CSN-505 ProjectLab

Privacy Preserving in Data Mining

Submitted By :

Aayushi Jain (20535001)

Contents

- Privacy Preserving in Data Mining(PPDM)
- Dataset
- Data Preprocessing
- k-Anonymity model
- Mondrian’s Algorithm
- Analysis after applying Mondrian’s Algorithm
- Possible attacks on k-anonymity model
- l-diversity model
- l-diversity model implementation
- Attack on l-diversity model
- Significance of implemented models
- Applications of PPDM

Privacy Preserving Data Mining(PPDM)

- The basic notion of information privacy is to have control over handling and collecting an individual's personal data. Collection of data from various sources may have many advantages, but it may also lead to information leakage.

- To deal with information leakage, methods have been proposed, which are known as **Privacy Preserving Data Mining (PPDM)** Techniques.

- PPDM techniques work by modifying user's original data. PPDM techniques are designed in such a way so as to hide the user's data, while maintain the data utility.

- I have implemented anonymization methods through properly generalizing the quasi-identifiers in the dataset in order to prevent linkage attacks and violations of privacy and security laws.

Dataset

- The dataset that I have chosen for this project is the Patient Disease dataset from Kaggle.

- This dataset contains 1338 rows of unique individuals which includes age, sex, bmi, children, smoker and disease.

- Some columns show us the quasi-identifiers of an individual, which include age, sex, bmi, children and region.

- The sensitive data would be smoker and disease that are associated with an individual because it can be used against an individual if their identity is released and not properly anonymized.

Dataset

- The dataset is shown below:

	age	sex	bmi	children	smoker	region	Disease
0	19	female	27.900	0	yes	southwest	Tumor
1	18	male	33.770	1	no	southeast	FLU
2	28	male	33.000	3	no	southeast	Stomach Inflammation
3	33	male	22.705	0	no	northwest	Bronchial Inflammation
4	32	male	28.880	0	no	northwest	FLU
...	...	...	...	...	...	...	...
1333	50	male	30.970	3	no	northwest	Hepatitis
1334	18	female	31.920	0	no	northeast	Cancer
1335	18	female	36.850	0	no	southeast	Cancer
1336	21	female	25.800	0	no	southwest	Cancer
1337	61	female	29.070	0	yes	northwest	Heart Disease

1338 rows × 7 columns

Data Preprocessing

- **Missing Values Handling:** Missing values were deleted, because there were very few missing values. There were around 5 to 6 missing values which were only for the column named "Disease".

- **Categorical variables Handling:**

  1. To make sure available data would be usable for machine learning models, I decided to map categorical variables to numerical values.

  For Eg: Gender contains male and female. Male is mapped to 0 and Female to 1.

  If a person is smoker then it is mapped to 1 and if not then mapped to 0.

- After all of these steps our dataset was ready to be worked on.

k-Anonymity Model

- k-anonymity model is a model which comes under data publishing privacy.

- **k-Anonymous :** If the identifier attributes of a record cannot be discriminated from k-1 records at the least, the dataset is said to be k-anonymous, i.e., any record in a dataset is similar to at least k other records.

- Using k-anonymity, it becomes difficult for a person to identify a person's sensitive attribute because any record is similar to k-1 other records.

- **Mondrian's Algorithm :** I have implemented Mondrian's algorithm to implement this.The algorithm utilizes a greedy search algorithm that allows for more desirable anonymizations than traditional exhaustive optimal algorithms.

Mondrian’s Algorithm

- It allows for multidimensional models, which is what’s best for our specific dataset.

- I have worked upon 3 different values of k viz. 5, 20 and 45.

- For different values of k, I’ll be showing the following data from next slide onwards:

  1. The partitions that I got after applying partitioning method which is followed by spanning and splitting method.

  2. Partitions’ coordinates are shown below. These are in the form of {(xl,yl),(xr,yr)} where x coordinates are for age and y are for bmi.

  3. Graph contains the partitions for age versus bmi.

  4. Final anonymized output for the dataset .

Mondrian's Algorithm

- **<u>For k =5</u>**

1.

```
[467]: finished_partitions
```

```
[467]: [Int64Index([35, 172, 232, 410, 681, 972, 1027, 1129, 1251], dtype='int64'),
        Int64Index([359, 362, 584, 747, 1033, 1212, 1231, 1282, 1316], dtype='int64'),
        Int64Index([121, 157, 295, 492, 940, 1041], dtype='int64'),
        Int64Index([0, 31, 133, 236, 296, 349, 529, 791, 1296], dtype='int64'),
        Int64Index([126, 134, 293, 375, 487, 604, 648, 875, 1163], dtype='int64'),
        Int64Index([37, 219, 388, 714, 821, 989, 1137], dtype='int64'),
        Int64Index([192, 452, 508, 579, 693, 816, 857, 1002], dtype='int64'),
        Int64Index([10, 40, 274, 340, 476, 548, 631, 1305], dtype='int64'),
        Int64Index([28, 248, 428, 593, 680, 802, 990, 1295, 1302], dtype='int64'),
        Int64Index([15, 326, 468, 469, 636, 897, 1016, 1023, 1048], dtype='int64'),
        Int64Index([105, 106, 149, 261, 270, 1026, 1170, 1299, 1315], dtype='int64'),
        Int64Index([76, 117, 249, 282, 291, 434, 465, 784, 952], dtype='int64'),
        Int64Index([104, 504, 535, 585, 827, 885, 954, 993, 1075, 1081, 1273], dtype='int64'),
        Int64Index([182, 250, 276, 364, 507, 586, 926, 1080, 1242], dtype='int64'),
        Int64Index([102, 471, 482, 808, 822, 1150], dtype='int64'),
        Int64Index([195, 397, 503, 513, 565, 581, 840, 1042, 1072, 1158, 1196], dtype='int64'),
        Int64Index([200, 259, 490, 614, 815, 911, 1139, 1147, 1334], dtype='int64'),
        Int64Index([374, 430, 525, 612, 618, 623, 663, 1244, 1268], dtype='int64'),
        Int64Index([423, 1093, 1181, 1182, 1235, 1250, 1267, 1276, 1308], dtype='int64'),
        Int64Index([22, 136, 194, 223, 385, 700, 1025, 1291], dtype='int64'),
```

```
[468]: print(len(finished_partitions))
```

```
198
```

