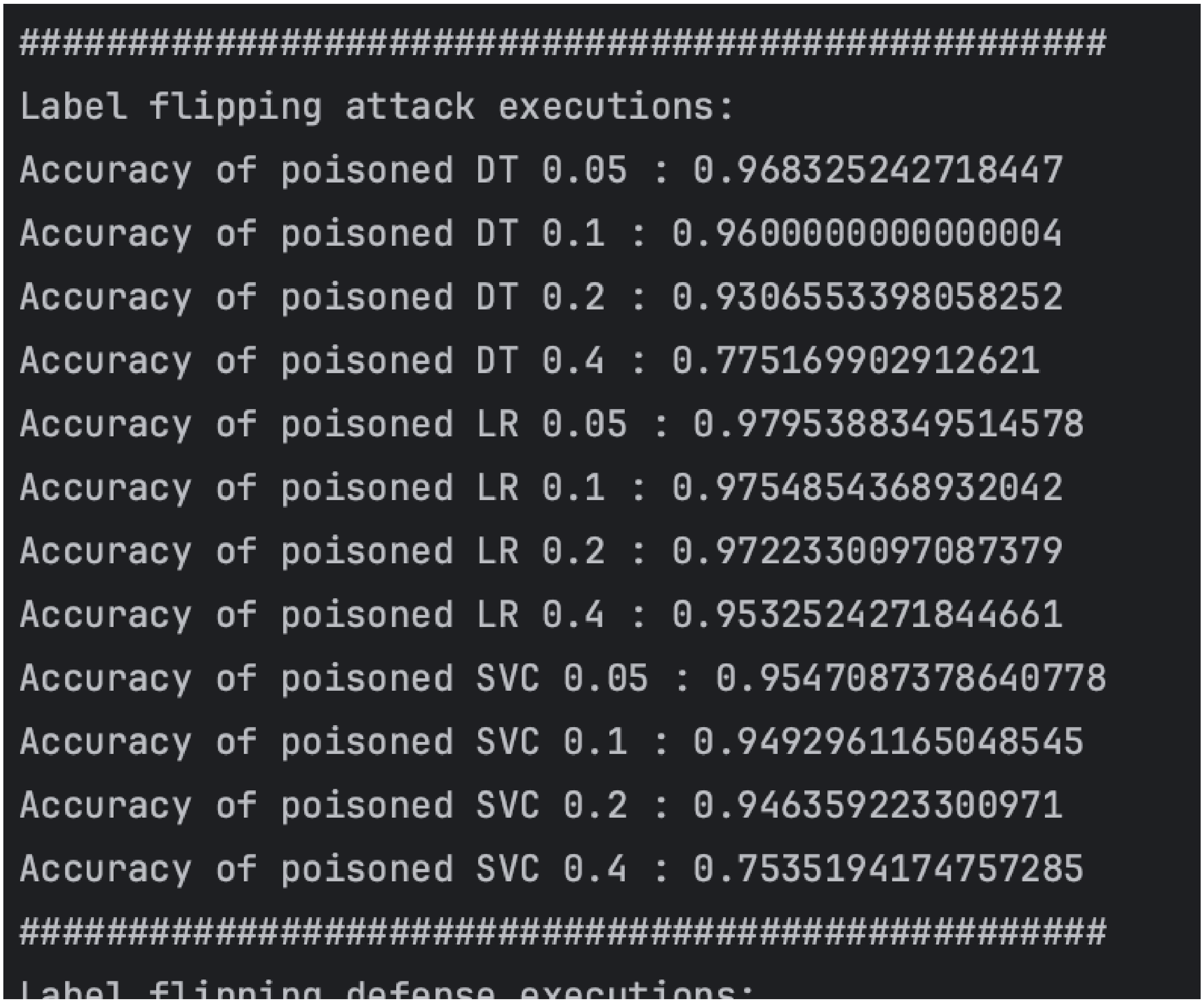
Comp430 HW3 Report

Batuhan Arat, 68665

Part 1:

