Predictive Analytics Mid term exam

A.

1. F
2. F
3. T
4. F
5. F
6. F
7. F
8. F
9. F
10. T
11. F
12. T
13. T
14. T
15. T

B.

1.

B1)

```
PCA (double[][] M , int n , int m) : {
    double[][] data = M
    int num components;
    double[] mean ;
    for i 0 to n :
        for j 0 to m :
            mean[j] + = M[i,j]

    for j 0 to m :
        Mean[j] / = n;

    for i 0 to n :
        for j 0 to m :
            M[i,j] = Mean[j]

    CoVM = matrixmmit (M^T , M)
(Eve , Eval) = Get EigenVector & Values (CoVM)

    Eve , Eval = Sort( Evec , Eval). desc()
    double[][] ~~EvectorMatrix~~ EigenMatrix = EigenVectors
                                               [: numComponent]
    ~~features = EvectorMatrix^T~~ .

    features = matrixmmit ( EigenMatrix^T , data)
    return features
}
```

2.

B) 2) $M = \begin{bmatrix} 1 & 2 \\ 2 & 1 \\ 3 & 5 \\ 4 & 3 \end{bmatrix}$

$$M^T \cdot M = \begin{bmatrix} 1 & 2 & 3 & 4 \\ 2 & 1 & 5 & 3 \end{bmatrix} \cdot \begin{bmatrix} 1 & 2 \\ 2 & 1 \\ 3 & 5 \\ 4 & 3 \end{bmatrix}$$

$$= \begin{bmatrix} 30 & 31 \\ 31 & 39 \end{bmatrix}$$

→ Eigen Values

$$(30-\lambda)(39-\lambda) - 31 \times 31 = 0$$

$$\lambda^2 - 60\lambda - 61 = 0$$

$$(\lambda - 61)(\lambda + 1) = 0$$

$\lambda = 3.17$ or $\lambda = 65.82$

For $\lambda = 65.82$

$$\begin{bmatrix} 30 & 31 \\ 31 & 39 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = 65.82 \begin{bmatrix} x \\ y \end{bmatrix}$$

$$30x + 31y = 65.82x$$

$$31x + 39y = 65.82y$$

$$y = \frac{35.82}{31} x \Rightarrow y = 1.15x$$

Eigen Vectors will be

$$\overrightarrow{PC_1} = \begin{bmatrix} 1/\sqrt{2.35} \\ 1.15/\sqrt{2.35} \end{bmatrix}$$

Ily to $\lambda = 3.17$

$$30x + 31y = 3.17x$$

$$31x + 39y = 3.17x.$$

$$y = -0.865x$$

Here Eigen Vectors will be

$$\overrightarrow{PC_2} = \begin{bmatrix} -1/\sqrt{2} \\ 1/\sqrt{2} \end{bmatrix} \begin{bmatrix} -1/\sqrt{1.64} \\ 0.865/\sqrt{1.64} \end{bmatrix}$$

Eigen Matrix :

$$\overrightarrow{PC} = \begin{bmatrix} 1/\sqrt{2} & -1/\sqrt{2} \\ 1/\sqrt{2} & 1/\sqrt{2} \end{bmatrix} \begin{bmatrix} 1/\sqrt{2.3} & -1/\sqrt{1.64} \\ 1.15/\sqrt{2.3} & 0.865/\sqrt{1.64} \end{bmatrix}$$

Now, users co-ordinates will be

$$\vec{M} \times \overrightarrow{PC} = \begin{bmatrix} 1 & 2 \\ 2 & 1 \\ 3 & 5 \\ 4 & 3 \end{bmatrix} \cdot \overrightarrow{PC} \begin{bmatrix} 1/\sqrt{2} & -1/\sqrt{2} \\ 1/\sqrt{2} & 1/\sqrt{2} \end{bmatrix}$$

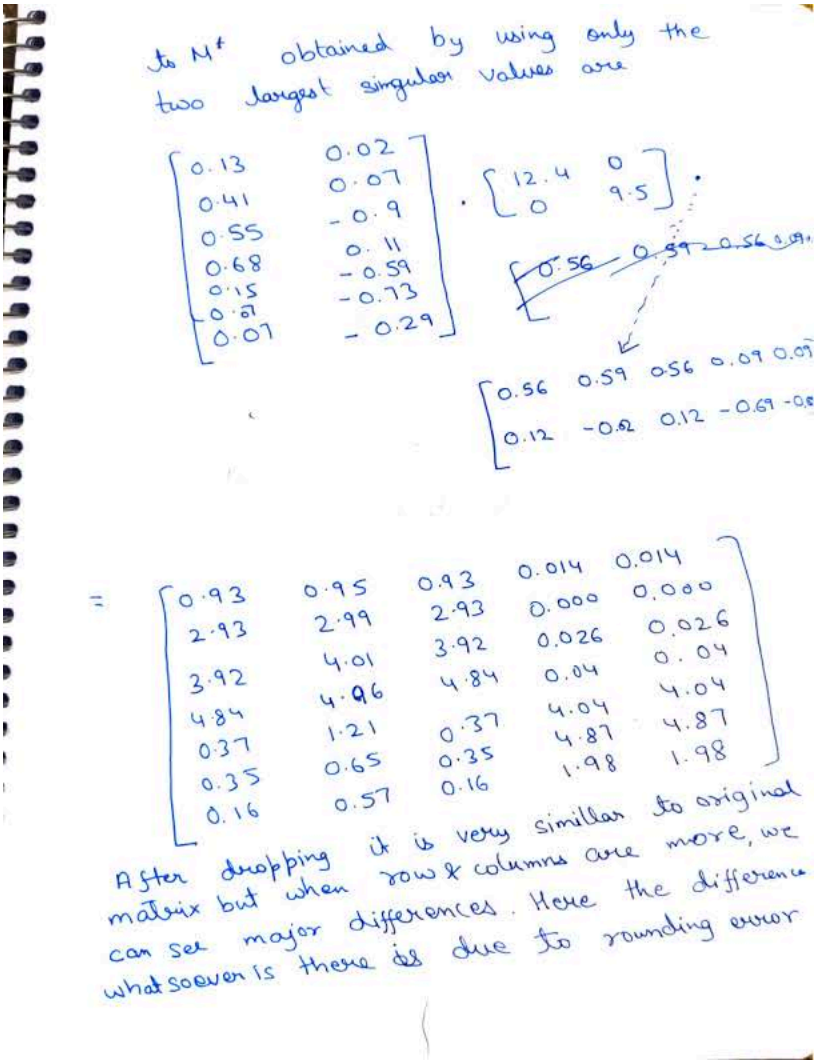
$$= \begin{bmatrix} 1/\sqrt{2} & \\ -1/\sqrt{2} & \\ 2/\sqrt{2} & \\ -2/\sqrt{2} & \end{bmatrix} \begin{bmatrix} \frac{3.3}{\sqrt{2.3}} & \frac{0.63}{\sqrt{1.64}} \\ \frac{3.3}{\sqrt{2.3}} & \frac{-1.13}{\sqrt{1.64}} \\ \frac{8.75}{\sqrt{2.3}} & \frac{1.1325}{\sqrt{1.64}} \\ \frac{13.8}{\sqrt{2.3}} & \frac{-1.405}{\sqrt{1.64}} \end{bmatrix}$$

