Q1. t-SNE, PCA and SVD are all algorithms used for dimension reduction. These algorithms help visualize and interpret high dimensional data.

- PCA - Dataset is rotated such that rotated features are statistically uncorrelated and a subset of these features are selected. The directional along which the variance is maximum will be the directional where features are most correlated and the directional with the most information orthogonal to the previous direction is the second component. These two directions constitute the PRINCIPAL COMPONENTS.
- t-SNE - finds a 2-D representation of the dataset while preserving the distance between points. It preserves information indication neighboring points and distant points while reducing the dimension.
- SVD - is a matrix decomposition method which reduces a matrix into its constituent parts. It produces a matrix of a lower dimension which is a close approximation. Unlike PCA, this estimator does not center the data before computing the singular value decomposition.

Stratified Sampling maintains the ratio of class percentages during train test split i.e if 10% of class A in present in the data then 10% of class A will be present in both the train and test split.

Class 0 - 0.09

Class 1 - 0.11

Class 2 - 0.09

Class 3 - 0.10

Class 4 - 0.09

Class 5 - 0.09

Class 6 - 0.10

Class 7 - 0.10

Class 8 - 0.09

Class 9 - 0.09

The distribution shows that the number of classes is similar and there is no class imbalance.

Accuracy t-SNE - 73%

Accuracy PCA - 45%

Accuracy SVD - 32%

PCA and SVD both have worse accuracy than t-SNE. Dimension reduction algorithms are good for visualizing data but do not ensure a good performance while training the model. t-SNE maintains distance between points which might be why the ML model is able to perform better.

Using t-SNE after PCA, is a more spread out representation of the data. It shows that after applying PCA, even though some labels are clearly separable, the majority labels are intertwined which probably is the explanation for the poor performance.

Using t-SNE after SVD also clearly demonstrates why the results were so poor because there is no clear distinction between the labels.