Question 1: Label Flipping Attack

Results:



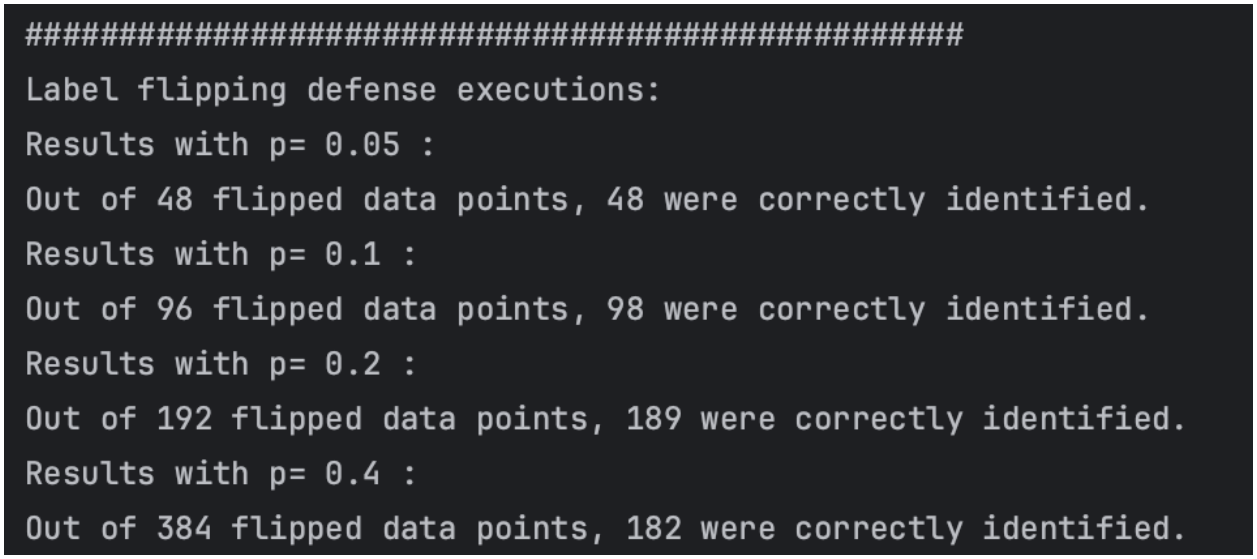
```
##################################################
Label flipping attack executions:
Accuracy of poisoned DT 0.05 : 0.968325242718447
Accuracy of poisoned DT 0.1 : 0.9600000000000004
Accuracy of poisoned DT 0.2 : 0.9306553398058252
Accuracy of poisoned DT 0.4 : 0.775169902912621
Accuracy of poisoned LR 0.05 : 0.9795388349514578
Accuracy of poisoned LR 0.1 : 0.9754854368932042
Accuracy of poisoned LR 0.2 : 0.9722330097087379
Accuracy of poisoned LR 0.4 : 0.9532524271844661
Accuracy of poisoned SVC 0.05 : 0.9547087378640778
Accuracy of poisoned SVC 0.1 : 0.9492961165048545
Accuracy of poisoned SVC 0.2 : 0.946359223300971
Accuracy of poisoned SVC 0.4 : 0.7535194174757285
##################################################
Label flipping defense executions:
```

As we can see while percentage of the p is increasing our accuracy is decreasing in all of the three models.This is quite reasonably because while our p percentage is increase we are basically flipping more data labels in the dataset, so the deviation from original data should decrease.

On the other hand LR, is not deviate like the other two. It is still decreasing but with a much lower fashion. So we can say LR is more robust to the flipping poison attack.

