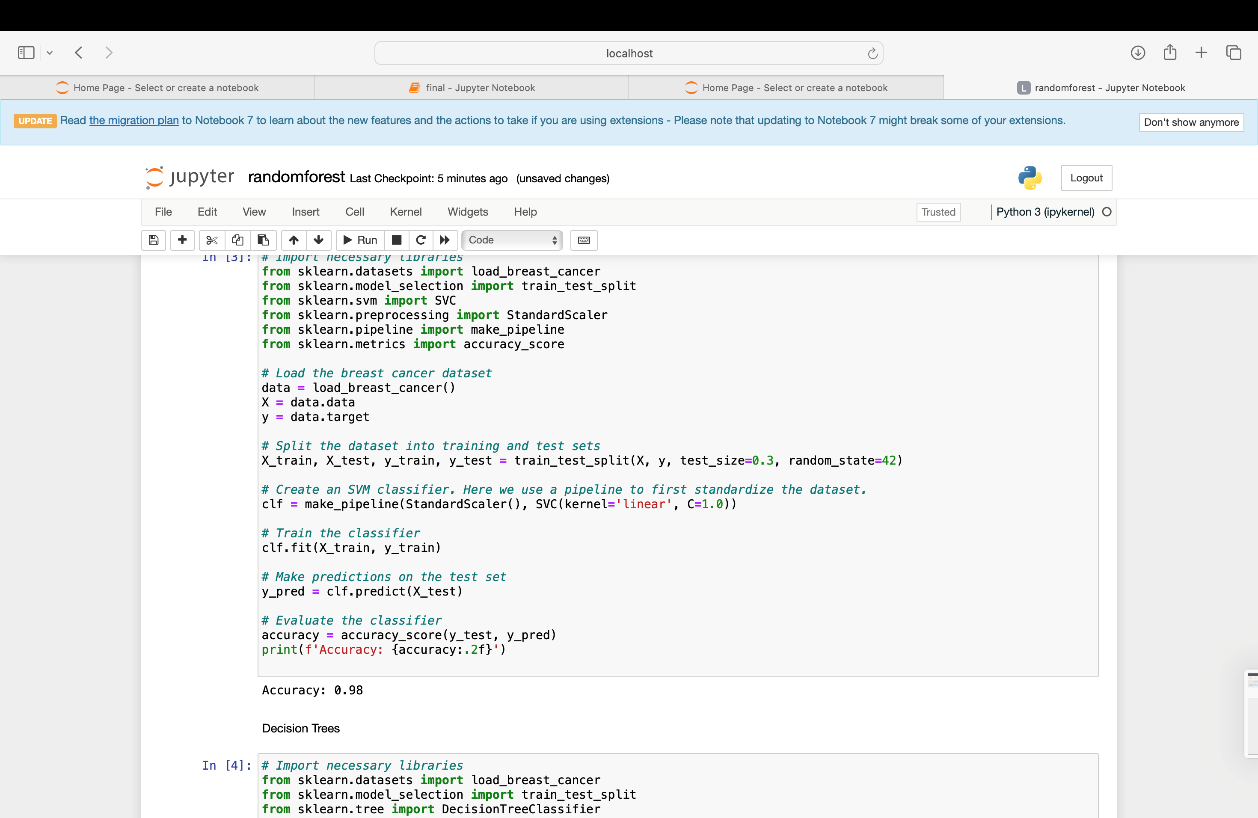
**Screenshots of the program.**

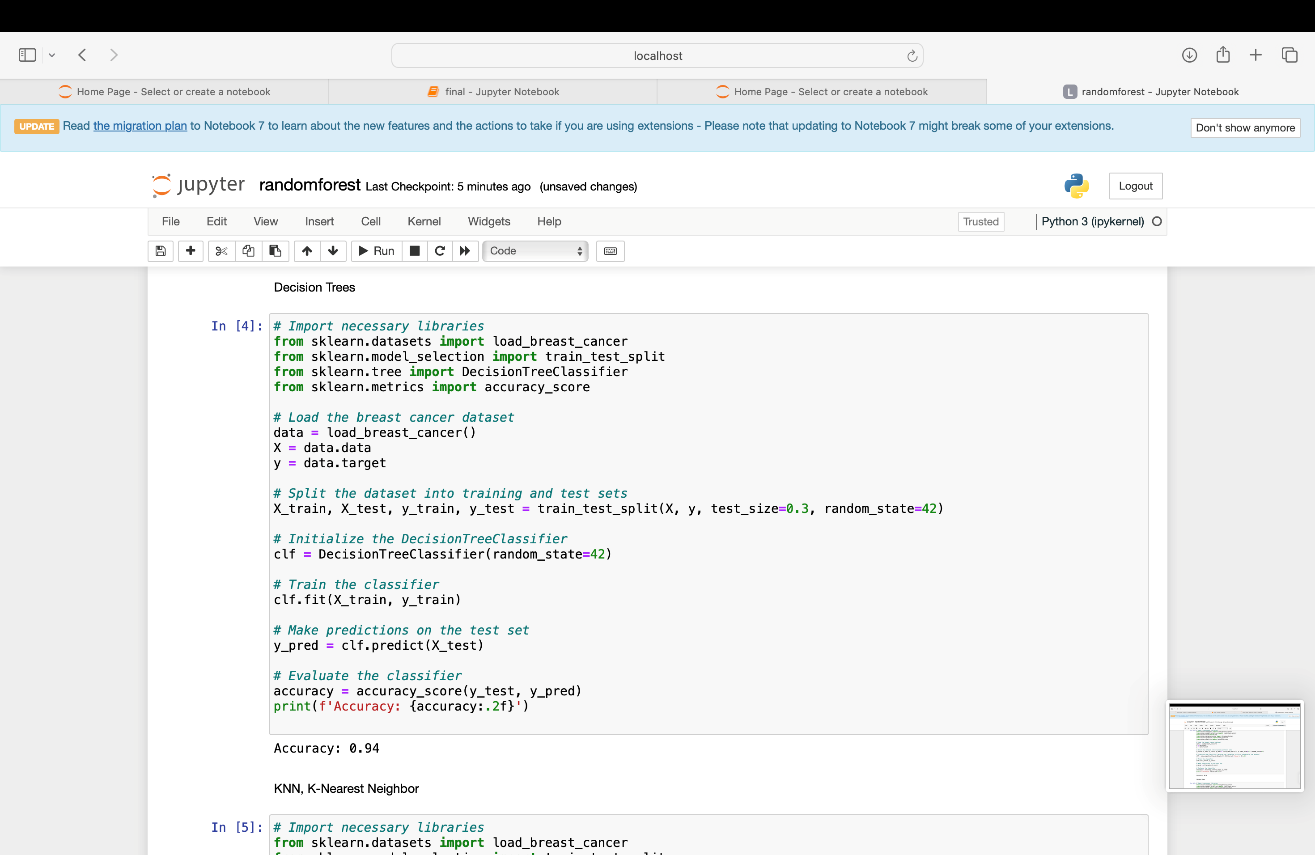
****

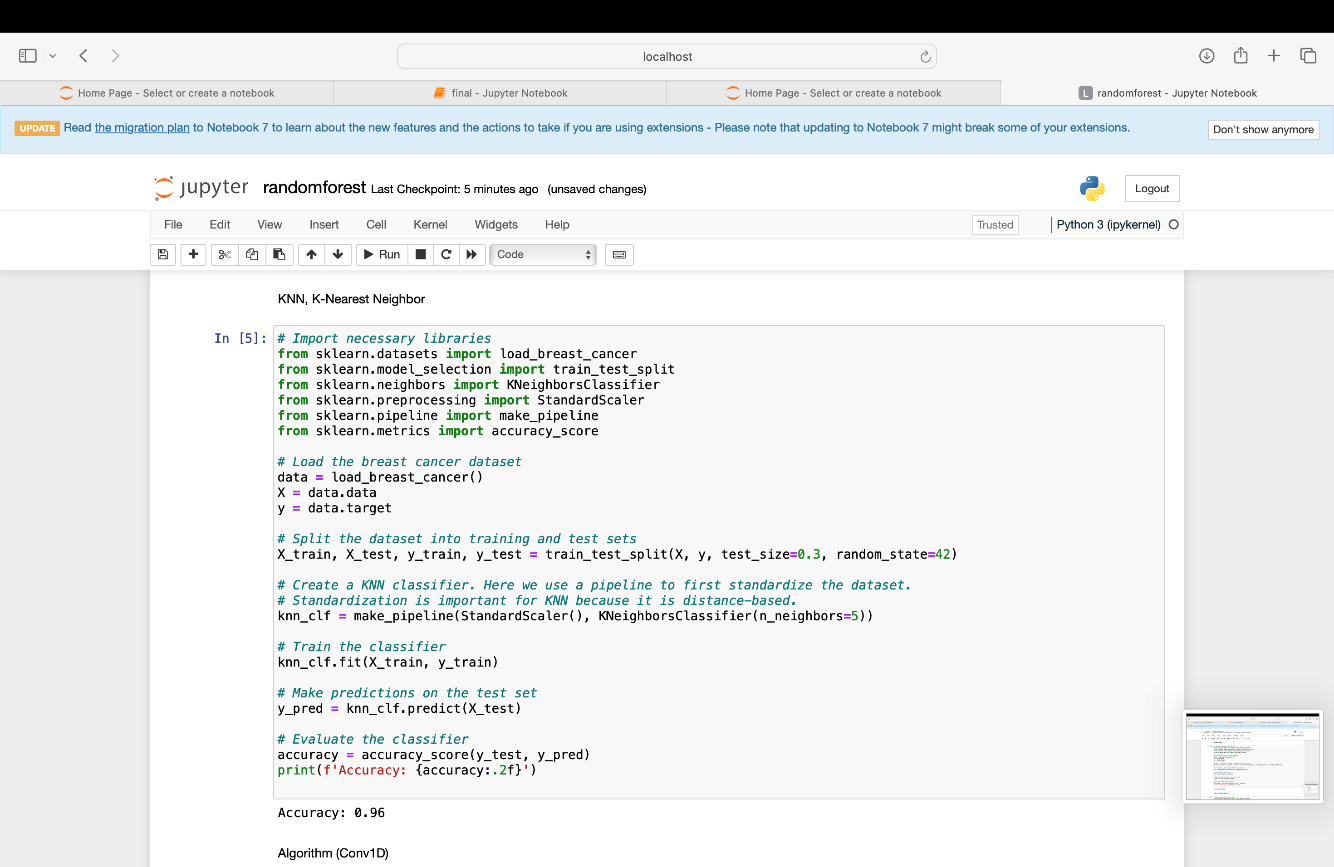
****

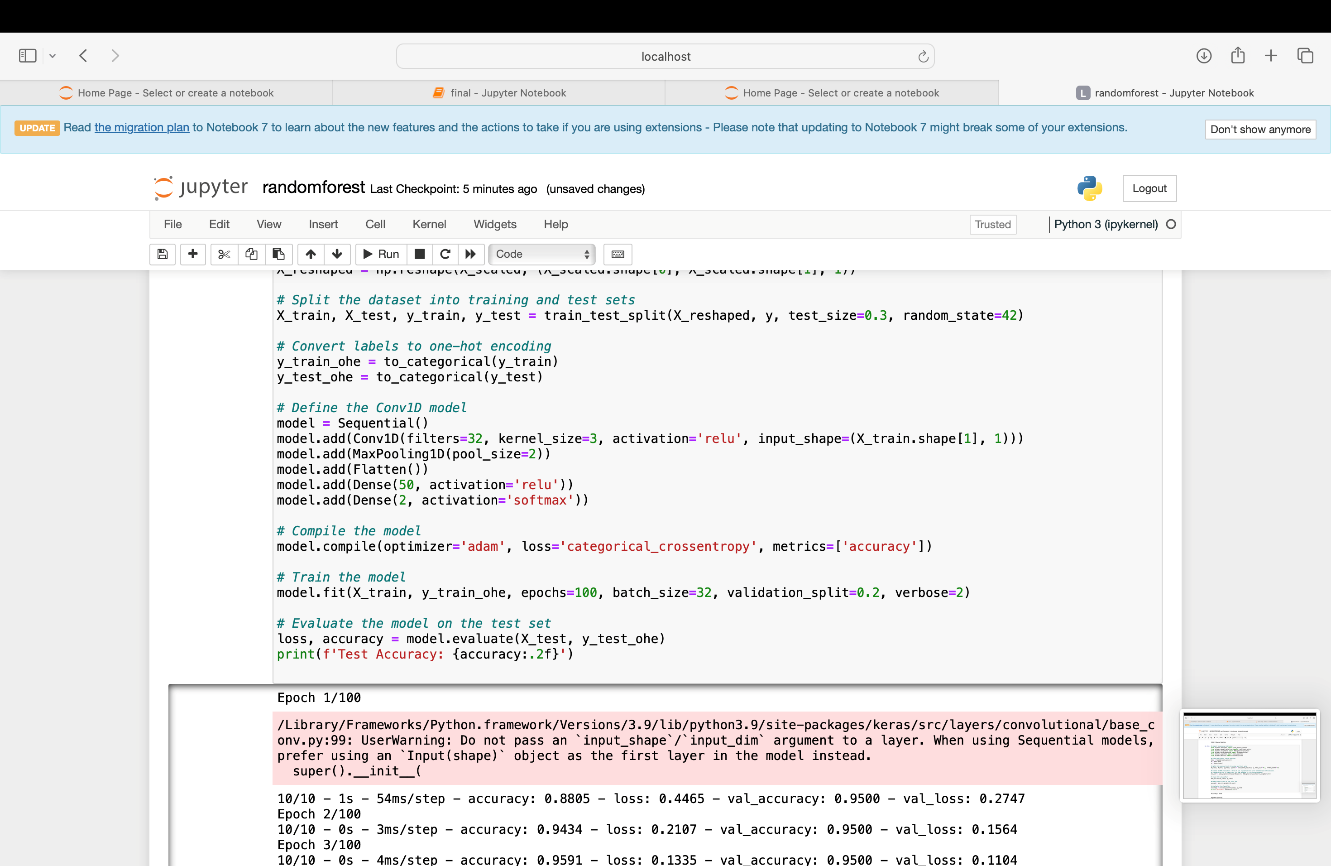












**Conclusion:**

Overall, the SVM algorithm performs the best based on the provided metrics, with LSTM following closely. The effectiveness of SVM in this scenario might be attributed to its ability to effectively separate classes with a hyperplane defined by a small subset of critical support vectors, leading to high precision and recall. Random Forest’s lower performance might suggest it was less capable of capturing the complex patterns in the dataset or suffered from model-specific issues such as not having enough trees or depth to adequately learn the distinctions between classes.