Convert in 4X1 matrix

$\vec{M} \times \vec{PC} = \begin{bmatrix} 3/\sqrt{2} \\ 3/\sqrt{2} \\ 4\sqrt{2} \\ 7/\sqrt{2} \end{bmatrix} \begin{bmatrix} \frac{3.3}{\sqrt{2.3}} \\ \frac{3.3}{\sqrt{2.3}} \\ \frac{8.75}{\sqrt{2.3}} \\ 13.8/\sqrt{2.3} \end{bmatrix}$

3) Representing U, Σ & V components of SVD of Matrix M becomes a difficult task when they are quite big. So, we need to reduce the dimensionality of three matrices by setting the smallest value to zero. If set s smallest ~~small~~ singular values to 0 then we can also eliminate corresponding columns of U X V. The given matrix has three singular matrices. Suppose we wants reduce dimension to 2, then we start setting smallest singular value 1.3 to zero, then when columns are multiplied by zero 0, the resultant matrix has zero columns as well. That is, approximati

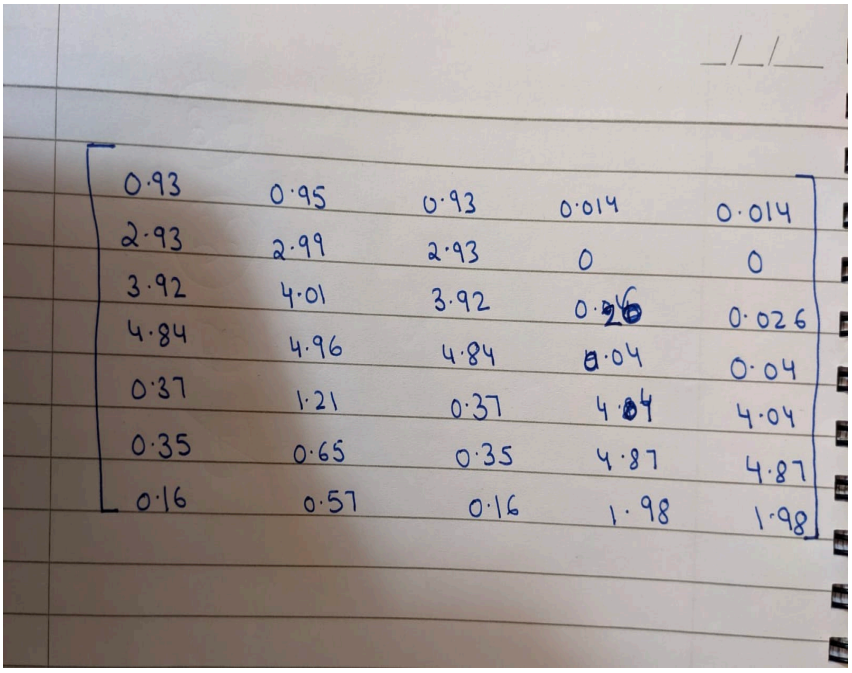
Bb



Here rank of matrix is r =3 because for the purpose of explanation, 1st , 6th and 7th rows are independent and no four rows are independent

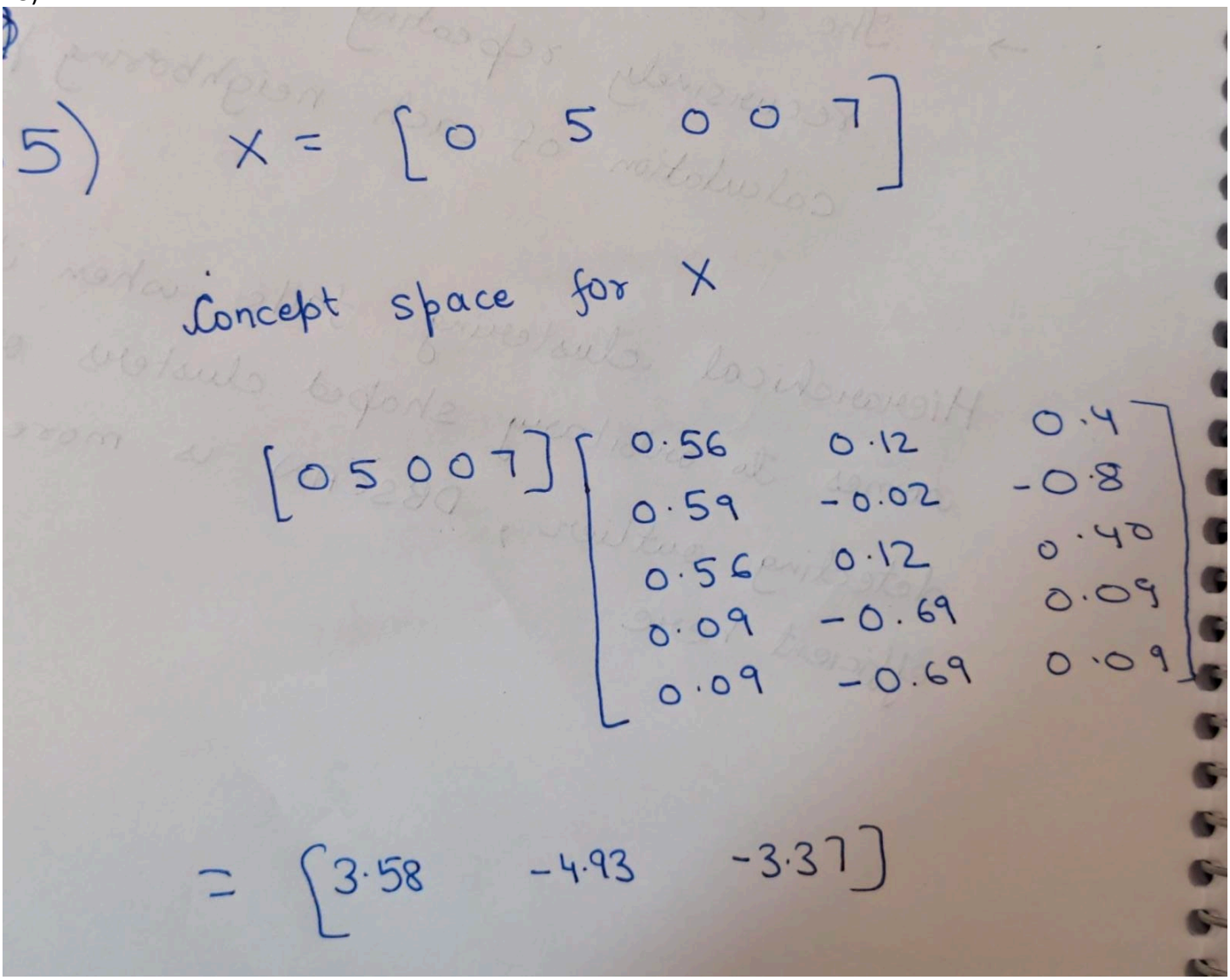
B 4)