Question 2: Defense Against Label Flipping

Results:



```
##################################################
Label flipping defense executions:
Results with p= 0.05 :
Out of 48 flipped data points, 48 were correctly identified.
Results with p= 0.1 :
Out of 96 flipped data points, 98 were correctly identified.
Results with p= 0.2 :
Out of 192 flipped data points, 189 were correctly identified.
Results with p= 0.4 :
Out of 384 flipped data points, 182 were correctly identified.
```

I don't use Local Outlier Factor or Isolation Forest as it is advised on the pdf because it confuses me. What I did is basically applying the heuristics of "if the current data point is surrounded by samples of the opposite class, then it is an outlier, therefore probably it was flipped" by k-means clustering. I have discover that sklearn has a class of this so I import `from sklearn.neighbors import NearestNeighbors`

I have constructed groups of 15 neighbors and I choose ball_tree algorithm it is more efficient one in terms of big and high dimension data.

For all groups I have checked wheter this group is 0 or 1 oriented by calculating each of its frequency. According to their dominant label I also create a threshold (which is 8 in this case) to assert its dominance level. If they are in a group that is dominance level above the threshold and if they are in the opposite label I marked as them flipped.

Since it is discussed on the discussion board, you are not looking for false positives I don’t take into account of falsely misclassified flippeds.

So it allows me a shortcut to directly take the real corrected ones. So i added another and in my if check to see if it is on the real flippeds. If they are I took them as successful.

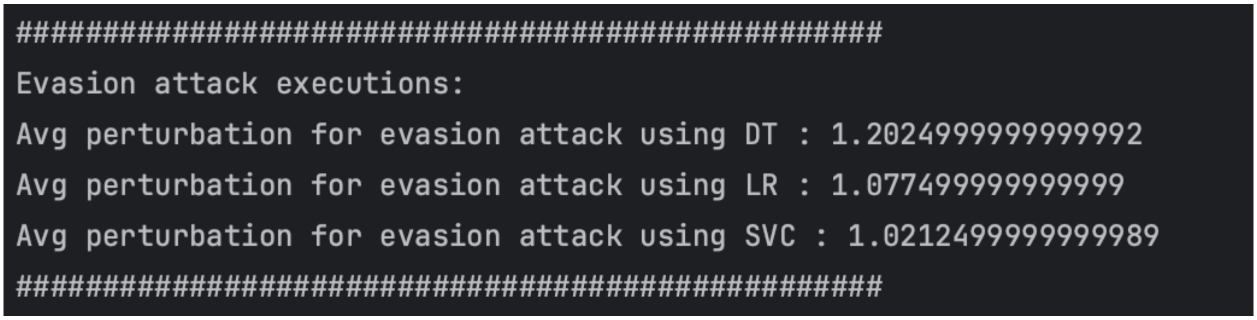
##for 0 case

```
if dominant_label == 0 and count_of_0 > 8 and y_train_flipped[indices[i][0]] == 1  and indices[i][0] in selected_indexes:
```

I think this is very effective defense, and results are supports it. This is very effective on small flipped counts but it gets less affective as p increases. Which is also reasonably because my defense mechanism based on detects the outliers from groups which are similar. Since p getting larger, randomness increase and pattern between similar data will decreasing. But in all cases it is still greater than > 40%. I choose parameters according to this.

Question 3: Evasion Attack

Results:



```
##################################################
Evasion attack executions:
Avg perturbation for evasion attack using DT : 1.2024999999999992
Avg perturbation for evasion attack using LR : 1.077499999999999
Avg perturbation for evasion attack using SVC : 1.0212499999999989
##################################################
```