Q2. Sklearn MSE = 100.635

Bias = 0.0000002506

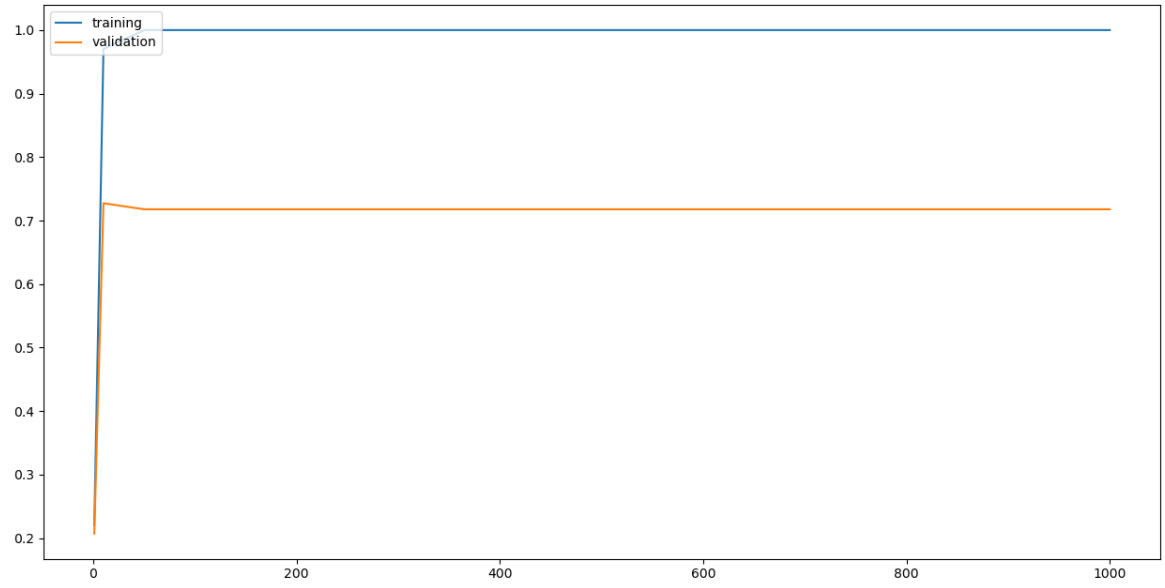
Variance = 0.2253

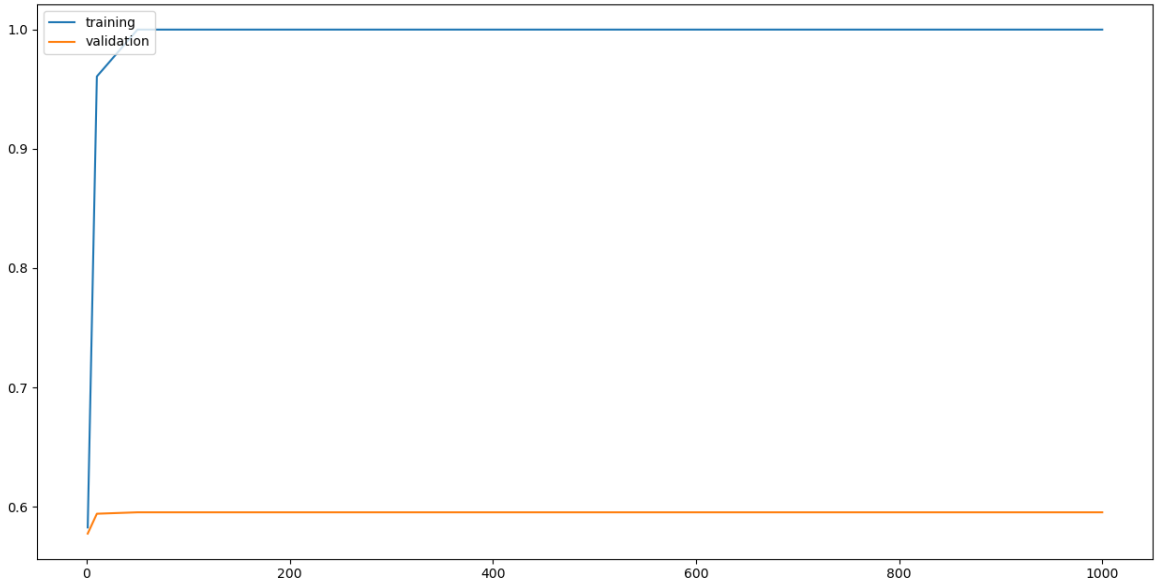
MSE – $Bias^2$ –Variance = 100.410

This value obtained is the Irreducible Uncertainty which is the randomness which is a part of every dataset. Dataset C comprise of gender and height to predict weights, and in this case the uncertainty is large because there are multiple other factors that contribute to the weight of a person and the value obtained is an approximation of contribution from other sources.

Q3. A) For best validation accuracy, 10 depth on the 1st fold gives best performance. For best training accuracy, 1000 depth on the 3rd fold gives the best performance.

b)





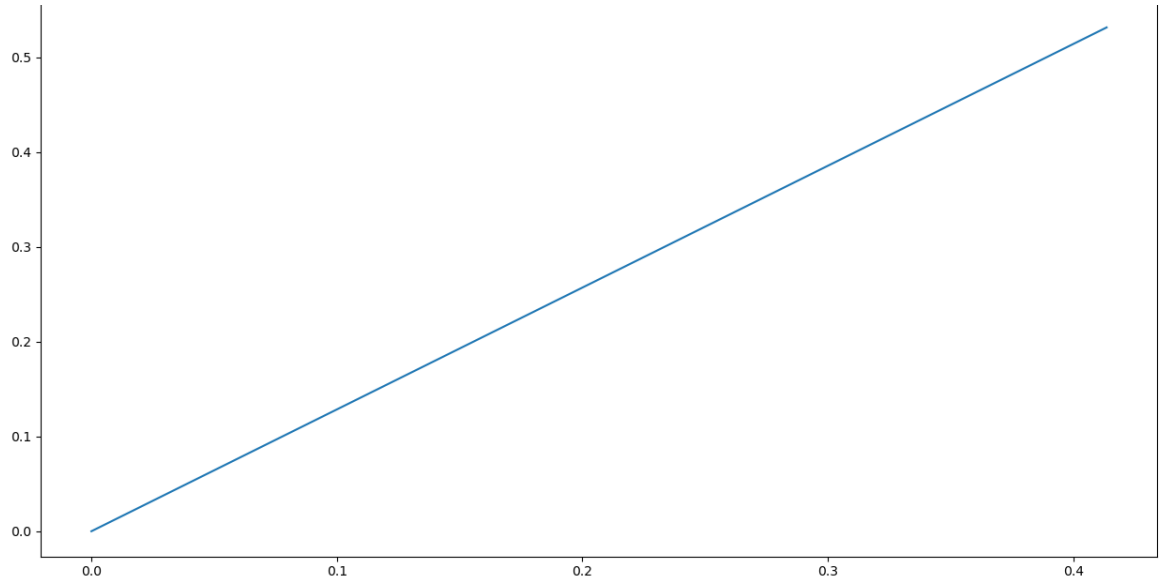
For both datasets a and b, maximum training accuracy 100% is reached at around 100 depth and the maximum validation accuracy, 0.72 for dataset a is reached at depth 10 and 0.59 for dataset b is reached at depth 50 and goes on slightly decreasing till 1000 depth. The model begins to start overfitting the data even though it might be not by much as the validation accuracy is above 60 in both cases but considering the fact that training accuracy is a complete 100% we should be getting a better validation accuracy.

