Mining the "Diabetes in 130 US Hospitals for Years 1999–2008" Dataset

Final Project in Data Mining

Dario Gjorgjevski[1]
gjorgjevski.dario@students.finki.ukim.mk

[1] Faculty of Computer Science and Engineering
Ss. Cyril and Methodius University in Skopje

September 22, 2016

Outline

An overview of the data

- The dataset is called “Diabetes in 130 US Hospitals for Years 1999–2008”
- It is available in the UCI repository
- Over 10 years (1999–2008) of clinical care at 130 hospitals
- 50 attributes representing patient and hospital outcomes
- Included are attributes indicating numbers and types of procedures, diagnoses, medications, etc.

What tasks have been performed

The tasks that I have performed include:

1. Preprocessing and visualization
2. Classification using decision trees, naïve Bayes, and k-nearest neighbors
3. Clustering using DBSCAN and k-means
4. Association rule mining using the apriori algorithm

The goal is to predict whether a patient will be readmitted within 30 days or not.

Potential pitfalls

The class distribution is very imbalanced:

Other	Readmitted
61 979	6559

Additionally, the dimensionality of the data is very high, exceeding 50.

Outline

Reading the data

- We read the data from the provided CSV file and remove any unique ID's

```
df <- read.csv("diabetic_data.csv", na.strings="?")
df <- subset(df,
             select=-c(encounter_id, patient_nbr))
```

- Afterwards, the data is sanitized so that nominal attributes are represented correctly within R

Imputation of missing values

- The dataset contains attributes with missing values that need to be imputed
- Some attributes (weight, medical specialty, and payer code) are missing in many observations (over 40 %), so we drop them entirely
- Other attributes are imputed according to the following strategy:
  - Numeric attributes are imputed by their means
  - Nominal attributes are imputed by sampling with replacement from the available data

Grouping of nominal attributes

In order to reduce complexity, nominal attributes with many distinct values are grouped together while preserving the distribution of the `readmitted` attribute. The following attributes are grouped:

- Similar diagnoses
- Similar reasons for admission and discharge
- Age groups showing similar characteristics

Outlier removal

Boxplots of numeric attributes reveal the presence of outliers.

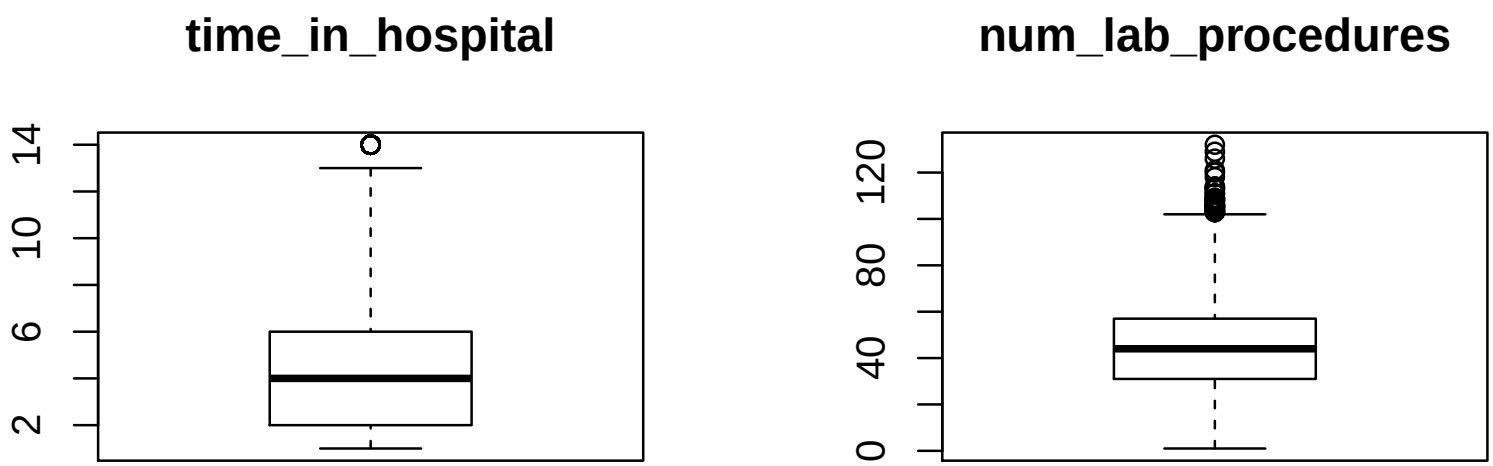


Figure 1: Boxplots showing outliers

Outlier removal

I will consider an outlier everything outside of the

$$[Q1 - 1.75 \cdot IQR, Q3 + 1.75 \cdot IQR] \tag{1}$$

range, where Q1 is the first, Q3 the third quartile, and IQR and inter-quartile range. Often authors use 1.5 instead of 1.75, but I have decided that 1.5 is too strict.