Mondrian's Algorithm

- **<u>For k =5</u>**

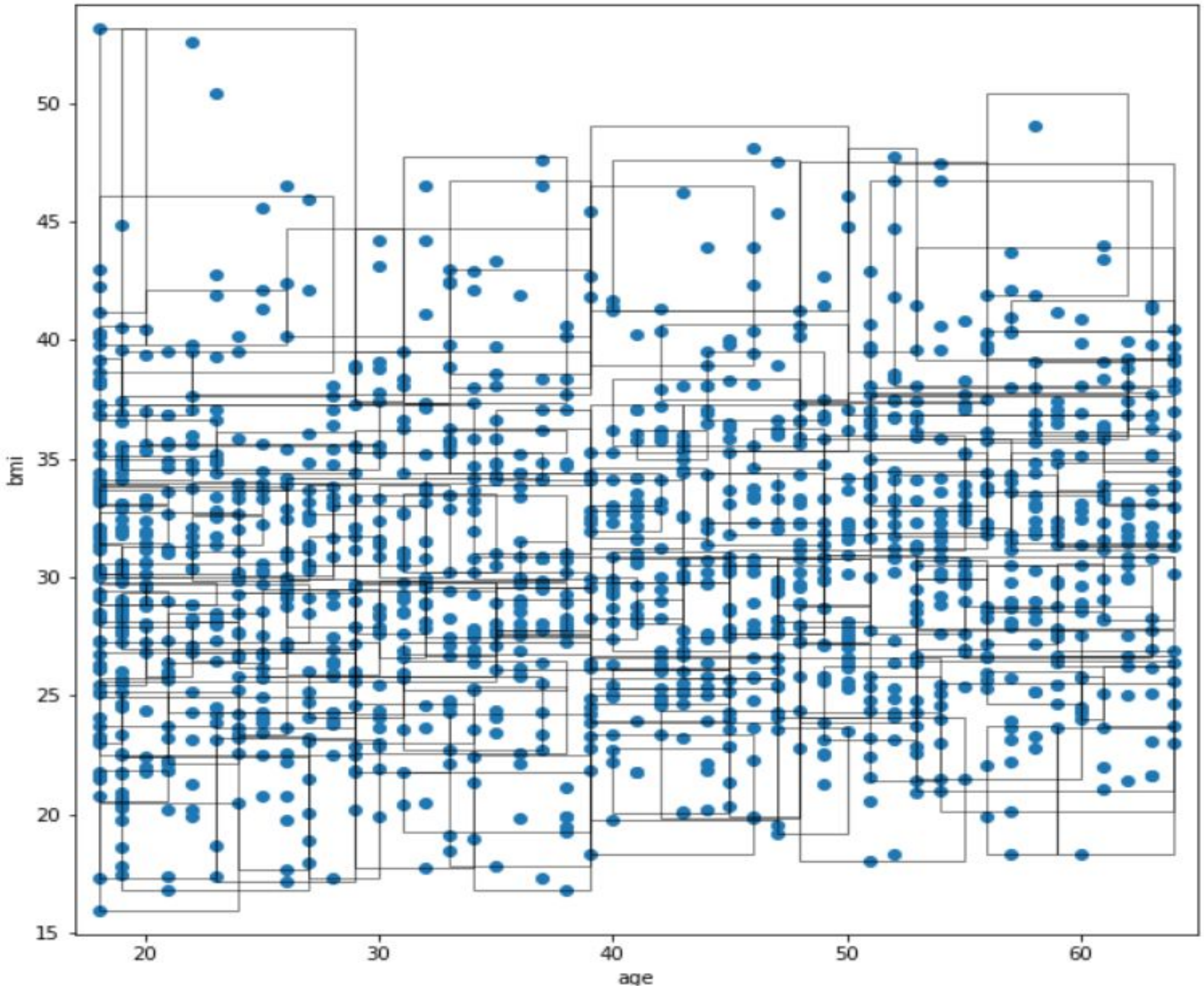
<u>2.</u>

```
472]: rects

472]: [((18.0, 15.96), (24.0, 20.52)),
        ((18.0, 20.6), (21.0, 21.85)),
        ((18.0, 22.99), (19.0, 25.2)),
        ((18.0, 25.46), (20.0, 27.93)),
        ((18.0, 28.12), (24.0, 28.88)),
        ((24.0, 17.67), (27.0, 23.275)),
        ((24.0, 23.4), (26.0, 25.84)),
        ((24.0, 26.22), (26.0, 29.355)),
        ((19.0, 16.815), (27.0, 22.42)),
        ((18.0, 22.515), (29.0, 25.6)),
        ((18.0, 25.745), (21.0, 29.4)),
        ((29.0, 27.645), (32.0, 29.7)),
        ((32.0, 27.5), (39.0, 29.6)),
        ((18.0, 17.29), (23.0, 23.76)),
        ((18.0, 30.115), (19.0, 31.4)),
        ((19.0, 30.02), (22.0, 31.3)),
        ((18.0, 31.46), (21.0, 33.06)),
        ((18.0, 33.1), (21.0, 33.915)),
        ((22.0, 29.83), (27.0, 31.1)),
        ((18.0, 34.1), (22.0, 34.96)),
```

Mondrian's Algorithm

3.