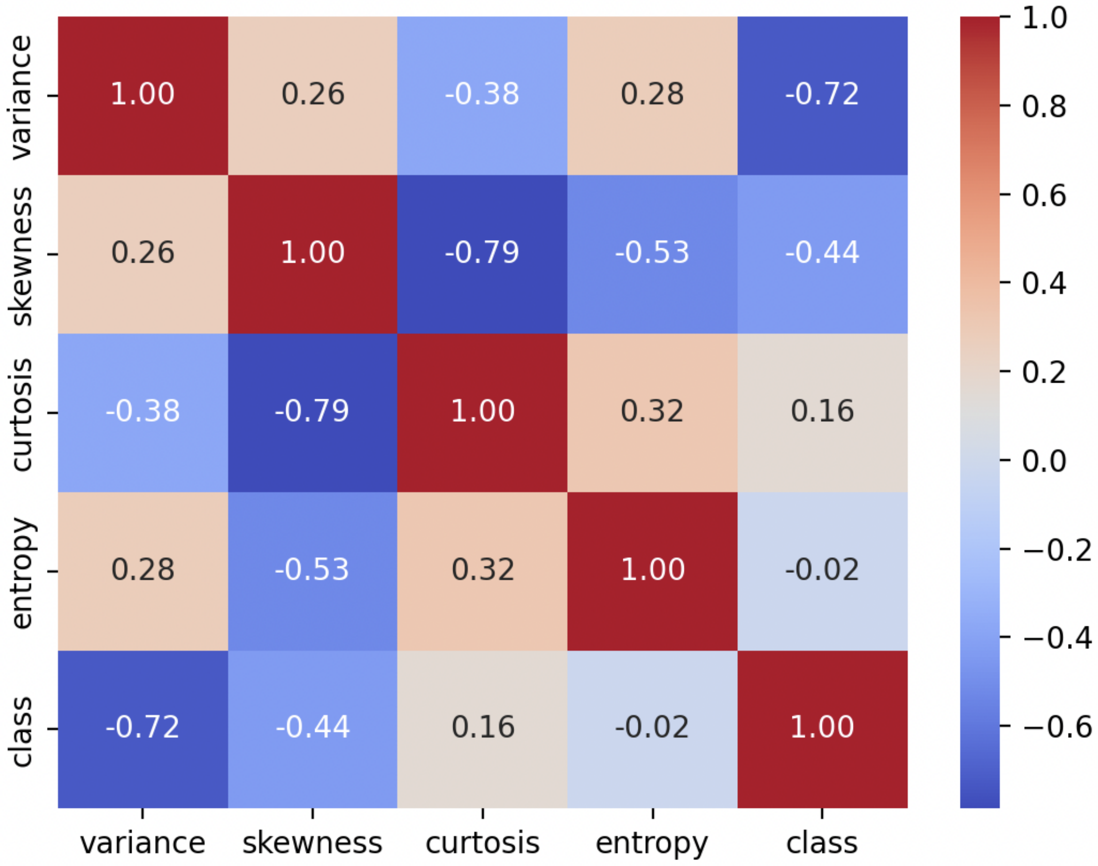


Figure1: *Heat map of the data frame*

Since randomly increasing any weight and break from the function if it is higher than the limit (3) is not efficient, I have to come up something more efficient.