Attribute cleaning and transformation

- Attributes with near-zero variance are dropped from the data, as they provide no meaningful information
- There are not any highly correlated attributes
- Numeric attributes are centered, scaled, and have a Box–Cox transformation applied to them in order to correct for skewed distributions
- PCA analysis shows that all components need to be retained in order to capture a significant portion of the variance

Correlation analysis

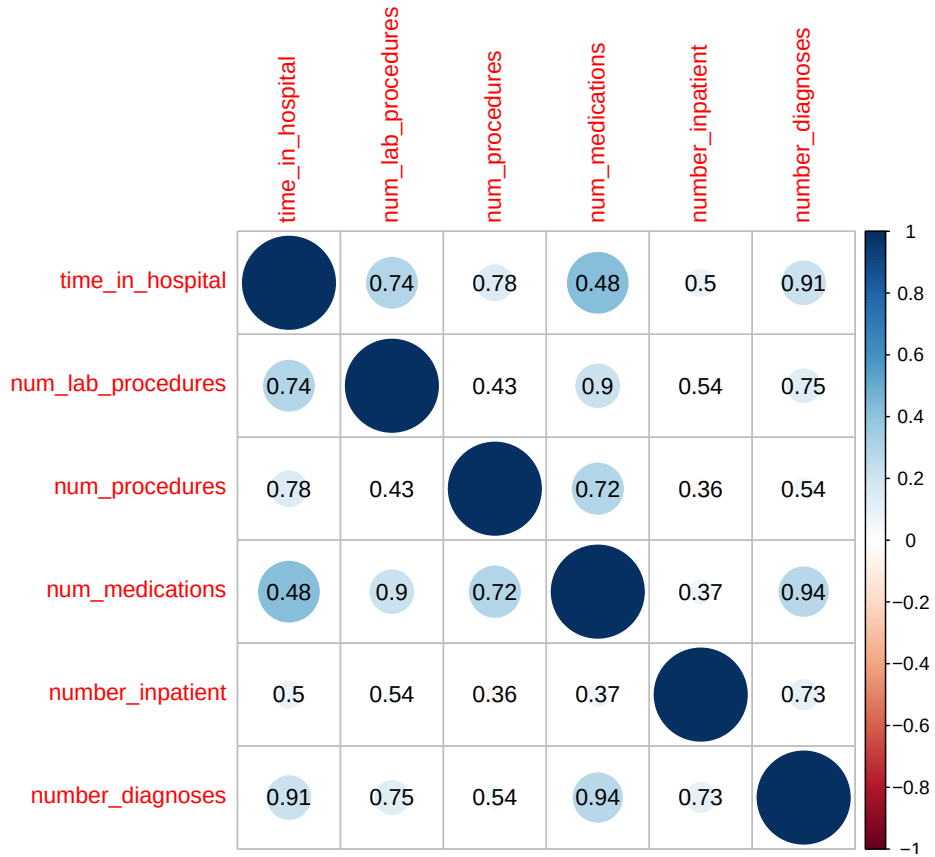


Figure 2: Correlation matrix with p-values

Visualizing medical care by race

Interestingly, patients receive the same amount of care regardless of race (the densities shown below are uniform).

