Question 1

Model	P	Accuracy
DT	0.05	0.9675728155339803
DT	0.1	0.9593446601941751
DT	0.2	0.9299999999999997
DT	0.4	0.7801699029126213
LR	0.05	0.980242718446603
LR	0.1	0.9753883495145633
LR	0.2	0.9709466019417478
LR	0.4	0.9529611650485438
SVC	0.05	0.9542233009708739
SVC	0.1	0.9485194174757292
SVC	0.2	0.9456553398058254
SVC	0.4	0.7472572815533981

Table 1

As we can see from the table 1 accuracy of the DT and SVC models are decreasing drastically as we are increasing the percent of the labels we are flipping. However, even though, there an accuracy decrease occurred in SVC algorithm, the decrease is very small. In general sense, as the poison in the data increases, the accuracy of the model decreases. We can conclude that in terms of label flipping the most resilient model among DT, LR and SVC is LR.

Question 2

In order to design my defence algorithm, I firstly checked outlier detection algorithms of the scikit-learn library's which are Local Outlier Factor (LOF) and Isolation Forest (IF). For comparing them I implemented the same logic with both of the algorithms. When I executed and compared the results of the LOF and IF based algorithms the LOF shoed better performance, therefore, I chose the LOF.

For LOF I used two parameters n_neighbor and contamination. Contamination is a parameter for the proportion of the outliers in the data, therefore, it is equal to our p value. "n_neigbor" parameter is the parameter for specifying the range of neigbors to check between the data point to understand whether the data is an outlier or not. In order to determine the value of n_neigbor, firstly, I tried 20 which is suggested in the documentation, and after that I tried 5. After this trials, I observed that when the number increases the accuracy drops, also, I tried 3 and 4, however, the performance when 5 is assigned was the best so I chose 5 as n_neigbor.

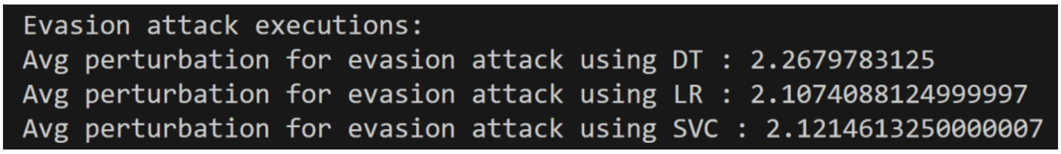
The accuracy of my algorithm is changing between 33% and 45%. It is possible to increase the accuracy by making a trade of and specifying the contamination parameter to a larger value other than the p value, however, this would cause detecting more outlier points than the actual.

Question 3

While developing my attack strategy, I checked the BankNote_Authentication dataset and released a pattern. Although, there are some exceptions, the more negative valued features the data have, the more probability that the label is 1. Therefore, in order to evade the model, I first make the actual model to predict the result. After the prediction, if the predicted label is 1, I found the smallest positive valued feature and multiplied it with -1. If the predicted label is 0, I found largest negative valued feature and multiplied it with -1.

The reason I find the smallest positive and the largest negative is to minimize the perturbation rate. However, some 1 labeled rows has all positive features. In order to handle this exception, I check whether the features has any negative value. If there are only positive values features, I increased the value of the smallest feature until the label is changed.

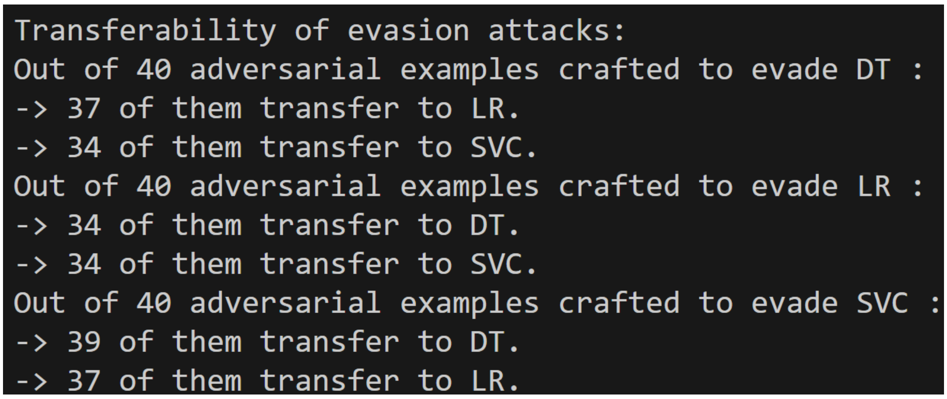
This approach has high perturbation rate (Figure 1) since we are directly changing the sign of the features, but it is less time consuming than manipulating the feature values with small values.



```
Evasion attack executions:
Avg perturbation for evasion attack using DT : 2.2679783125
Avg perturbation for evasion attack using LR : 2.1074088124999997
Avg perturbation for evasion attack using SVC : 2.1214613250000007
```

Figure 1

Question 4



```
Transferability of evasion attacks:
Out of 40 adversarial examples crafted to evade DT :
-> 37 of them transfer to LR.
-> 34 of them transfer to SVC.
Out of 40 adversarial examples crafted to evade LR :
-> 34 of them transfer to DT.
-> 34 of them transfer to SVC.
Out of 40 adversarial examples crafted to evade SVC :
-> 39 of them transfer to DT.
-> 37 of them transfer to LR.
```

Figure 2

As it can be clearly seen from the figure 2, my evasion attack has a very high cross-model transferability. If we compare the results from homework pdf and my experimental results, we can say that the higher perturbation, the higher cross-model transferability. Therefore, if we want our strategy to have a high cross-model transferability, we need to have highly perturbed examples.