d) Sample output for decision tree on dataset a

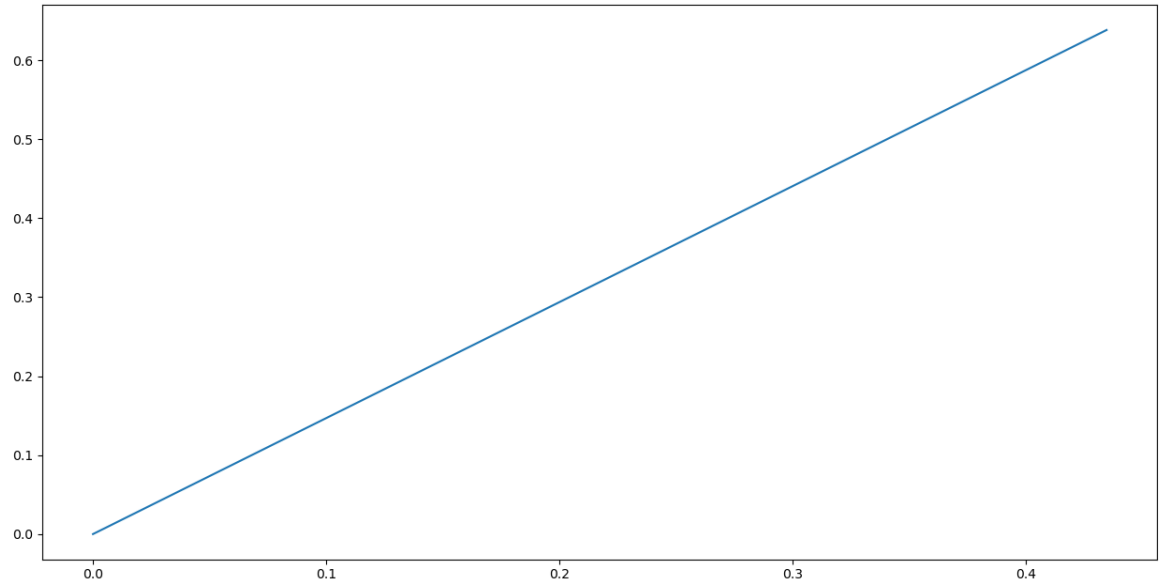
```
Decision Tree, a
accuracy 0.5584415584415584
precision 0.7166666666666667
recall 0.7166666666666667
 Precision       Recall  F1
0 0.82           0.82    0.82
1 0.79           0.89    0.8370238095238095
2 0.71           0.67    0.6894202898550725
3 0.7            0.68    0.6898550724637681
4 0.79           0.72    0.7533774834437086
5 0.6            0.62    0.6098360655737705
6 0.83           0.75    0.7879746835443037
7 0.77           0.79    0.7798717948717949
8 0.46           0.46    0.46
9 0.64           0.71    0.6731851851851851
Macro Avg Precision 0.711
Macro Avg Recall 0.711
Macro Avg F1 0.7100544384461414
[[64.  0.  3.  2.  0.  3.  1.  2.  3.  0.]
 [ 0. 77.  0.  3.  1.  3.  0.  0.  3.  0.]
```

```
[ 1.  5. 61.  4.  2.  3.  4.  3.  5.  3.]
[ 0.  1.  4. 51.  1.  6.  0.  1.  6.  5.]
[ 1.  4.  5.  1. 63.  2.  2.  3.  2.  4.]
[ 3.  1.  1.  3.  1. 43.  6.  1.  9.  1.]
[ 3.  2.  2.  1.  6.  3. 76.  2.  7.  0.]
[ 0.  0.  3.  0.  1.  0.  2. 75.  3. 11.]
[ 5.  5.  6.  7.  1.  6.  0.  3. 35.  8.]
[ 1.  2.  1.  1.  4.  3.  1.  7.  3. 57.]]
```