- The decomposition of question given has three singular values. Suppose we want to reduce the number of dimensions to two.
- Then we set the smallest of the singular values, which is 1.3, to zero.
- The effect on the expression is that the third column of U and the third row of V T are multiplied only by 0's when we perform the multiplication, so this row and this column may as well not be there.
- That is, the approximation to M' obtained by using only the two largest singular values is that shown in Fig. below.
- The resulting matrix is quite close to the matrix M ' of our original matrix. Ideally, the entire difference is the result of making the last singular value be 0. However, in this simple example, much of the difference is due to rounding error caused by the fact that the decomposition of M' was only correct to two significant digits.



0.93	0.95	0.93	0.014	0.014
2.93	2.99	2.93	0	0
3.92	4.01	3.92	0.026	0.026
4.84	4.96	4.84	0.04	0.04
0.37	1.21	0.37	4.04	4.04
0.35	0.65	0.35	4.87	4.87
0.16	0.57	0.16	1.98	1.98

B5)

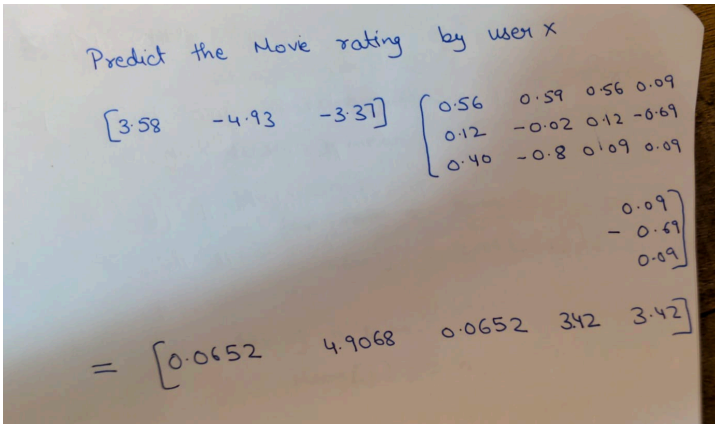


5) $X = [0 \quad 5 \quad 0 \quad 0 \quad 7]$

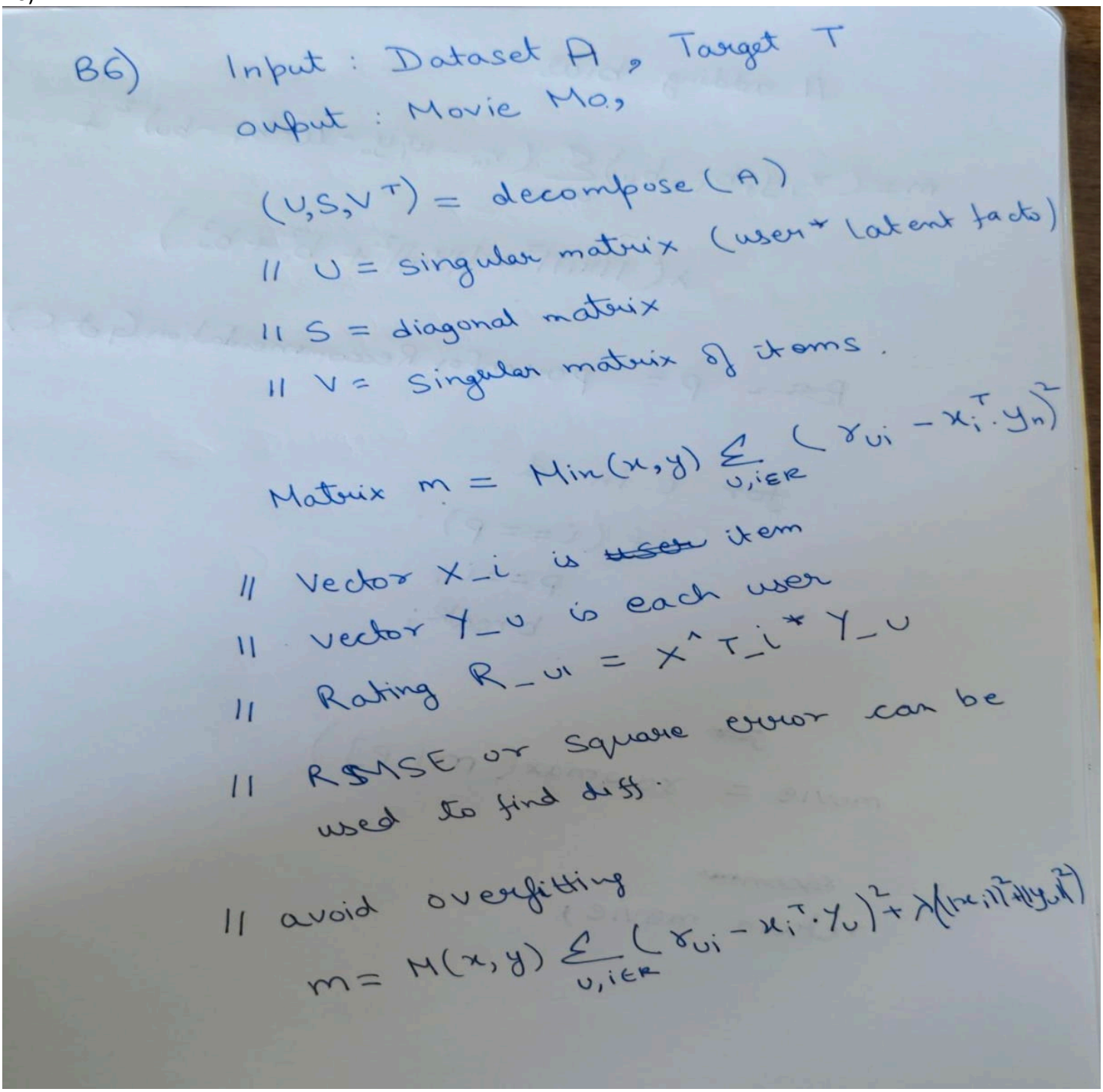
Concept space for X

$$[0\ 5\ 0\ 0\ 7] \begin{bmatrix} 0.56 & 0.12 & 0.4 \\ 0.59 & -0.02 & -0.8 \\ 0.56 & 0.12 & 0.40 \\ 0.09 & -0.69 & 0.09 \\ 0.09 & -0.69 & 0.09 \end{bmatrix}$$