Mondrian's Algorithm

- **For k =5**

4.

```
]: dfn
```

]:

	age	bmi	children	Disease
0	19.666667	18.812778	0	FLU
1	19.666667	18.812778	0	Bronchial Inflammation
2	18.666667	21.337778	0	FLU
3	18.666667	21.337778	0	Hepatitis
4	18.666667	21.337778	0	Bronchial Inflammation
...	...	...	...	...
490	61.666667	32.369167	2	Tumor
491	61.666667	32.369167	2	Bronchitus
492	60.571429	32.701429	3	Hepatitis

Mondrian's Algorithm

- **For k =20**

1.

```
[506]: finished_partitions

[506]: [Int64Index([  17,   35,   64,  121,  137,  157,  172,  232,  277,  295,  311,
              359,  362,  410,  464,  492,  584,  681,  747,  792,  882,  899,
              940,  943,  972, 1027, 1033, 1041, 1114, 1129, 1212, 1223, 1231,
             1251, 1282, 1292, 1316],
            dtype='int64'),
 Int64Index([   0,   31,   65,  122,  126,  133,  134,  135,  236,  238,  293,
              296,  349,  375,  427,  453,  472,  487,  495,  529,  576,  604,
              648,  690,  751,  773,  791,  804,  855,  875, 1038, 1077, 1163,
             1175, 1189, 1252, 1296, 1336],
            dtype='int64'),
 Int64Index([   3,    5,   37,   70,   99,  101,  192,  217,  219,  388,  404,
              406,  452,  508,  579,  606,  693,  714,  799,  816,  821,  831,
              848,  857,  863,  909,  971,  975,  981,  989, 1002, 1043, 1054,
             1137, 1194, 1260, 1286, 1306],
            dtype='int64'),
 Int64Index([   4,   10,   40,  108,  125,  164,  191,  274,  324,  340,  352,
              439,  476,  548,  551,  570,  625,  631,  672,  709,  741,  743,
              750,  763,  795,  961,  999, 1014, 1032, 1040, 1104, 1165, 1179,
             1254, 1274, 1277, 1305, 1311],
            dtype='int64')

[507]: print(len(finished_partitions))

44
```