ROC for Random Forest on Dataset B



ROC for Naïve Bayes on Dataset B



Both plots show that both the models have poor recall which is true because for both the models the recall is around 0.5 and the accuracy is around 0.4. This is why we get a straight line instead of how a usual ROC graphs look like because the

performance of the model is worse than randomly predicting which should theoretically give 0.5 accuracy.

Q4.

```
Dataset1

train 0.8564625850340136
test 0.8158730158730159
sklearn acc train 0.641156462585034
sklearn acc test 0.6071428571428571

 Dataset2

train 0.5874149659863945
test 0.5761904761904761
sklearn acc train 0.5744897959183674
sklearn acc test 0.5738095238095238
```

Q5)a,c)

Entropy $= -\left(\frac{9}{14}\log\frac{9}{14} + \frac{5}{14}\log\frac{5}{14}\right)$

$= 0.94$

Infogain(climate) =

$\Rightarrow 0.94 - \frac{4}{14}\left(-\left(\frac{1}{2}\log\frac{1}{2} + \frac{1}{2}\log\frac{1}{2}\right)\right) -$

$\frac{6}{14}\left(-\left(\frac{4}{6}\log\frac{4}{6} + \frac{2}{6}\log\frac{2}{6}\right)\right) - \frac{4}{14}\left(-\left(\frac{3}{4}\log\frac{3}{4} + \frac{1}{4}\log\frac{1}{4}\right)\right)$

$\rightarrow$ 0.029

Infogain(wind)

$\Rightarrow 0.94 - \frac{8}{14}\left(-\left(\frac{6}{8}\log\frac{6}{8} + \frac{2}{8}\log\frac{2}{8}\right)\right)$

$- \frac{6}{14}\left(-\left(\frac{3}{6}\log\frac{3}{6} + \frac{3}{6}\log\frac{3}{6}\right)\right)$

$= 0.048$

Info gain(outlook)

$\Rightarrow 0.94 - \frac{5}{14}\left(-\left(\frac{3}{5}\log\frac{3}{5} + \frac{2}{5}\log\frac{2}{5}\right)\right)$

$- \frac{4}{14}\left(\frac{4}{4}\log\frac{4}{4} + \frac{0}{4}\log\frac{0}{4}\right)$

$- \frac{5}{14}\left(-\left(\frac{3}{5}\log\frac{3}{5} + \frac{2}{5}\log\frac{2}{5}\right)\right)$