$$= [3.58 \quad -4.93 \quad -3.37]$$


Predict the Move rating by user x
[3.58 -4.93 -3.37]
0.56 0.59 0.56 0.09
0.12 -0.02 0.12 -0.69
0.40 -0.8 0.09 0.09
0.09
-0.69
0.09
= [0.0652 4.9068 0.0652 3.42 3.42]

B6)


B6) Input : Dataset A, Target T
ouput : Movie Mo,
(U,S,V^T) = decompose(A)
// U = singular matrix (user * latent facto)
// S = diagonal matrix
// V = Singular matrix of items.
Matrix m = Min(x,y) Σ_{u,i∈R} (r_ui - x_i^T . y_n)^2
// Vector X_i is item
// vector Y_u is each user
// Rating R_ui = X^T_i * Y_u
// RMSE or Square error can be used to find diff.
// avoid overfitting
m = M(x,y) Σ_{u,i∈R} (r_ui - x_i^T . Y_u)^2 + λ(|x_i|^2 + |y_u|^2)

// adding bias.

$$m = (x, y, b_i, b_u) \sum_{u,i \in R} (r_{ui} - x_i^T y_u - \mu - b_i - b_u)^2 +$$

$$\lambda(||x_i||^2 + ||y_u||^2 + b_i^2 + b_u^2)$$

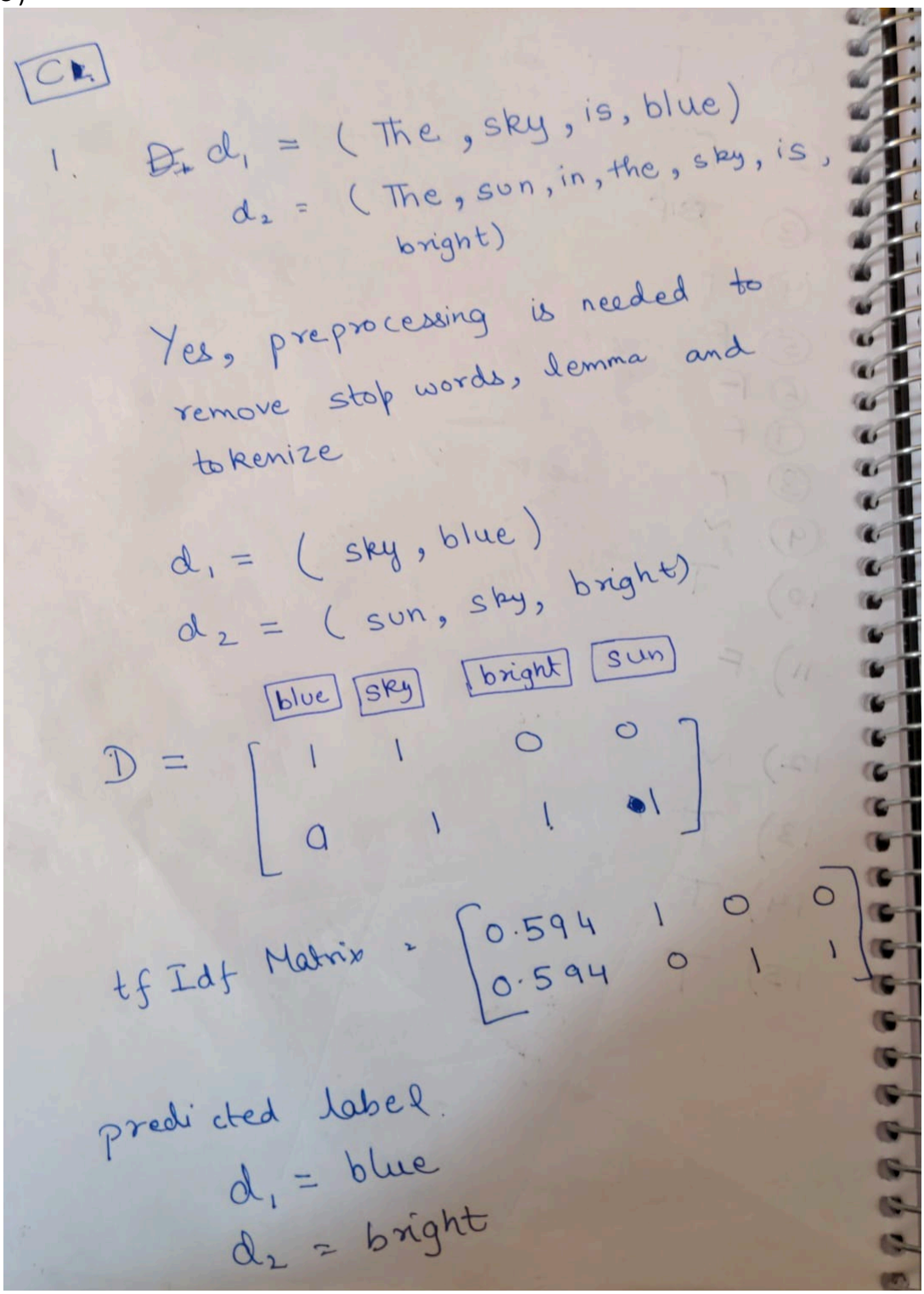
p = personforRecommendationGet()

```
for i in d:
    if (i == p):
        p == i;
        break;
```

movie = rowmax(m[p])

return movie;

C)


C)
1. d1 = (The , sky , is , blue)
d2 = (The , sun , in , the , sky , is , bright)
Yes, preprocessing is needed to remove stop words, lemma and tokenize
d1 = (sky , blue)
d2 = (sun , sky , bright)
blue
sky
bright
sun
D = [1 1 0 0 ; 0 1 1 1]
tf Idf Matrix = [0.594 1 0 0 ; 0.594 0 1 1]
predicted label.
d1 = blue
d2 = bright

Preprocessing.