Mondrian's Algorithm

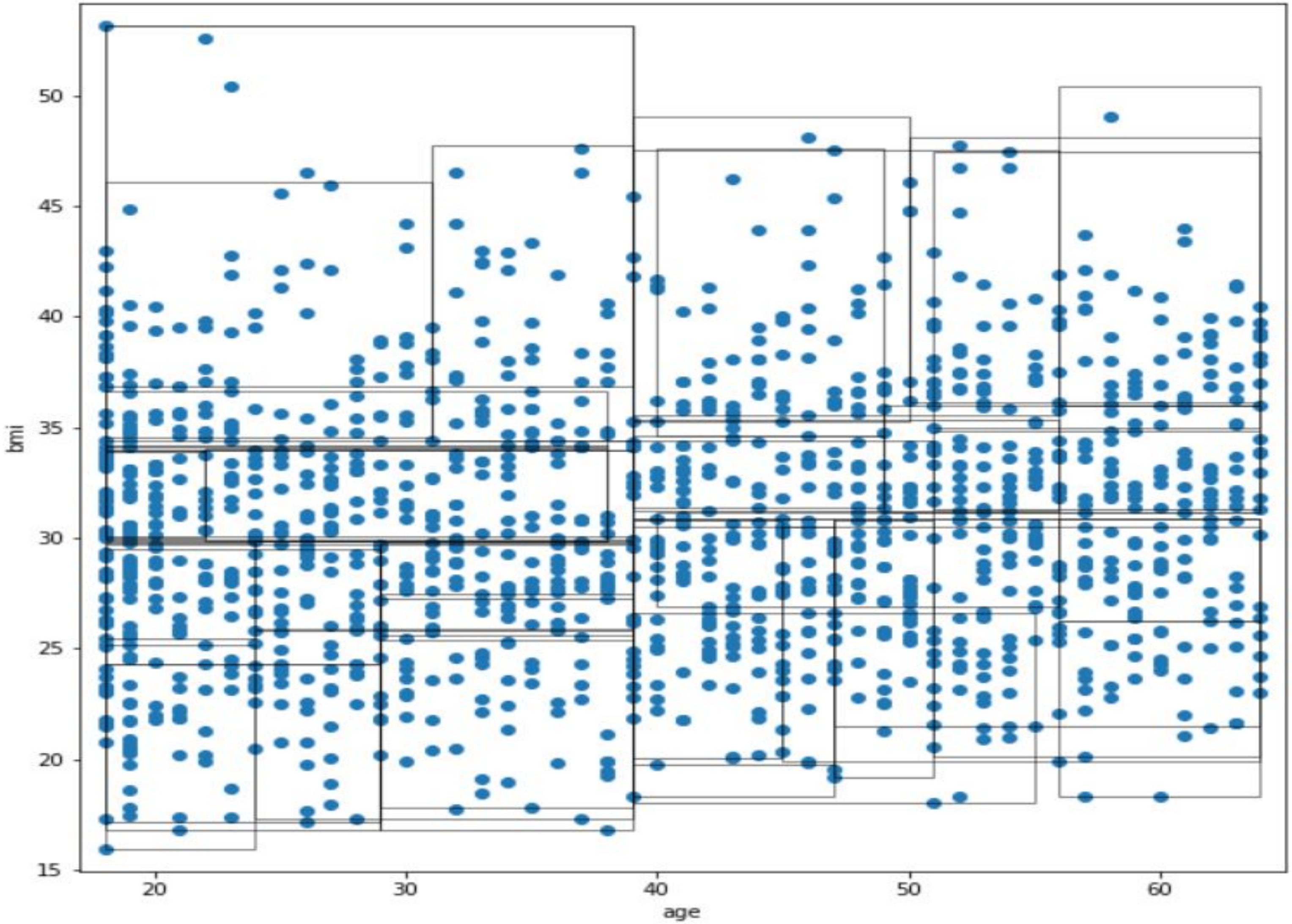
- **<u>For k =20</u>**

<u>2.</u>

```
11]: rects
11]: [((18.0, 15.96), (24.0, 25.2)),
      ((18.0, 25.46), (24.0, 29.81)),
      ((24.0, 17.29), (39.0, 25.85)),
      ((24.0, 25.9), (39.0, 29.83)),
      ((18.0, 16.815), (29.0, 29.5)),
      ((18.0, 30.02), (22.0, 33.915)),
      ((22.0, 29.83), (38.0, 34.01)),
      ((18.0, 34.1), (38.0, 36.6)),
      ((18.0, 36.85), (39.0, 53.13)),
      ((18.0, 29.92), (39.0, 33.99)),
      ((18.0, 33.99), (39.0, 53.13)),
      ((39.0, 18.05), (55.0, 26.62)),
      ((40.0, 26.885), (56.0, 31.16)),
      ((56.0, 18.335), (64.0, 26.29)),
      ((56.0, 26.29), (64.0, 30.875)),
      ((39.0, 31.2), (56.0, 35.31)),
      ((39.0, 35.53), (56.0, 47.52)),
      ((56.0, 31.16), (64.0, 36.0)),
      ((56.0, 36.005), (64.0, 50.38)),
      ((39.0, 19.8), (47.0, 30.5)),
      ((29.0, 17.86), (39.0, 27.28)),
      ((29.0, 27.5), (39.0, 29.7)),
      ((18.0, 17.195), (29.0, 24.3)),
      ((18.0, 24.3), (29.0, 29.735)),
```

Mondrian's Algorithm

3.



Mondrian's Algorithm

- **<u>For k =20</u>**

<u>4.</u>

```
5]: dfn
```

5]:

	age	bmi	children	Disease
0	19.540541	21.968649	0	FLU
1	19.540541	21.968649	0	Hepatitis
2	19.540541	21.968649	0	Tumor
3	19.540541	21.968649	0	Bronchial Inflammation
4	19.540541	21.968649	0	Heart Disease
...	...	...	...	...
211	55.250000	38.765781	2	Tumor
212	55.250000	38.765781	2	Bronchial Inflammation
213	55.250000	38.765781	2	Heart Disease
214	55.250000	38.765781	2	Stomach Inflammation

Mondrian's Algorithm

- **For k =45**

1.

```
532]: finished_partitions
```

```
532]: [Int64Index([   0,   17,   31,   35,   64,   65,  121,  122,  126,  133,  134,
                    135,  137,  157,  172,  232,  236,  238,  277,  293,  295,  296,
                    311,  349,  359,  362,  375,  410,  427,  453,  464,  472,  487,
                    492,  495,  529,  576,  584,  604,  648,  681,  690,  747,  751,
                    773,  791,  792,  804,  855,  875,  882,  899,  940,  943,  972,
                   1027, 1033, 1038, 1041, 1077, 1114, 1129, 1163, 1175, 1189, 1212,
                   1223, 1231, 1251, 1252, 1282, 1292, 1296, 1316, 1336],
                  dtype='int64'),
        Int64Index([   3,    4,    5,   10,   37,   40,   70,   99,  101,  108,  125,
                    164,  191,  192,  217,  219,  274,  324,  340,  352,  388,  404,
                    406,  439,  452,  476,  508,  548,  551,  570,  579,  606,  625,
                    631,  672,  693,  709,  714,  741,  743,  750,  763,  795,  799,
                    816,  821,  831,  848,  857,  863,  909,  961,  971,  975,  981,
                    989,  999, 1002, 1014, 1032, 1040, 1043, 1054, 1104, 1137, 1165,
                   1179, 1194, 1254, 1260, 1274, 1277, 1286, 1305, 1306, 1311],
                  dtype='int64'),
        Int64Index([  15,   28,   63,   76,   80,  104,  105,  106,  117,  149,  150,
                    205,  213,  241,  248,  249,  261,  270,  282,  291,  320,  326,
                    363,  426,  428,  434,  451,  465,  468,  469,  504,  535,  583,
                    585,  593,  636,  680,  684,  703,  727,  762,  784,  797,  802
```

```
533]: print(len(finished_partitions))
```

```
21
```

