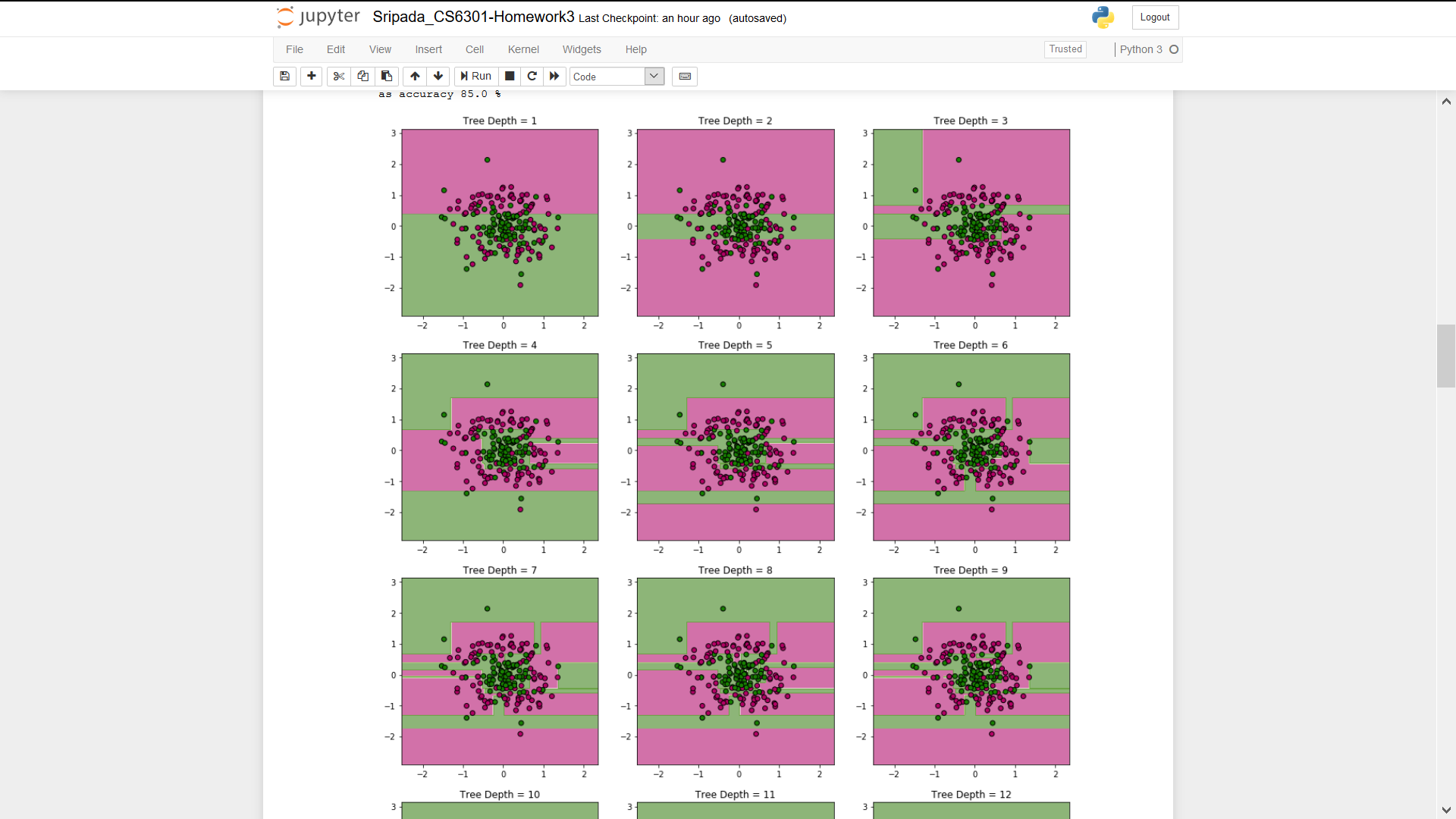