→ When the document is given it contains lot of noise.

→ This noise needs to be removed to do better analytics

→ In this question, we can see lot of stop words ie: 'the', 'is' which needs to removed. Removal of stop words although is not necessary in all cases for eg.. we want to classify novels based on their writers. We need to check style in that case. Not relevant to our case.

→ Then we use NLP lib to check if there are any NGrams. Further to lemma, annotate and tokenize.

C2

d_3 : (sun , sky)

↑

After preprocessing as explained in previous question.

$$tf\,Idf = \begin{matrix} d_1 \\ d_2 \\ d_3 \end{matrix} \begin{bmatrix} 0.712 & 1.40504 & 0 & 0 \\ 0.712 & 0 & 1 & 1.405 \\ 0.712 & 0 & 1 & 0 \end{bmatrix}$$

For cosine, we will use tfIdf matrix

d_1 to d_1 → 1

$$\text{cosine}(A,B) = \frac{\text{dot}\,\vec{A} \cdot \text{dot}\,\vec{B}}{\sqrt{\sum_{i=0}^{2} A_i^2} \times \sqrt{\sum_{i=0}^{2} B_i^2}}$$

After calculating d_1 to d_2, d_1 to d_3 & d_2 to d_3, d_2 to d_3 are simmilar to each other. ↳ (0.8).

→ Here, vector A & vector B in formula are rows in tf Idf matrix

→ So we take out cosine distance ~~fro~~ between each document.

C3

~~Step I : Read file~~

~~Step II : Pre Pass document~~

```
Analyze() : {
    Readfile rf = Read File (X.txt)
pre = preprocessDocument (rf)
tf = ~~Sort~~ . build Tf Matrix (pre)
g = sort TF Martrix (tf).
    print (g[0]) . }
}
preprocessDocument (rf) : {
    p = removeStopWords(rf)
pipeline = Stanford CoreNLP (Props).
        props → annotators, tokenize,
                ssplit, pos, lemma,
                ner"
for each word in document :
    modifiedText . add ( SCN . do props(p
                        (pipeline))
```

```
NGrams = NGrams( modified text)
getMatrix ( N Grams)
}
```

C4)

- Named entity recognition (NER) is a process that seeks to locate and classify named entities in text into pre-defined categories which are named entities such as names of persons, organizations, locations, expressions of times, quantities, monetary values, percentages, etc.

  Example - European authorities fined Google.

  European, JJ
  Google, NNP

  which are category definitions in the "nltk" library

- Stanford NLP group is the group which is responsible for creation of libraries such as nltk which is one of the most commonly used libraries for NLP tasks such as work tokenization, POS tagging. They provide pre trained models and functionality in the library. They are also the developers for the GloVe word embeddings concept which is highly used in the current scenarios. The group is a part of the Stanford AI Lab.

- Stemming is the process of reducing inflections from words. It is done by removing ends of the words and hoping that the stemming happened properly. Main aim is to go to the root word so that a word and its derivative forms are not considered as separate forms. Ex - likes, liked will be stemmed to like which is the root word
- A sliding window is a window that takes sublists that runs over a main list incrementally by moving the window across the main list. An example of using this is while computing Ngrams. Window size defines the length of the window that is sliding over the list. This algo is also useful for computing general CS concepts such as running average.

- Lemmatization is the process of reducing inflectional terms to their root word by using vocabulary and morphological rules so that the words formed after lemmatization are of the correct form which is a problem faced in stemming. The words are converted to lemmas which is the root word. Aim is same as stemming so that all inflected words are reduced to the same root word form. Ex - "Better" with stemming will not give a correct output. "Better" with lemmatization will give "good" which is the correct lemma.

- N-gram is a contiguous sequence of n words in textual data. It is a type of probabilistic language model which is used for predicting the next word in a given sentence. Before the advent of deep learning, n-gram model was heavily used in predicting next words when using soft keyboards in mobile phones. Here "n" can be unigram, bigram, etc. which signify the number of characters which are taken to predict the next characters.
- Document term matrix is a matrix that depicts the frequency of terms in a list of documents. The rows correspond to the documents and the columns correspond to the terms in those documents. Value for each element is the term frequency in all those documents. It functions as a lookup table where frequency of each term is required for further processing.
- Cluster visualization is the process of rendering the cluster data in an interactive map. By visualizing the data, it becomes easier to solve certain defects in the code and can see the emerging patterns. One such algorithm for cluster visualization is Principal Component Analysis (PCA) which is used for dimensionality reduction which in turn helps with the visualization since it is difficult to perceive higher dimensional data.

- Precision is the ratio of correctly predicted positive observations to the total predicted positive observations. This metric refers to false positive rates. High precision means low false positive.