Mondrian's Algorithm

- **<u>For k =45</u>**

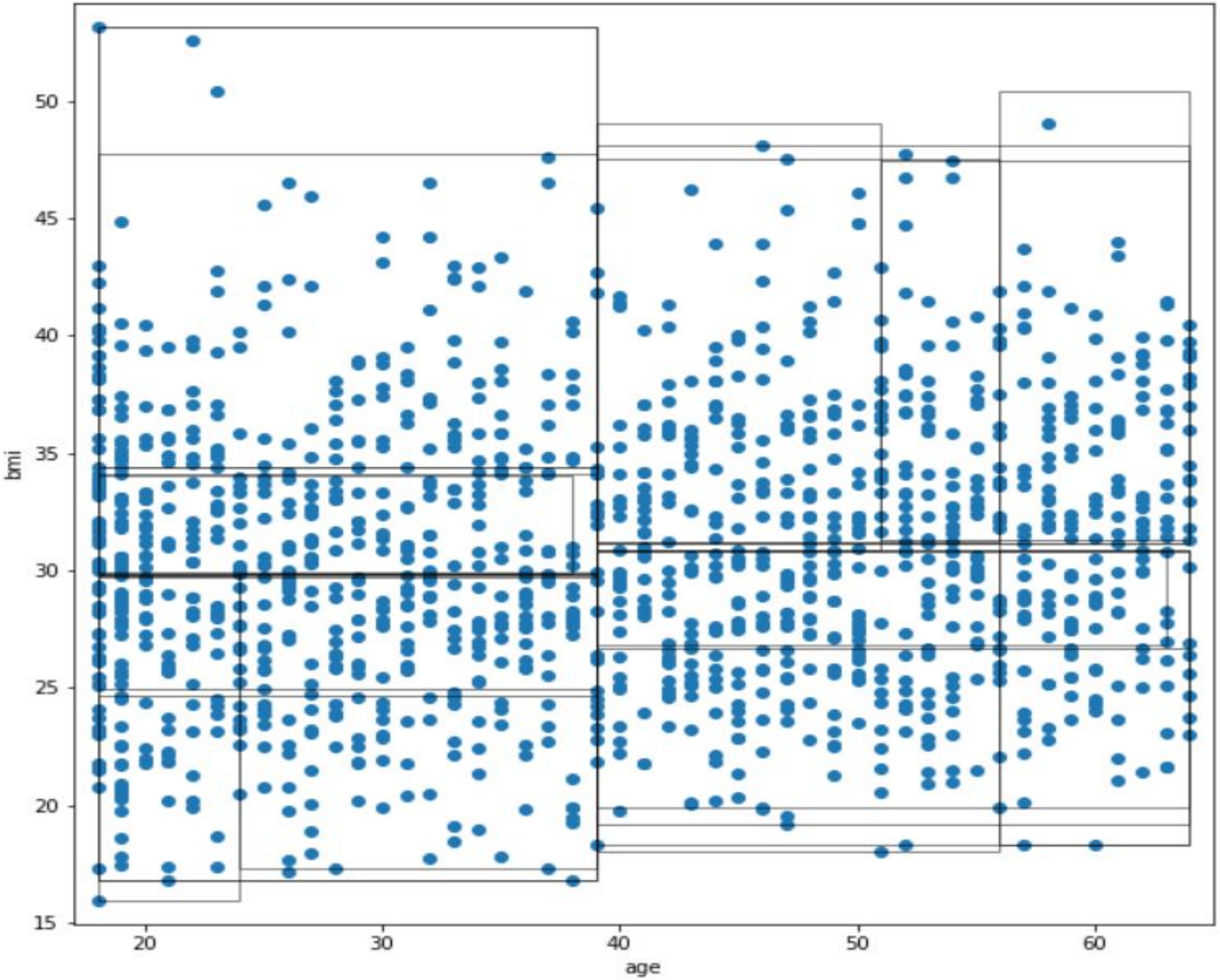
2.

```
7]: rects

7]: [((18.0, 15.96), (24.0, 29.81)),
     ((24.0, 17.29), (39.0, 29.83)),
     ((18.0, 16.815), (39.0, 29.7)),
     ((18.0, 29.83), (38.0, 34.01)),
     ((18.0, 34.1), (39.0, 53.13)),
     ((18.0, 29.92), (39.0, 53.13)),
     ((39.0, 18.05), (56.0, 31.16)),
     ((56.0, 18.335), (64.0, 30.875)),
     ((39.0, 31.2), (56.0, 47.52)),
     ((56.0, 31.16), (64.0, 50.38)),
     ((39.0, 31.16), (64.0, 48.07)),
     ((18.0, 16.815), (39.0, 24.64)),
     ((18.0, 24.985), (39.0, 29.81)),
     ((18.0, 29.83), (39.0, 34.39)),
     ((18.0, 34.39), (39.0, 47.74)),
     ((39.0, 19.19), (64.0, 26.7)),
     ((39.0, 26.8), (63.0, 30.875)),
     ((39.0, 19.95), (64.0, 30.78)),
     ((39.0, 18.3), (64.0, 30.9)),
     ((39.0, 30.9), (51.0, 49.06)),
     ((51.0, 31.3), (64.0, 47.41))]
```

Mondrian's Algorithm

3.



Mondrian's Algorithm

- **For k =45**

4.

```
[4]: dfn
```

[4]:

	age	bmi	children	Disease
0	19.773333	24.973600	0	FLU
1	19.773333	24.973600	0	Cancer
2	19.773333	24.973600	0	Hepatitis
3	19.773333	24.973600	0	Tumor
4	19.773333	24.973600	0	Bronchial Inflammation
...	...	...	...	...
127	56.158730	35.837222	2	Bronchial Inflammation
128	56.158730	35.837222	2	Heart Disease
129	56.158730	35.837222	2	Stomach Inflammation
130	56.158730	35.837222	2	Bronchitus

Analysis after applying Mondrian's Algorithm

- The Mondrian algorithm generalized our dataset over the calculated partitions with k-anonymity of k = 5, k = 20 , k = 45 .

- Our dataset lost information variables [‘sex’, ‘smoker’, ‘region’] during the anonymization.

- Some of the columns in the dataset such as age and bmi were generalized to be the mean value of their partition. This helped with making the entries indistinguishable and not as easily recognizable to an adversary.

- Additionally, the sensitive data value, charges, was given a different value in comparison to the original data, in that the numbers were swapped and unordered so that an adversary could not fully comprehend the true values and link it to the individual.