$= 0.247$

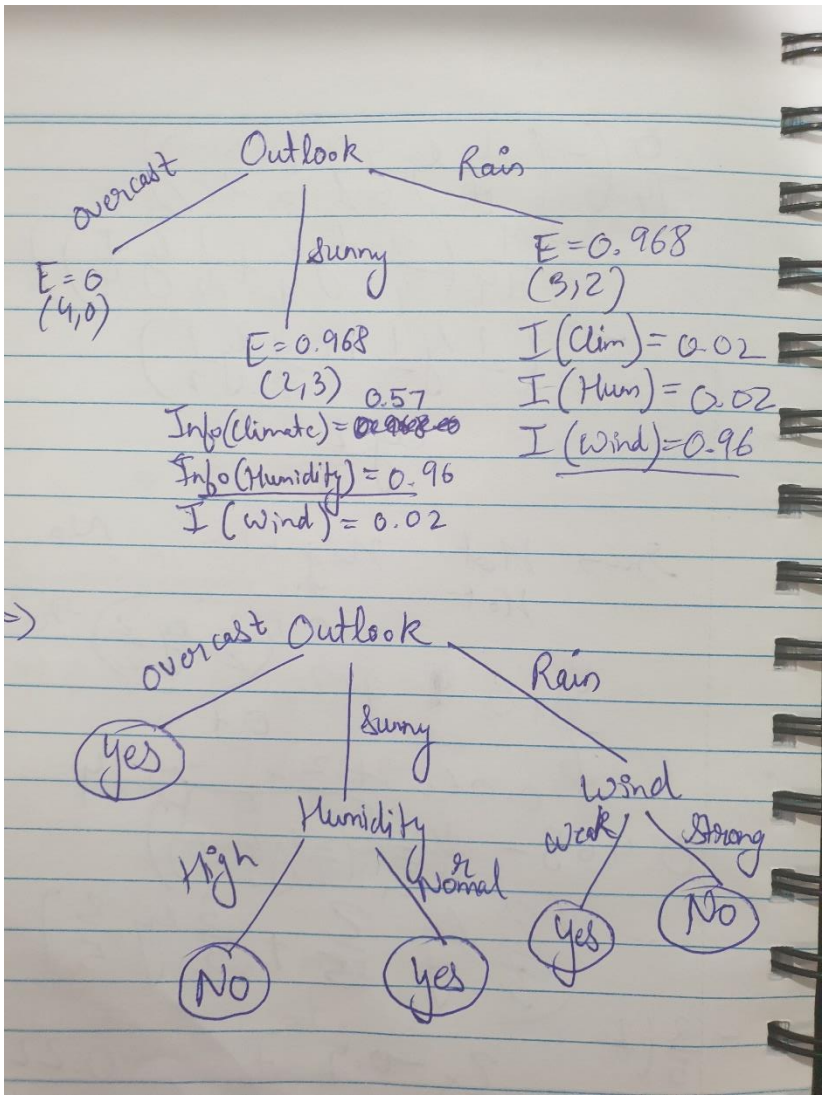
Infogain(Humidity)

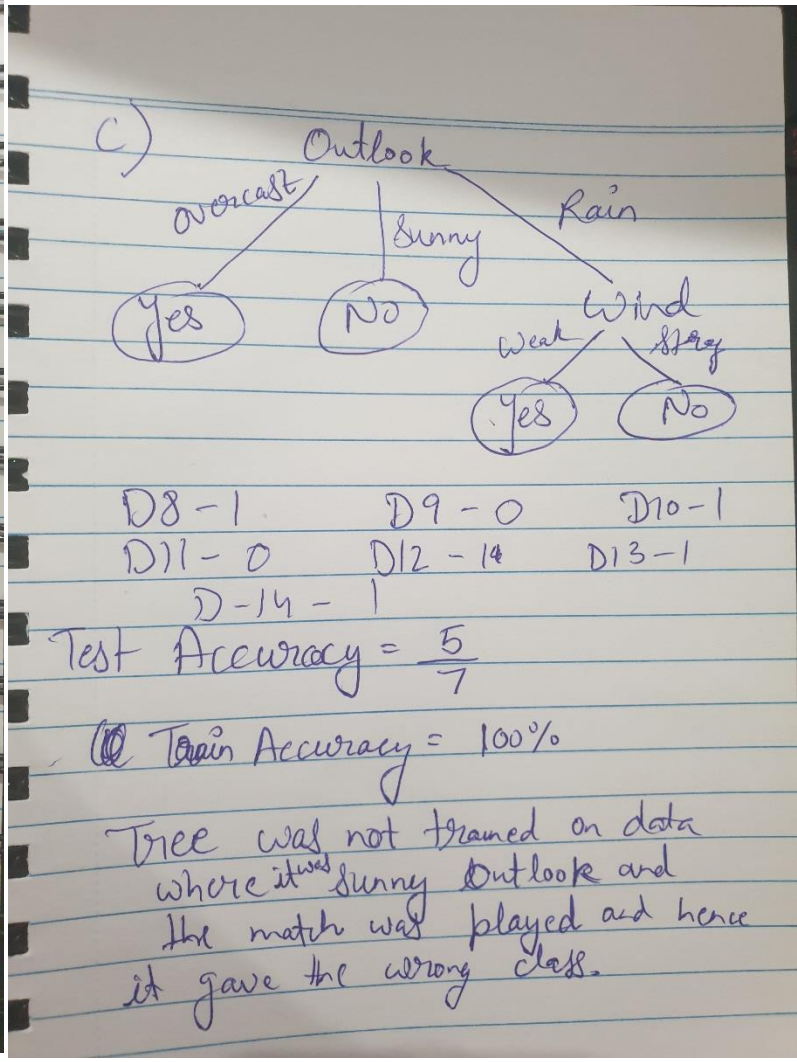
$\Rightarrow 0.94 - \frac{7}{14}\left(-\left(\frac{3}{7}\log\frac{3}{7} + \frac{4}{7}\log\frac{4}{7}\right)\right)$