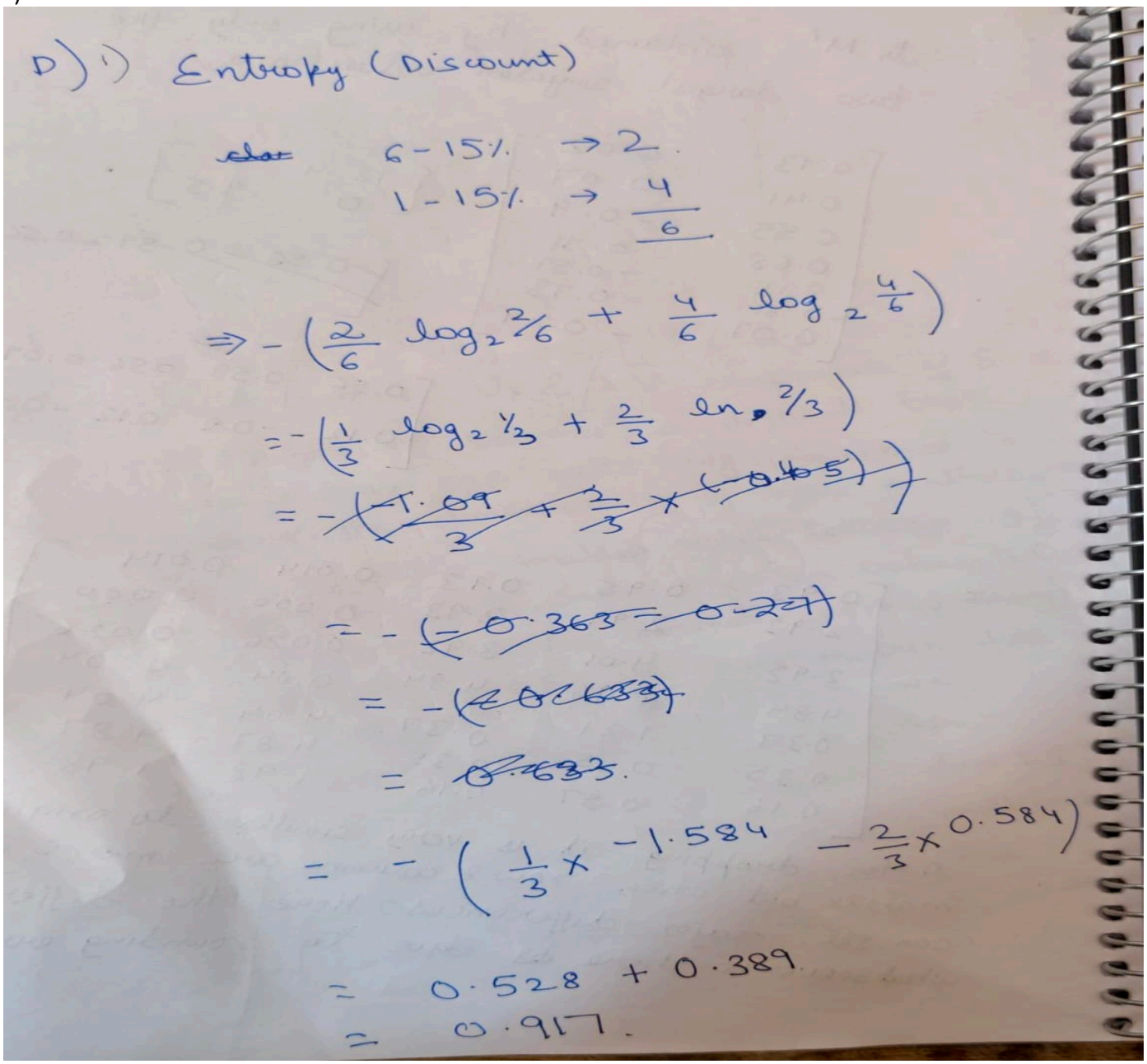
  Formula = (True_positive) / (True_positive + False_positive)
- Recall is the ratio of correctly predicted positive observations to the all observations in actual class. This metric refers to how many samples were predicted in the set of observations.

  Formula = (True_positive) / (True_positive + False_negative)

- F-measure is the weighted average of precision and recall which can also be described as the harmonic mean of precision and recall. This metric is better than accuracy since this can account for class imbalance.

  Formula = 2 * (Recall * Precision) / (Recall + Precision)

D)



Entropy (zipcode)

$= -(1/1 \log_2 1/1 \quad \ldots\ldots$ 6 times)

$= 0$

Gain (dis, zip) $= 0.917 - 0$

$= 0.917.$

zip code won't be considered

2) Entropy (LifyStyle)

Good $= 3\left(2(6-15\%) + 1(1\% - 5\%)\right)$

Average $= 3\left(2(1-5\%) + 1(6\% - 15\%)\right)$

$$\varepsilon(GA) = \frac{3}{6}\,\varepsilon\left(\frac{2}{3}, \frac{1}{3}\right) + \frac{3}{6}\,\varepsilon\left(\frac{3}{3}, \frac{0}{3}\right)$$

$$= \frac{1}{2} \times 0.918 = 0.459.$$

$IG = 0.917 - 0.459$

$= 0.459$

Entropy (Age)

$20+30\%(a) = 2(2(1-5\%))$

$30-40\%(b) = 3(2(6\%-15\%)+1(1-5\%))$

$40-50\%(c) = 1(1\%-5\%)$

$= \frac{3}{6}\,\varepsilon\left(\frac{2}{3},\frac{1}{3}\right) + \frac{2}{6}\,\varepsilon\left(\frac{3}{3},\frac{0}{3}\right) + \frac{1}{6}\,\varepsilon(1,0)$

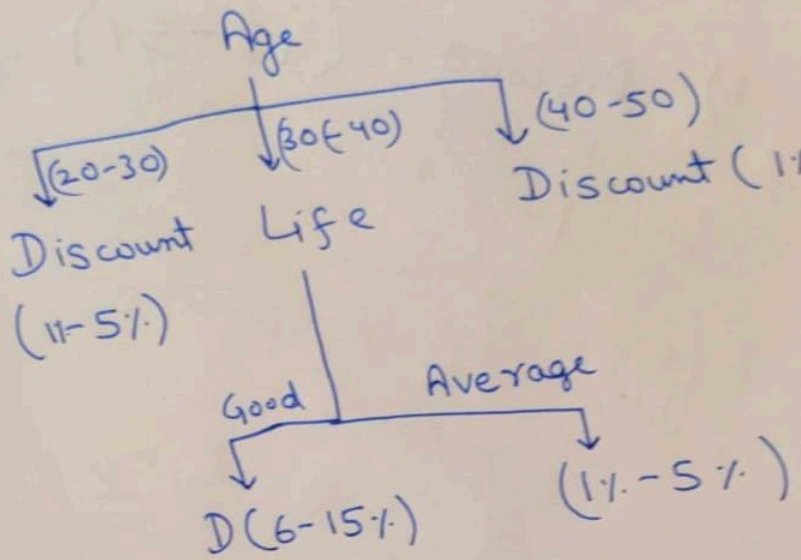
$= \frac{1}{2} \times 0.918$

$= 0.459.$

$IG = 0.917 - 0.459$

$= 0.459.$

3) Dicision tree.