Possible attacks on k-Anonymity Model

- **Homogeneity Attack**: This attack leverages the case where all the values for a sensitive value within a set of *k* records are identical.In such cases, even though the data has been *k*-anonymized, the sensitive value for the set of *k* records may be exactly predicted.
- **Background Knowledge Attack:** This attack leverages an association between one or more quasi-identifier attributes with the sensitive attribute to reduce the set of possible values for the sensitive attribute. Common known facts or background knowledge can de-anonymize the identity of a person.

L-Diversity Model

- Because of the limitations of the k-anonymity model, the l-diversity model was proposed. The l-diversity model is an expansion of the k-anonymity model, in the sense that it follows the l-diversity principle in each equivalence class.

- **l-Diversity Principle:** The l-diversity principle states that "in each equivalence class, at least l 'well represented values' exist for the sensitive attributes." A dataset is said to be l-diverse, if all the equivalence classes follow the property of l-diversity.

- **Implementation of l-Diversity Model:**

1. I have chosen l=5.

2. I have shown implementation for k = 5, 20 and 45 for l-diversity model as well.

l-Diversity Model Implementation

- Number of partitions that I got after applying partitioning method are given below:

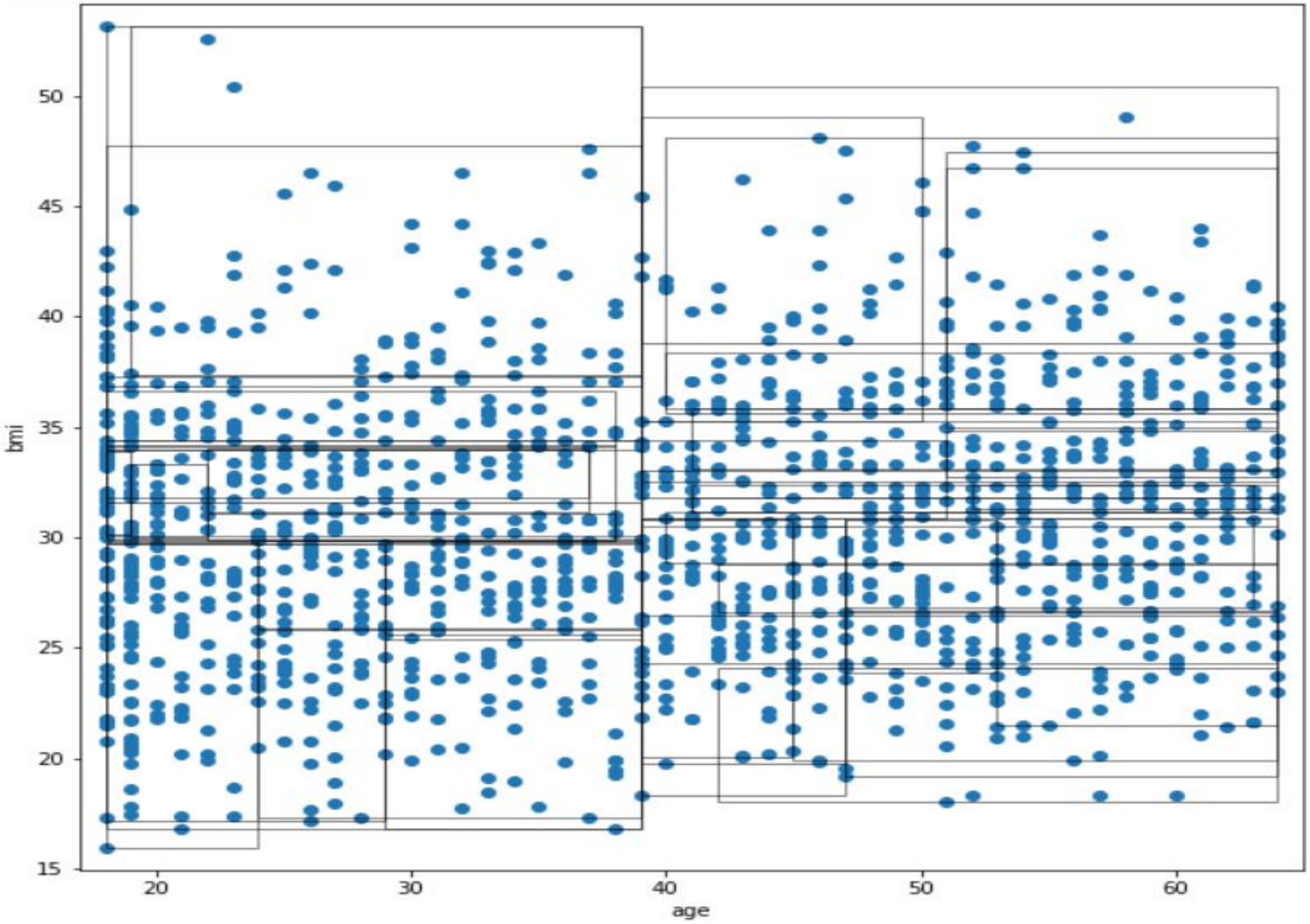
1. **k=5 :** 46 partitions
2. **k=20:** 38 partitions
3. **k=45:** 21 partitions

- Graph for the partitions for different values of k for age versus bmi is shown in next slides.