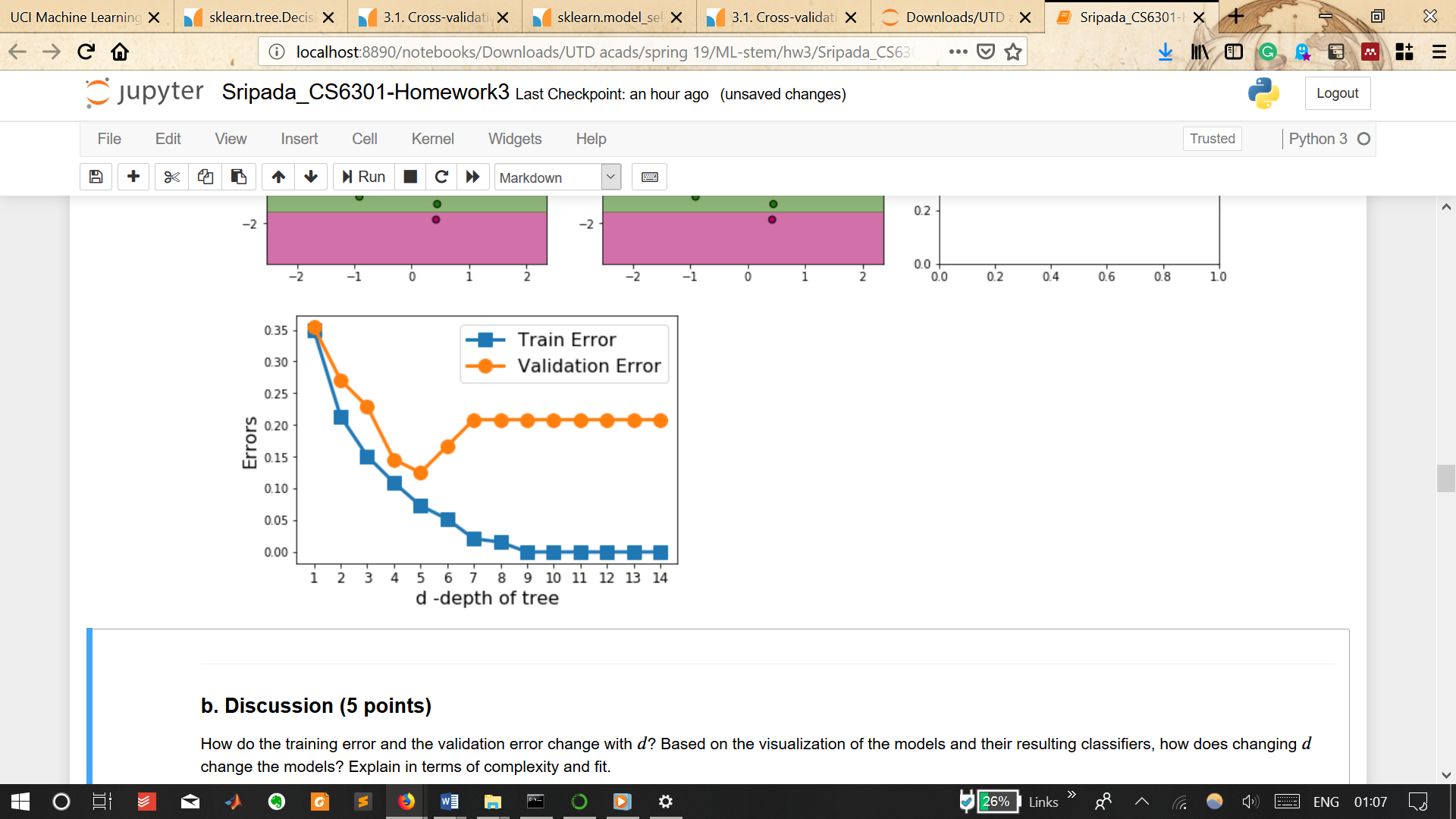
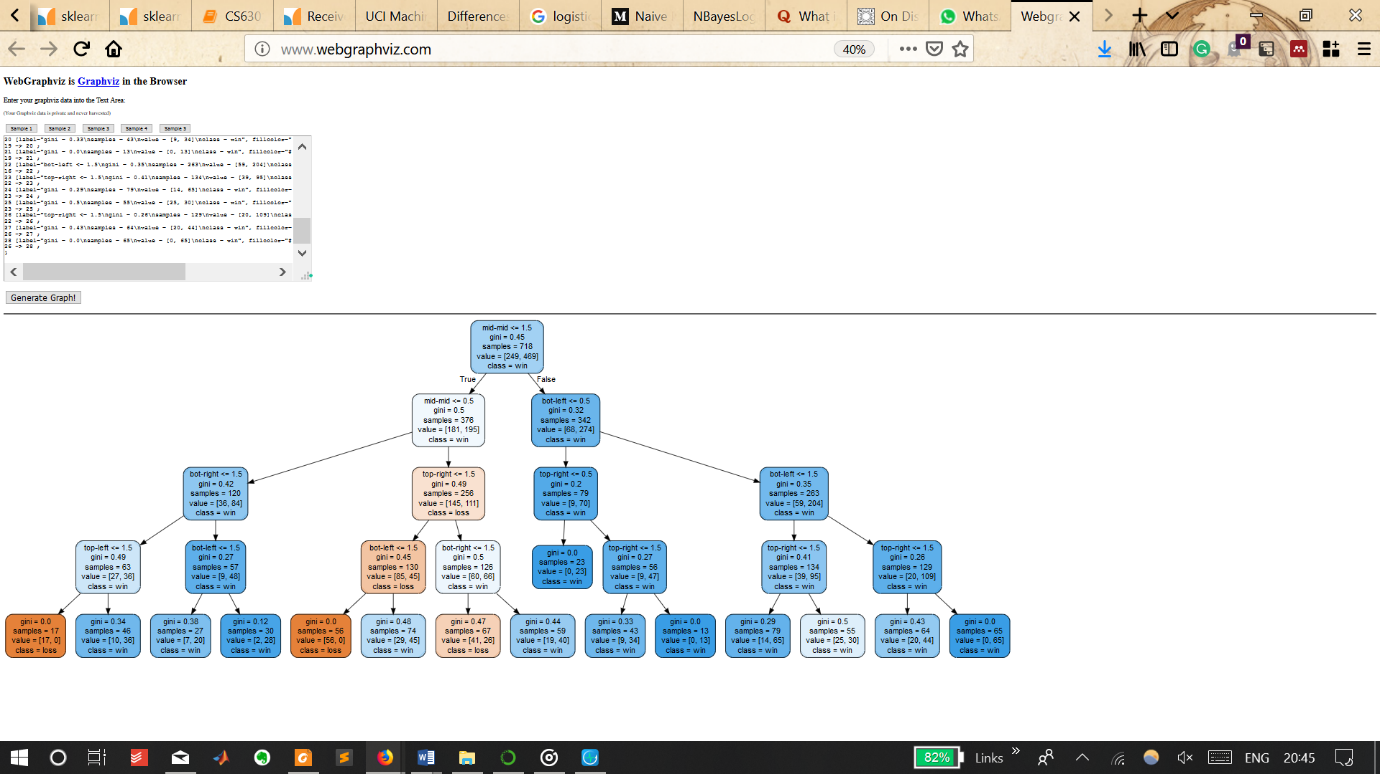