→ ~~Since~~ Steps ~~I~~:

I) Calculate entropy of target

II) Calculate entropy of each attribute, then add it proportionally, to get total entropy of split. The resulting entropy is substracted from entropy before split

III) Choose attribute with largest IG, divide dataset by its branches and repeat the same process for each branch

IV) A branch with entropy of 0 is a leaf node if $\mathcal{E} \neq 0$, it needs further splitting

V) Run I-IV steps again on non-leaf branches, until all data is classified.

4) a) Discount - (1% - 5%)

b) Discount - (1% - 5%)

D 5)

D5) ~~FAS~~ FSA

The main drawback of FAS is that needs huge amount of computational power when number of features are large. Time complexity & space complexity increases drastically.

⇒ Designed algorithm:

Steps:

I) Begin with a model that contains all variables under consideration.

II) Start removing the least significant variable one after another

III) Until a pre-specified stopping rule is reached or until no variable is left.

Input : Dataset D, Target T

Output : Selected variables S

$S \leftarrow \phi$

$R \leftarrow V$

while S changes do:

$V_{worst} \leftarrow \text{argmax } PERF(S/V)$

if $Perf(S/V_{worst}) \geq PERF(S)$ then

$S \leftarrow S/V_{worst}$

end if

end while

return S.

drawback → It does not consider casual relationship between variables

E)

E) 1) a) Euclidean distance

b) Cosine distance.

a) It is ordinary distance between two points

$$d(p,q) = \sqrt{(q_1-p_1)^2+(q_2-p_2)^2 \cdots +(q_n-p_n)^2}$$

$$= \sqrt{\sum_{i=0}^{n} (q_i-p_i)^2}.$$

for eq.

$$D = \begin{matrix} p \\ q \end{matrix}\begin{bmatrix} 1 & 1 \\ 2 & 1 \end{bmatrix}$$

$$d(p,q) = \sqrt{(2-1)^2+(1-1)^2}$$

$$= 1.$$

b) Cosine distance

It is a measure of similarity between two non-zero vectors of an inner product space

$$A \cdot B = ||A|| ||B| \cos\theta$$

$$d(A,B) = \frac{A \cdot B}{|A| \, |B|}$$

For eg.

$$D = \begin{matrix} A \\ B \end{matrix} \begin{bmatrix} 1 & 1 \\ 2 & 1 \end{bmatrix}$$

$$= \frac{1 \cdot 2 + 1 \cdot 1}{\sqrt{1^2+1^2} \cdot \sqrt{2^2+1^2}}$$

$$= \frac{3}{\sqrt{2}\sqrt{5}}$$

$$= \frac{3}{\sqrt{10}}$$

2) $D = \begin{matrix} 1 \\ 2 \\ 3 \\ 4 \end{matrix} \begin{bmatrix} 1 & 1 \\ 4 & 3 \\ 5 & 4 \\ 2 & 1 \end{bmatrix}$

Let's choose (1, 3) as centroids, Subtract 1 from d since we are using cos

$d(1,1) = 1 \quad = 1-0 = 0$

$d(1,2) = \frac{4+3}{\sqrt{2}\sqrt{25}}$

$= \frac{7}{5\sqrt{2}} = 1-0.9 = 0.010$

~~$d(1,3) = \frac{9}{\sqrt{2}\cdot\sqrt{41}}$~~ $d(1,3) = \frac{9}{\sqrt{2}\sqrt{41}}$

$d(1,4) = \frac{3}{\sqrt{2}\sqrt{5}}$

$= \frac{3}{\sqrt{10}} = 1-0.94 = 0.05$

$d(3,1) = 1 - \frac{9}{\sqrt{2}\sqrt{41}} = 0.007$

$d(2,3) = \frac{20+12}{\sqrt{25}\sqrt{41}} = \frac{32}{5\sqrt{41}} = 0.0004$

$d(3,3) = 1 \quad = 1-1 = 0$

$$d(3,4) = \frac{14}{\sqrt{41}\sqrt{5}} = 1\text{-Ans}$$

$$= 0.022$$

New clusters

(1) , $(2, 3, 4)$

New centroids

$(1,1), \left(\frac{11}{3}, \frac{8}{3}\right)$

$C1 \rightarrow (1,1)$

$C2 \rightarrow \left(\frac{11}{3}, 8/3\right)$

$$(C1,1) = \frac{1\cdot1 + 1\cdot1}{\sqrt{2}\times\sqrt{2}}$$

$$(C1,2) = \frac{4+3}{\sqrt{25}\sqrt{2}}$$

$$(C1,3) = \frac{9}{\sqrt{41}\sqrt{2}}$$

$$(C1,4) = \frac{3}{\sqrt{5}\sqrt{2}}$$

$$(C2,1) = \frac{\frac{11}{3} + \frac{8}{3}}{\sqrt{2} \times \sqrt{\left(\frac{11}{3}\right)^2 + \left(\frac{8}{3}\right)^2}}$$

$$(C2,2) = \frac{\frac{44}{3} + 8}{\sqrt{\left(\frac{11}{3}\right)^2 + \left(\frac{8}{3}\right)^2} \times \sqrt{25}}$$

$$(C2,3) = \frac{\frac{53}{3} + \frac{32}{3}}{\sqrt{\left(\frac{11}{3}\right)^2 + \left(\frac{8}{3}\right)^2} \cdot \sqrt{41}}$$

$$(C2,4) = \frac{\frac{22}{3} + \frac{8}{3}}{\sqrt{5} \cdot \sqrt{\left(\frac{11}{3}\right)^2 + \left(\frac{8}{3}\right)^2}}$$

New clusters....

C1	(1, 4)
C2	(2, 3)

2) Taking (1, 2) as centroids in Kmeans (algo)

$d(1,1) = 1 - 1 = 0$

$d(1,2) = 0.010$

$d(1,3) = 1 - \frac{9}{\sqrt{2}\sqrt{41}} = 0.007$

$d(1,4) = 0.05$

$d(2,1) = 0.010$

$d(2,2) = 0$

$d(2,3) = 0.0001$

$d(2,4) = \frac{11}{\sqrt{5}\sqrt{5}} = 0.016$

New clusters

(1) (2, 3, 4)

centroids $(1,1)$, $\left(\frac{11}{3}, \frac{8}{3}\right)$

$(c1,1) = 0$

$(c1,2) = 0.010$

$(c1,3) = 0.01$

$(c1,4) = (0.05)$

$(c2,1) = 0.02$

$(c2,2) = 0.001$

$(c2,3) = 0.001$

$(c2,4) = 0.1$

New Clusters.