- Final anonymized dataset for different values of k are also shown on next slide.

l-Diversity Model Implementation

- **For k =5:**



l-Diversity Model Implementation

- **<u>For k =5:</u>**

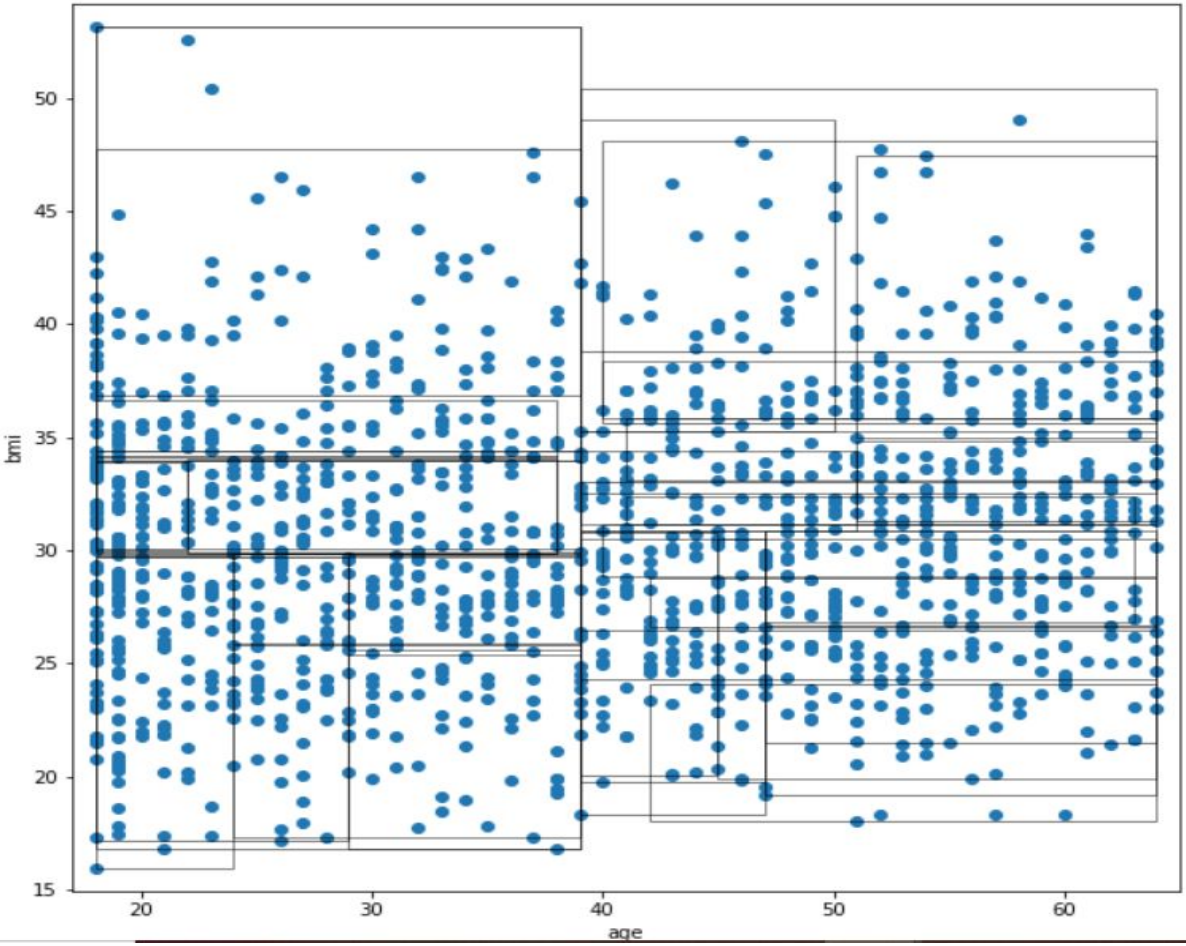
)3]:

	age	bmi	children	Disease
103	18.000000	31.874231	0	Bronchial Inflammation
101	18.000000	31.874231	0	Cancer
100	18.000000	31.874231	0	FLU
102	18.000000	31.874231	0	Hepatitis
104	18.000000	31.874231	0	Stomach Inflammation
...	...	...	...	...
213	56.705882	27.142353	3	Tumor
220	58.000000	31.954000	2	AIDS
219	58.000000	31.954000	2	Heart Disease
217	58.000000	31.954000	2	Hepatitis