$- \frac{7}{14}\left(-\left(\frac{6}{7}\log\frac{6}{7} + \frac{1}{7}\log\frac{1}{7}\right)\right)$

$= 0.151$

Info gain is max for outlook so we split it.

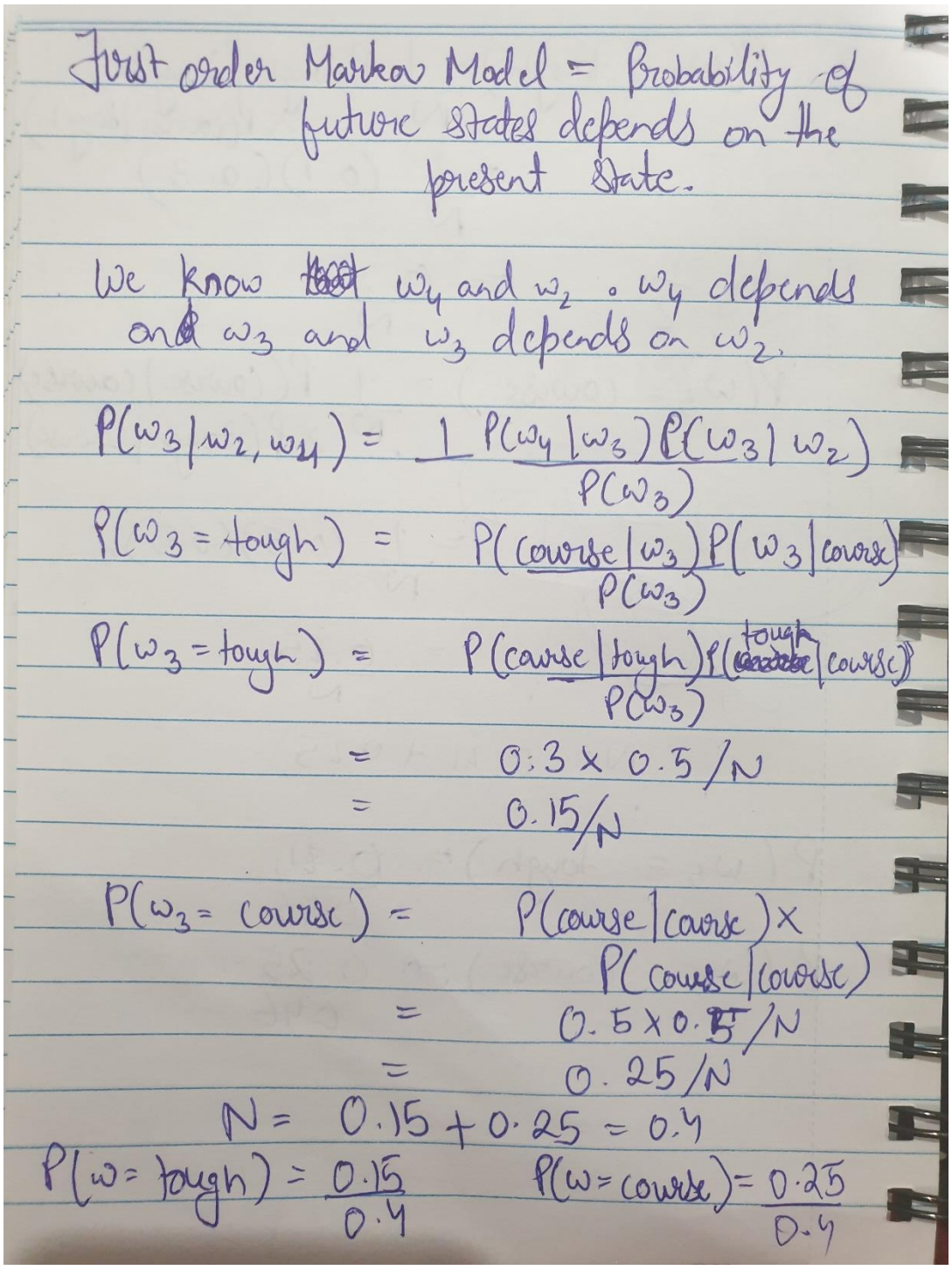




b) Yes, If we take the subset [D1,D2,D4,D10,D11,D12]. In this subset, we have only 2 values of climate and for every hot climate the only outcome is No while for every mild climate the only outcome is Yes. In this way entropy of Climate becomes 0 and in this tree the only node will be climate.

d)Limit tree depth, limit the number of features are some of the ways to prevent overfitting.