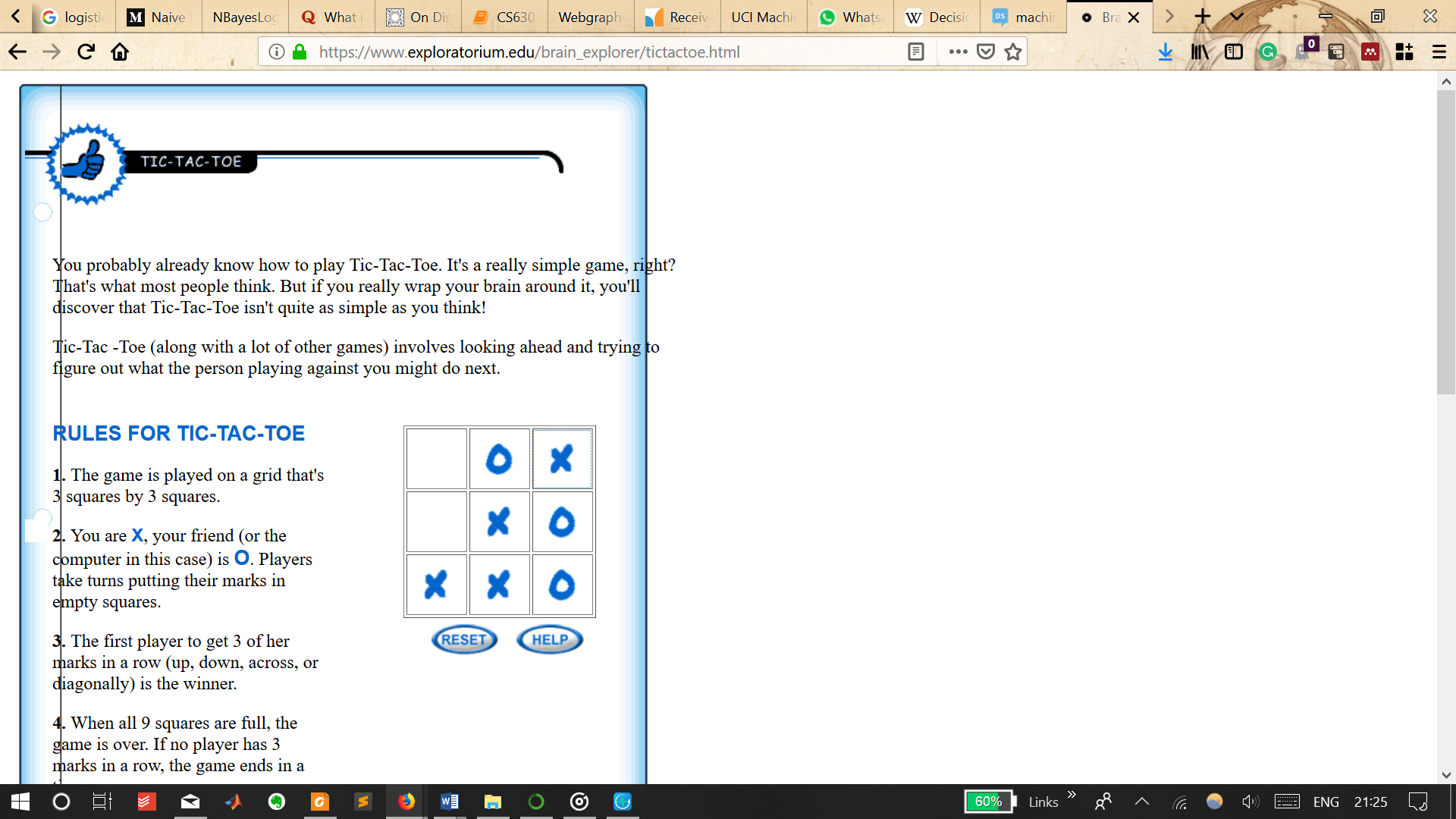