$(1,4)$ $(2,3)$

C1 $(1,4)$ → centroid $(1.5, 1)$

C2 $(2,3)$ → centroid $(3.5, 3.5)$

3) K. chooseCentroid (double [][] data, k) {

centroids;

C_1 = random.choice (data)

centroids.add (C_1)

for c in 1 to k-1

distance = 0;

for j in 1 to data.shape [0]:

p = data[j]

for k in 1 to centroid.length

td = distance (p, centroids [k])

d = min (d, td)

distance.add (d)

next_c.inden = argmax (distances)

next_c = data [next_c_inden]

centroids.ddd (next_c)

return centroids

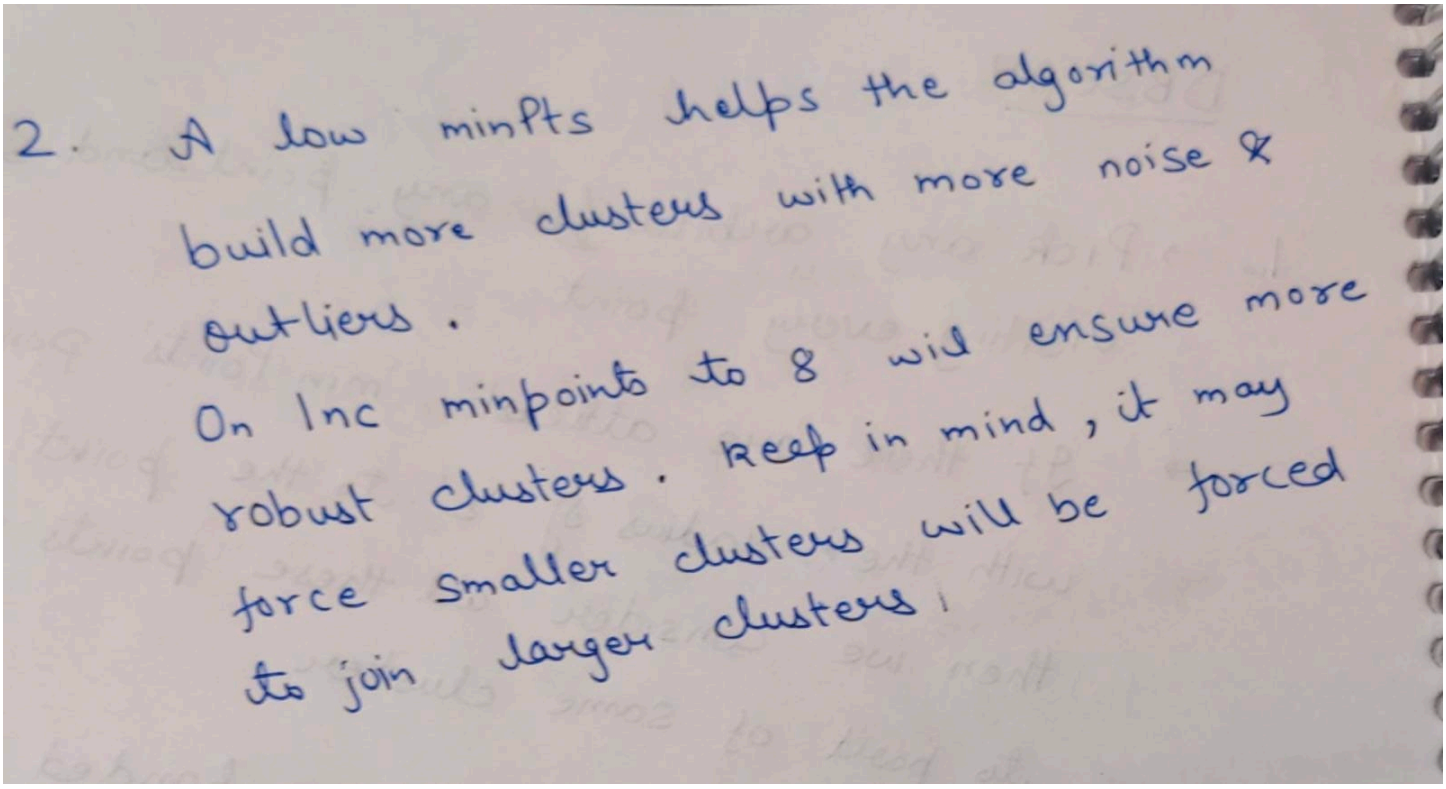
DBSCAN

1. → Pick any arbitarily any point and start visiting every point.

→ If there are atleast 'minPoints' points with the radius of 'ε' to the point then we consider all these points to part of same cluster

→ The clusters are then expanded by recursively repeating the neighborhood calculation of each neighboring point

Hierarchical clustering fails when it comes to arbitary shaped clusters or detecting outlier, DBSCAN is more efficient here



F 1) Steps to examine data:

- scatter-plots,
- correlation plots,
- density plots

Steps for preparing data for analytics:

- Deleting Rows with missing values
- Impute missing values for continuous variable
- Impute missing values for categorical variable

F2.1) Politicians want to profile voters and for that, they need data to analyze. In recent elections too, many politicians modeled their speeches and campaigns based on the results they got from PA. They mostly use clustering techniques for this.

F2.2) Data sets are used by financial managers and other professionals within finance sectors. PA is required to provide unique and timely insights into investment opportunities. Recently, a Satellite image is being used for people counting which can be used to check if the stock is hot or cold.

F3.3) Alternative data sources are increasingly seen being used for health statistics and policy analysis. However, such data sources must be evaluated carefully. Many insurance companies have started giving discounts on premiums after analyzing user-health statistics. Further body movements and facial expressions data can be useful to check if the user has started to develop any medical condition.

3. Project report

We have nearly completed data preprocessing and we are in sync with the timeline that we gave during presentation.

1. We have acquired and analyzed data from CSI market interface
2. We have almost completed analyzing and understanding the “Daily financial news for 6k+ stocks” dataset
3. We have secured data from social media websites, and we have cleaned, preprocessed and normalized our data
4. Now we are thinking of possibilities of how to use Alternative data sources to boost our accuracy and refine our model
5. Since the financial systems are prone to black swann events, we want to build built a robust a system that will take care of this.
6. Currently, we are our working on various potential data sets, assemble systems and systems and testing them through Rapid miner to get the idea of their performance metrics on our dataset before we deploy our models