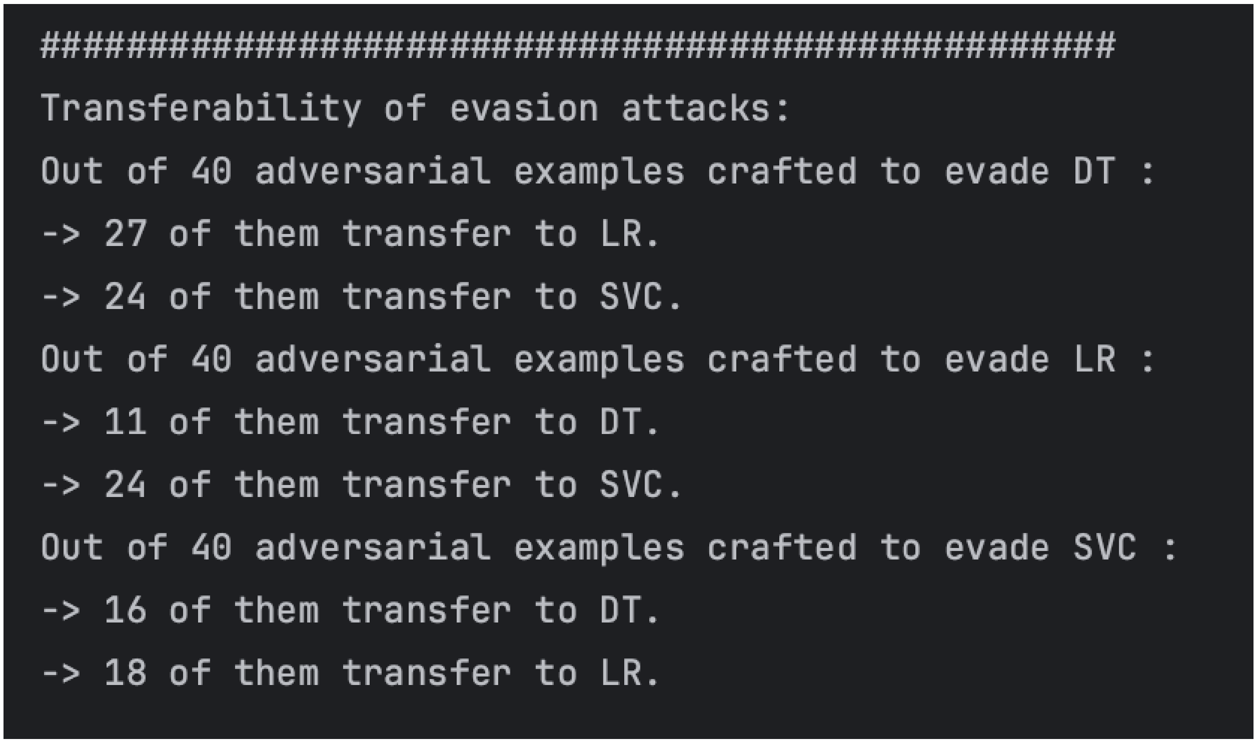
First I look at this data frame covariance matrix and heat map to correctly understand the nature of this data. At the last row we see the relationship between labels and features. Since we can see labels are highly correlated with variance and skewness, we can come up with an approach of only changing those in order to make efficient weight increases.

I figured that if variance and skewness were negative it is probably labeled as 1, and it is supported on the covariance matrix by showed as negatively correlated.

So if it is 1, to evade it we can make it 0 by adding weights on just the variance and skewness, and vice versa on the 0.

Question 4: Evasion Attack Transferability

Results:



```
##################################################
Transferability of evasion attacks:
Out of 40 adversarial examples crafted to evade DT :
-> 27 of them transfer to LR.
-> 24 of them transfer to SVC.
Out of 40 adversarial examples crafted to evade LR :
-> 11 of them transfer to DT.
-> 24 of them transfer to SVC.
Out of 40 adversarial examples crafted to evade SVC :
-> 16 of them transfer to DT.
-> 18 of them transfer to LR.
```

In this part we are comparing the prediction of the two algorithms and see if they are same or not. If the more point becomes same we can say those two model are more transferable. For example when we are looking to the results we can say that DT is more transferable to LR than SVC. LR more transferrable to SVC than DT. SVC is more transferrable to LR then DT

Part 2:

Question 3: Defense

Since on the hint it is said “*A defense mechanism based on detecting English or non-English words can be devised.”* I look up the nlp packages that I can use for detecting non-english words.

I found nltk is a wide approach.

So basically, by traversing the words at text at the data frame, I check if this is an english word, it is number, or it is punctions. If it is not, i remove from the sentence and construct all of the text like this.

About excell

What can we deduct from the file is, while trigger sentence length increase it is more accurately classified on the sentence level backdoor. Success rate is also getting increased. When we are comparing with the other ML models RF and LR are the best ones.

At the word level backdoor data we can see that my defense is not very powerful. I implemented in a way that it cleanse the trigger keys but it also cleanse the normal data. That affects the behavior.