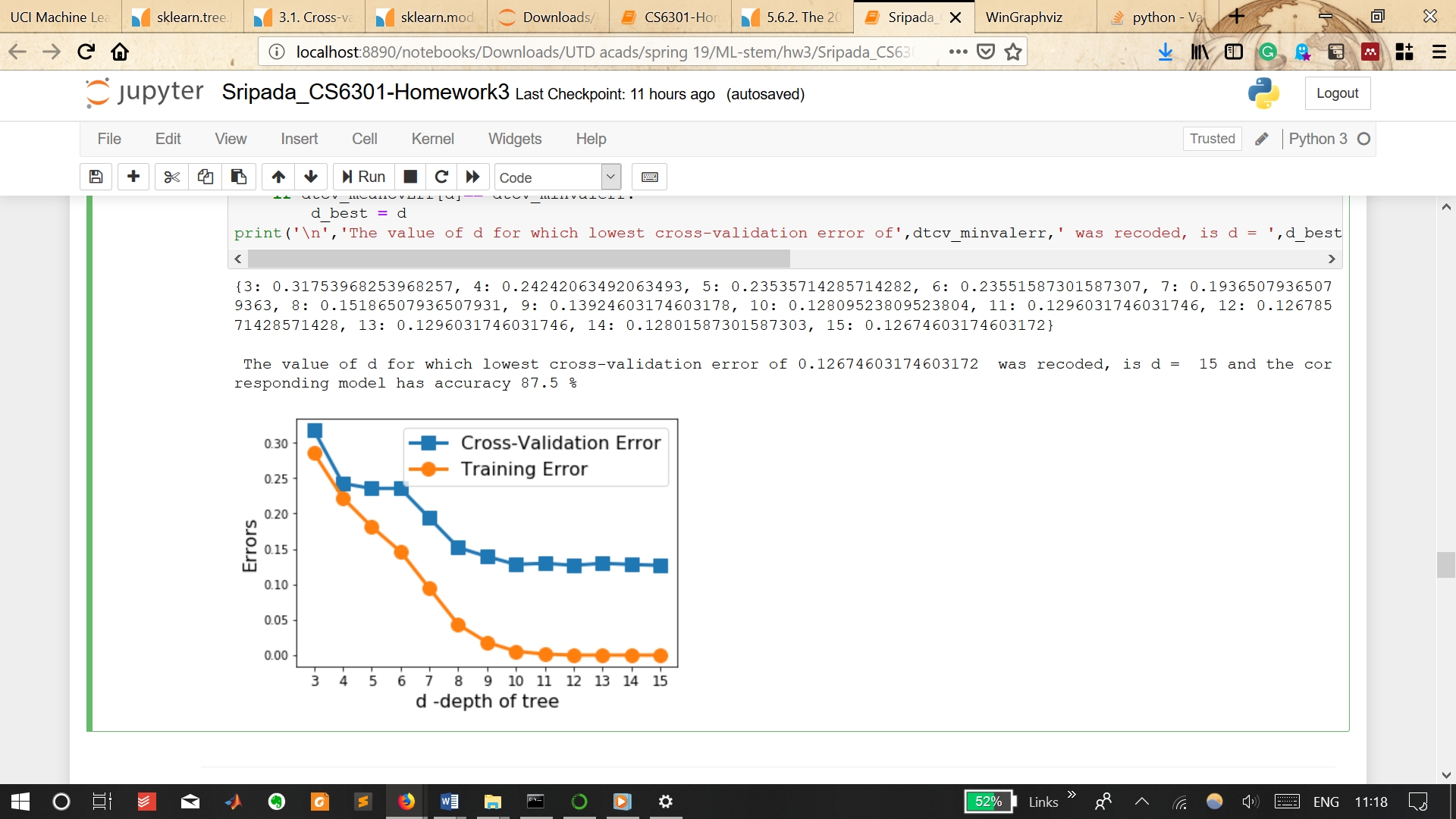
**CS6301.010 : HW 3 : DT-CV,NB,LR :- Siva Saket Sripada (2021432772)**