Q6.



Q7)a) Logistic Regression fits a line to fit the data in the best possible way. This obviously would not be good if a single linear boundary cannot divide the data into accurate classes and this is the case with most data and also the reason why logistic regression is a much less powerful and less complex algorithm compared to decision trees which divide the data into small regions. Also, logistic regression

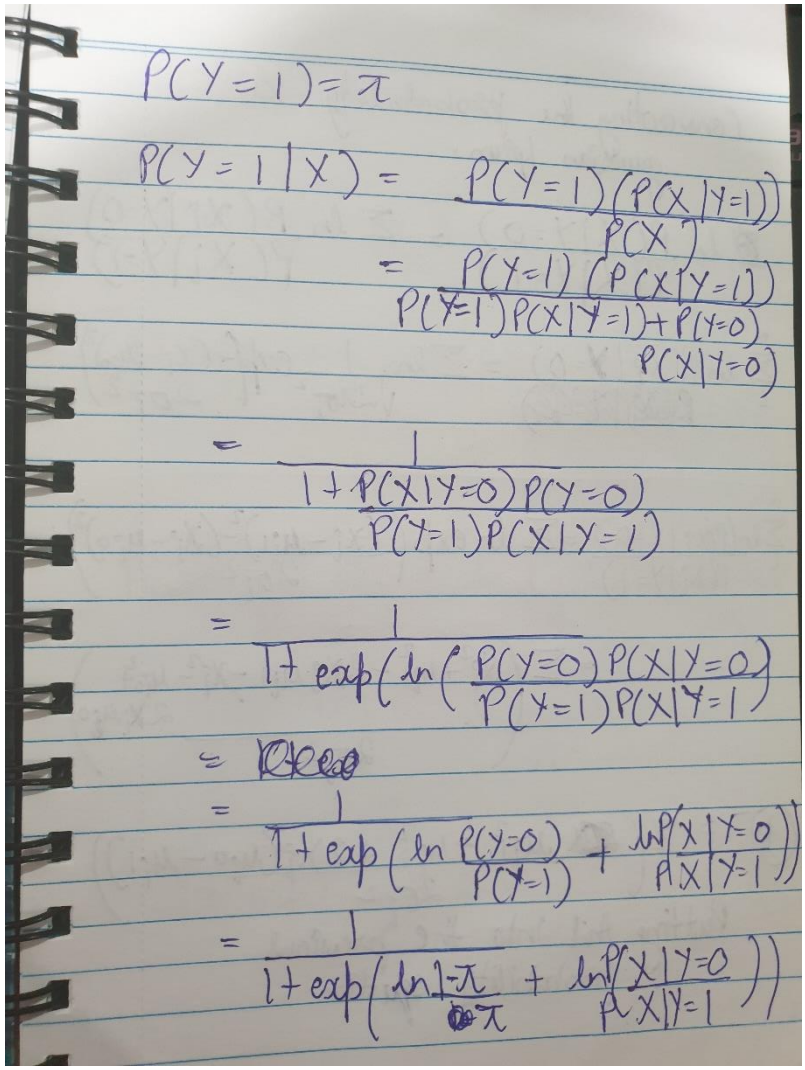
assumes that the features are independent whereas decision trees do not and hence can establish relationship between such features.