l-Diversity Model Implementation

- **For k =20:**

```
pl.show()
```



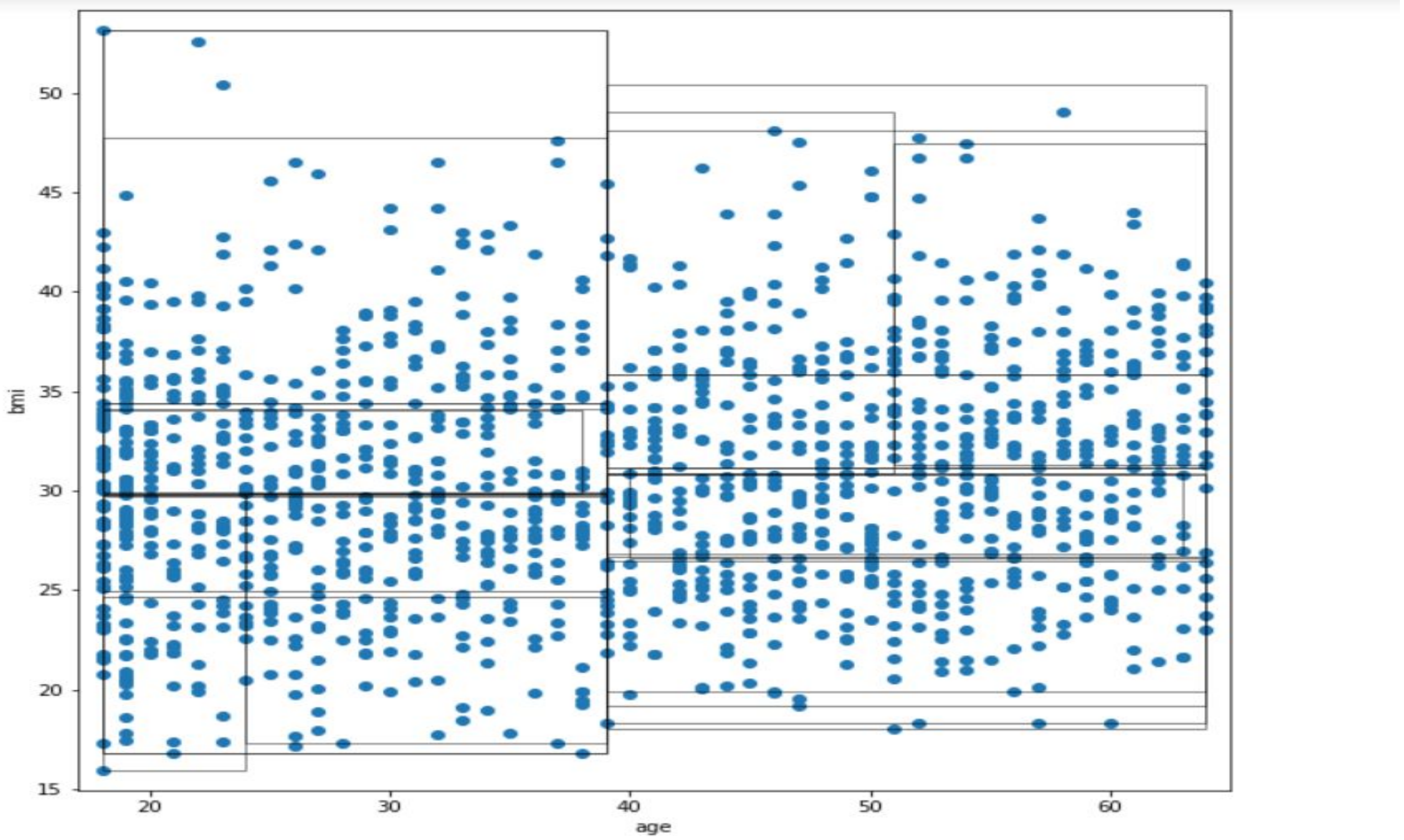
l-Diversity Model Implementation

- **For k =20:**

	age	bmi	children	Disease
0	19.773333	24.973600	0	FLU
1	19.773333	24.973600	0	Cancer
2	19.773333	24.973600	0	Hepatitis
3	19.773333	24.973600	0	Tumor
4	19.773333	24.973600	0	Bronchial Inflammation
...	...	...	...	...
203	55.250000	38.765781	2	Tumor
204	55.250000	38.765781	2	Bronchial Inflammation
205	55.250000	38.765781	2	Heart Disease
206	55.250000	38.765781	2	Stomach Inflammation

l-Diversity Model Implementation

- **For k =45:**



l-Diversity Model Implementation

- **<u>For k =45:</u>**

```
6]: dfl
```

6]:

	age	bmi	children	Disease
0	19.773333	24.973600	0	FLU
1	19.773333	24.973600	0	Cancer
2	19.773333	24.973600	0	Hepatitis
3	19.773333	24.973600	0	Tumor
4	19.773333	24.973600	0	Bronchial Inflammation
...	...	...	...	...
130	56.158730	35.837222	2	Bronchial Inflammation
131	56.158730	35.837222	2	Heart Disease
132	56.158730	35.837222	2	Stomach Inflammation
133	56.158730	35.837222	2	Bronchitus

Attack on l-Diversity Model

- **<u>Similarity attack :</u>** When the sensitive attribute values in an equivalence class are distinct but semantically similar, an adversary can learn important information. There may be occurrence leakage of sensitive information because while l-diversity requirement ensures "diversity" of sensitive values in each group, it does not take into account the semantical closeness of these values. This attack can be overcome using t-closeness model. For Eg; in our dataset, we have stomach inflammation and bronchial inflammation.

567]:

	age	sex	bmi	children	smoker	region	Disease
0	19	female	27.900	0	yes	southwest	Tumor
1	18	male	33.770	1	no	southeast	FLU
2	28	male	33.000	3	no	southeast	Stomach Inflammation
3	33	male	22.705	0	no	northwest	Bronchial Inflammation
4	32	male	28.880	0	no	northwest	FLU
...	...	...	...	...	...	...	...

Significance of the implemented models

- Since this dataset pertains to patient disease data records of individuals, If this data was to be leaked, many individual’s personal information is at risk. This can lead to reidentification of the person by a linkage attack. This is significant because the dataset includes sensitive data such as the individual’s number of children, if they are a smoker, and the disease with which they are suffering from.

- Another reason this problem is significant is because if the data is leaked, legal and governmental issues such as HIPAA violations may occur because the data was not properly anonymized.

- There have been countless medical data leaks in the past and they have caused numerous individuals’ private information

Applications of PPDM

- **PPDM in Cloud:** Cloud is a distributed infrastructure with great storage and computation capabilities that is accessible through the network, anytime and anywhere. Therefore, applications (or services) that collect, store and analyse large data quantities often require the cloud.

- **PPDM in E-Health:**Health records are considered to be extremely private, as much of this data is considered sensitive. However, the increase in the amount of data,combined with the favourable properties of the cloud has led health services to store and exchange medical records through this infrastructure .

- **PPDM in location based services:** Technologies such as GPS have a gained a great importance in recent times, as they allow to gain highly accurate location information. The location information can be used to keep a track of a user's activities.

Thank You