1. Training error decreases with increasing depth and saturates beyond d=10 (error is 0), which is expected since deeper trees are more complex and overfit data by minimising error and variance.
2. Validation error, however, decreases till d=5 but increases again for deeper trees and eventually saturates.
3. Based on classification visualization of the models it seems that trees with d<4 underfit data and are minimising complexity (making them high bias models).
4. While models with d>6 overfit data by binning outliers into a separate class thereby minimising variance ignoring complexity optimisation.

**Q1**

Thus, model visualisation as well as model selection based on validation error suggests d=5 seems to be the best depth for the problem of classifying concentric circles, with an accuracy of 80%.



Q2-Tic-Tac-Toe

1. The **leaf node with highest number of samples and least gini (corresponding to least entropy and thus highest confidence)** in the tree, which also corresponds to a win (for 'x') is the one with gini=0 & samples=65.
2. To get to this his leaf node, we need a false evaluation from the previous branching node (top-right<=1.5 & gini=0.26) – which corresponds to top-right = 0 (empty) or 1 (‘o’) as false, i.e. top right = 2 (‘x’).
3. To get to this above node (in point #2 of the reverse-tree-trace from leaf to root), we need a false evaluation in prior branching node (bot-left<=1.5 & gini =0.35) - which corresponds to bottom-left !=0 or 1, i.e. bottom-right=2 (‘x’).
4. To get to above node in pt#3, we need a false evaluation in prior branching node (bot-left<=0.5 & gini=0.32) – which corresponds to bottom-left !=0, i.e. bottom-left is occupied (either 1 or 2).
5. To get to above node in pt#4, we need a false evaluation in prior branching node (mid-mid<=1.5 & gini=0.45) – which corresponds to mid-mid != 0 or 1, i.e. mid-mid=2 (‘x’).

So, the game-flow is desired to get ‘x’ to occupy the minor diagonal (bottom-left + mid-mid + top-right) – such as, for instance, shown in image above.

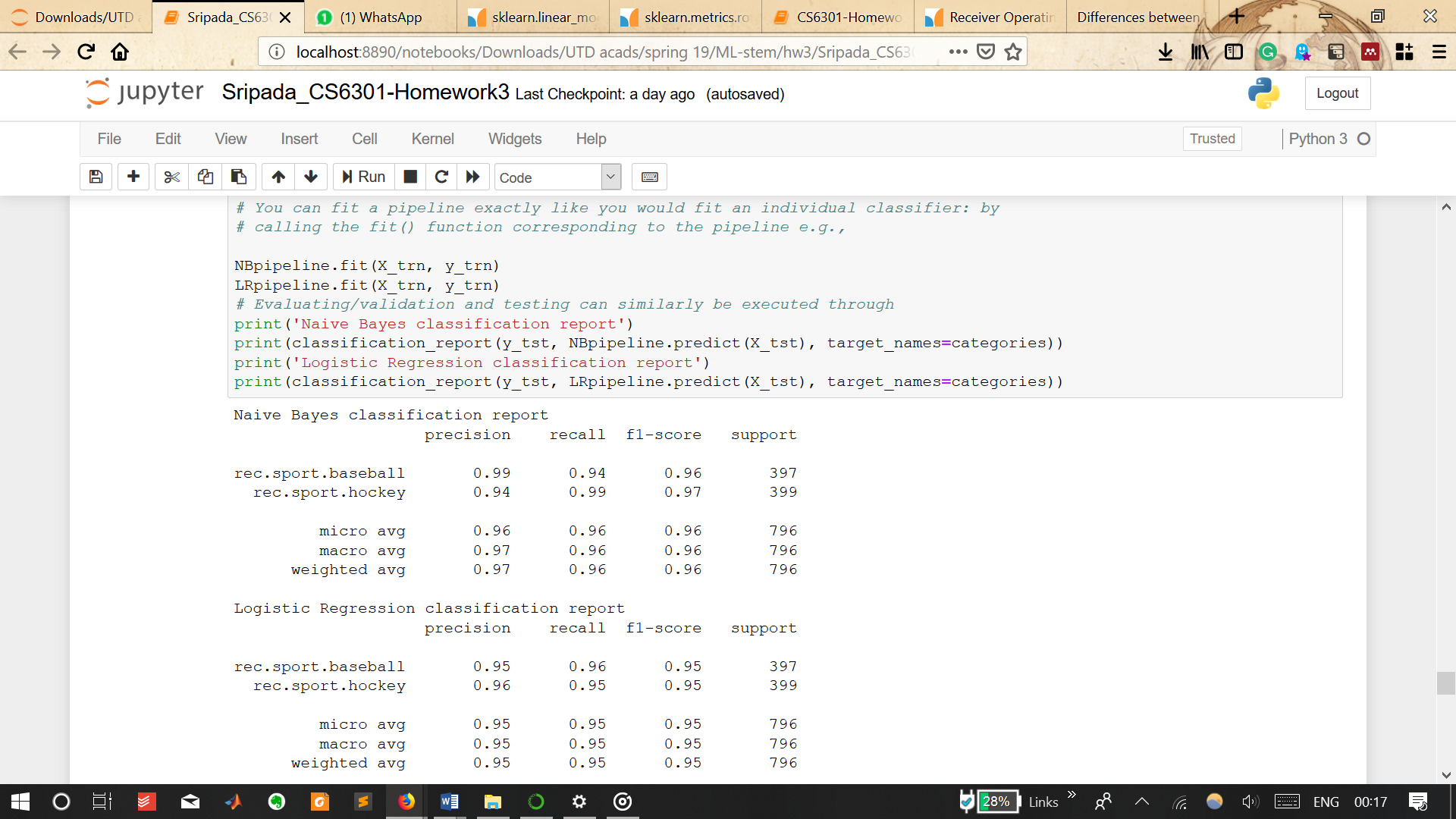
Round 1: ‘x’ in mid-mid & ‘o’ anywhere except top-right or bottom-left (else we violate the above sequence)