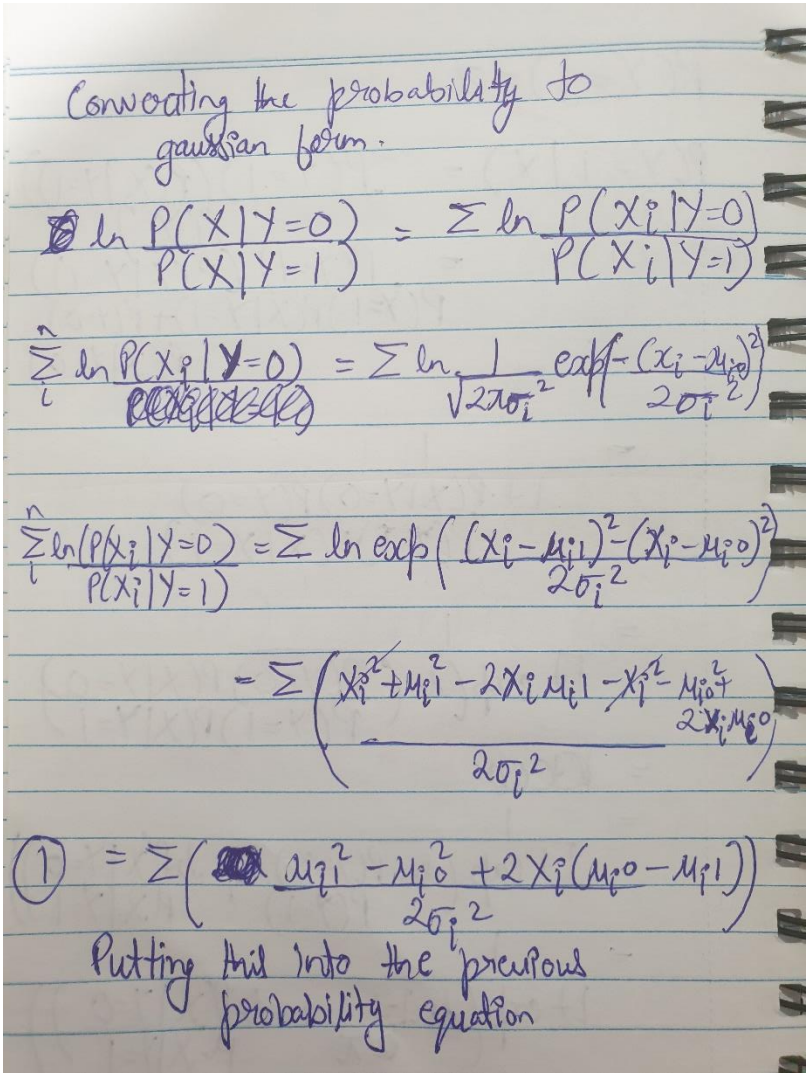
b) When there is no clear separation between classes, a linear boundary may act as a better and in this case logistic regression will outperform decision tree. Also, as the decision tree regions go on smaller it will tend to over fit the data.

c) Yes, decision tree can classify these vectors. For every x1, there will be a threshold x2 above which y will be 1 and below which y will be −1. To split the entire data, we will need a log(n) depth tree and so for to classify points max depth will be log(n)+1. Upper bound = O(logn)

d) in this case, we cannot assign a threshold to x2 and so we would need to consider different values of x2 along with x1. The depth on the tree will be log(n) for x2 and log(n) for x1, so the total depth will be 2log(n). Upper bound = O(logn)

Q8.





$$P(Y=1|X) =$$

$$= \frac{1}{1 + \exp\left(\ln \frac{1-\pi}{\pi} + (1)\right)}$$

$$= \frac{1}{1 + \exp\left(\ln \frac{1-\pi}{\pi} + \sum \left(\frac{\mu_{0i} - \mu_{i1}}{\sigma_i^2}\right) X_i + \left(\frac{\mu_{i1}^2 - \mu_{i0}^2}{2\sigma_i^2}\right)\right)}$$

Assuming $w_0 = \ln \frac{1-\pi}{\pi} + \sum \frac{\mu_{i1}^2 - \mu_{i0}^2}{2\sigma_i^2}$

$$w_i = \frac{\mu_{i0} - \mu_{i1}}{\sigma_i^2}$$

$$= \frac{1}{1 + \exp(w_0 + \sum w_i X_i)}$$