Round 2: This step is a seriously involved and explaining verbatim is avoided in interest of brevity.

* However, it might suffice, perhaps, to say that the placement of ‘x’ in round 2 is such that ‘o’ is not allowed to freely choose to occupy the minor diagonal - which in this sequence is desired to be filled by ‘x’.
* Another caveat is ensuring that two ‘o’s are adjacent, which would force ‘x’ to be placed elsewhere in an undesired manner.
* And so in this case of the image I attached, where ‘o’ is placed in one of the \*-mid boxes in round 1, ***‘x’ is placed in such a way as to force ‘o’ to occupy another \*-mid box.***

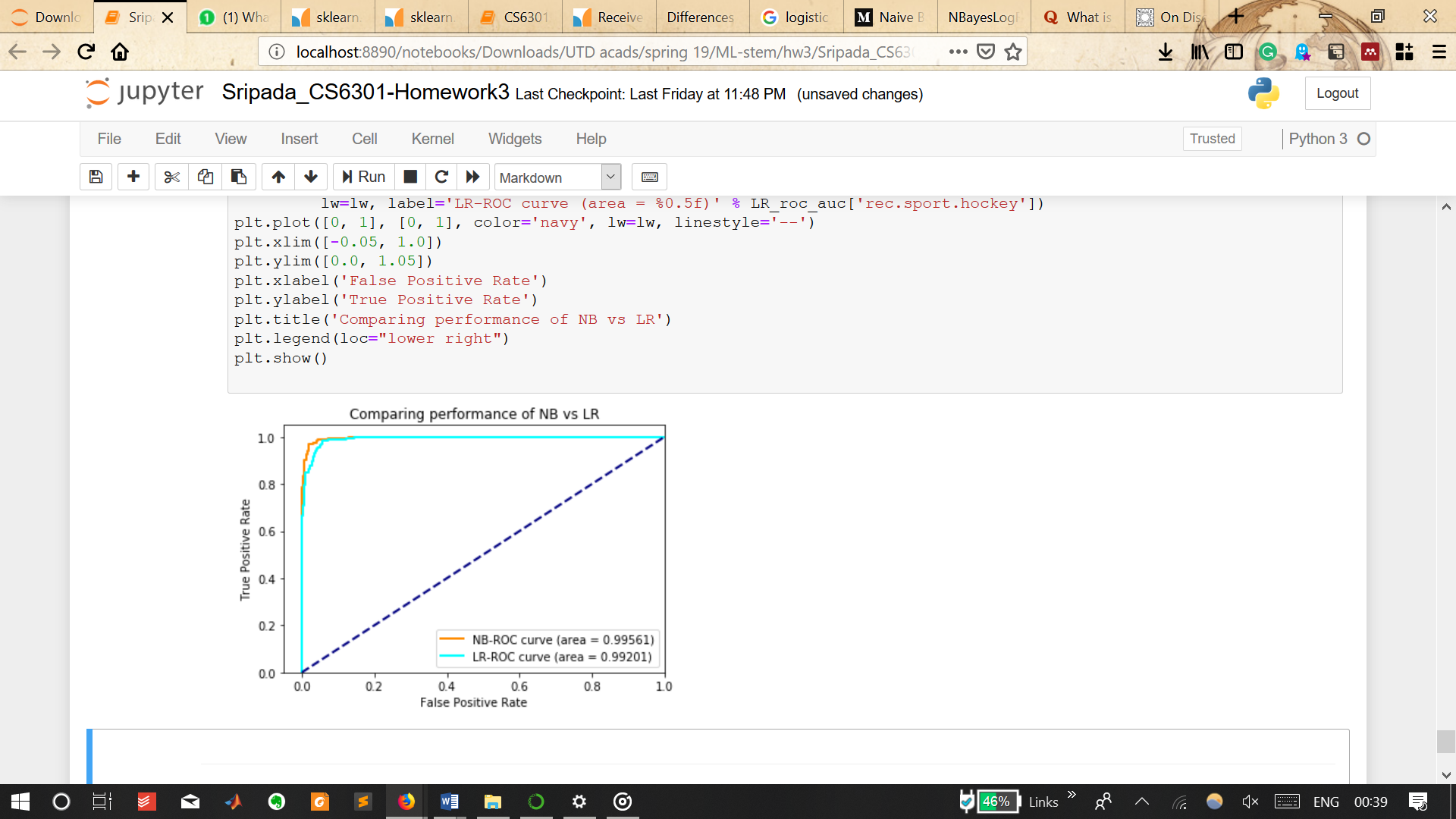
Round 3: In my game, in round 1 ‘o’ was placed in the mid-right box. Following the placement of ‘x’ as described above for round 2 (in the bottom-mid box in my picture, eliciting ‘o’ to be placed in top-mid box), ***in round 3, ‘x’ is placed in the bottom-left box, forcing ‘o’ to be placed in the bottom-right box****.* This is what is desired. End game by placing ‘x’ in top-right box.

**Q3 – Text classification using Naïve Bayes and Logistic regression pipelines**



Naïve Bayes is a generative algorithm in that it tries to conserve the underlying distribution of words and learns the joint probability distribution as is, albeit making a bold assumption of conditional independence among the different words.

For a bag-of-words representation like in this case, this assumption is not as bad, since the probability of occurrence of words in the respective categories is vastly different, and so prediction is not affected.

However, for very similar categories such as tennis and table-tennis or badminton, the assumption breaks the model’s predictions since the semantics and information on dependence of words (feature co-occurrence and similarity information) is key to finer classifications within the broad classes (such as sports).

Logistic Regression is a discriminatory algorithm in that it fits the examples(x) and labels(y) to a functional form and assumes the data to be linear separable as well as that each training example is identically and independently distributed (IID). This is just as bad as the Naïve Bayes assumption.

While the predictions of both algorithms are based on assumptions that ignore the semantics of sentences, the generative (and unassuming) nature of Naïve Bayes allows for better modelling of text and hence more accurate predictions.