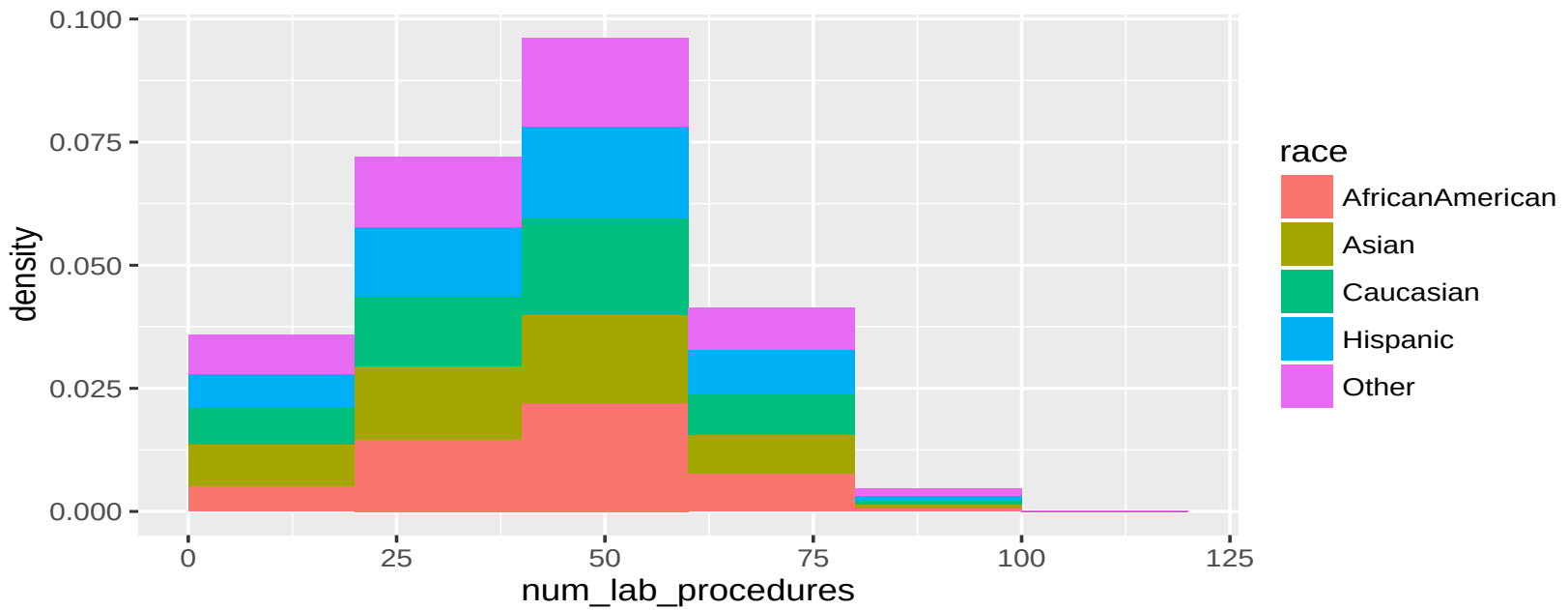


Figure 3: Number of lab procedures by race

Number of diagnoses for Caucasian patients

... however, Caucasian patients receive at the median more diagnoses.

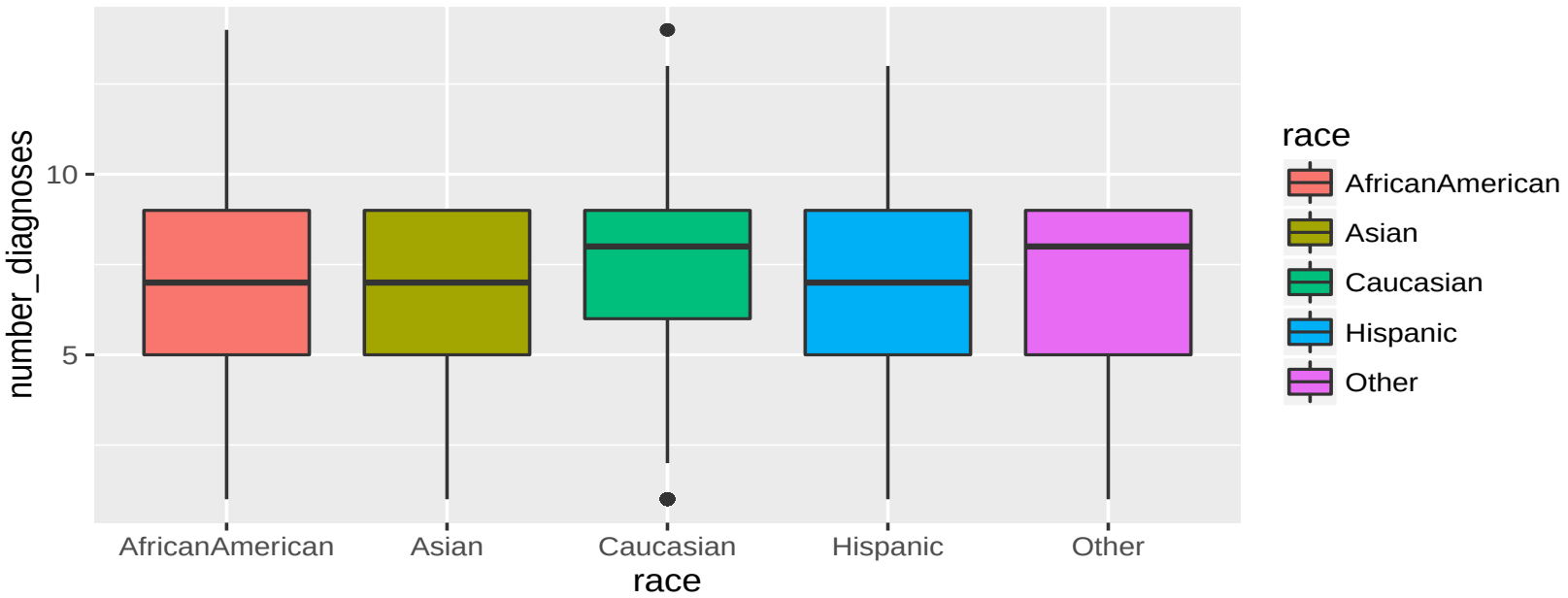


Figure 4: Number of diagnoses by race

Outline

How classification will be performed

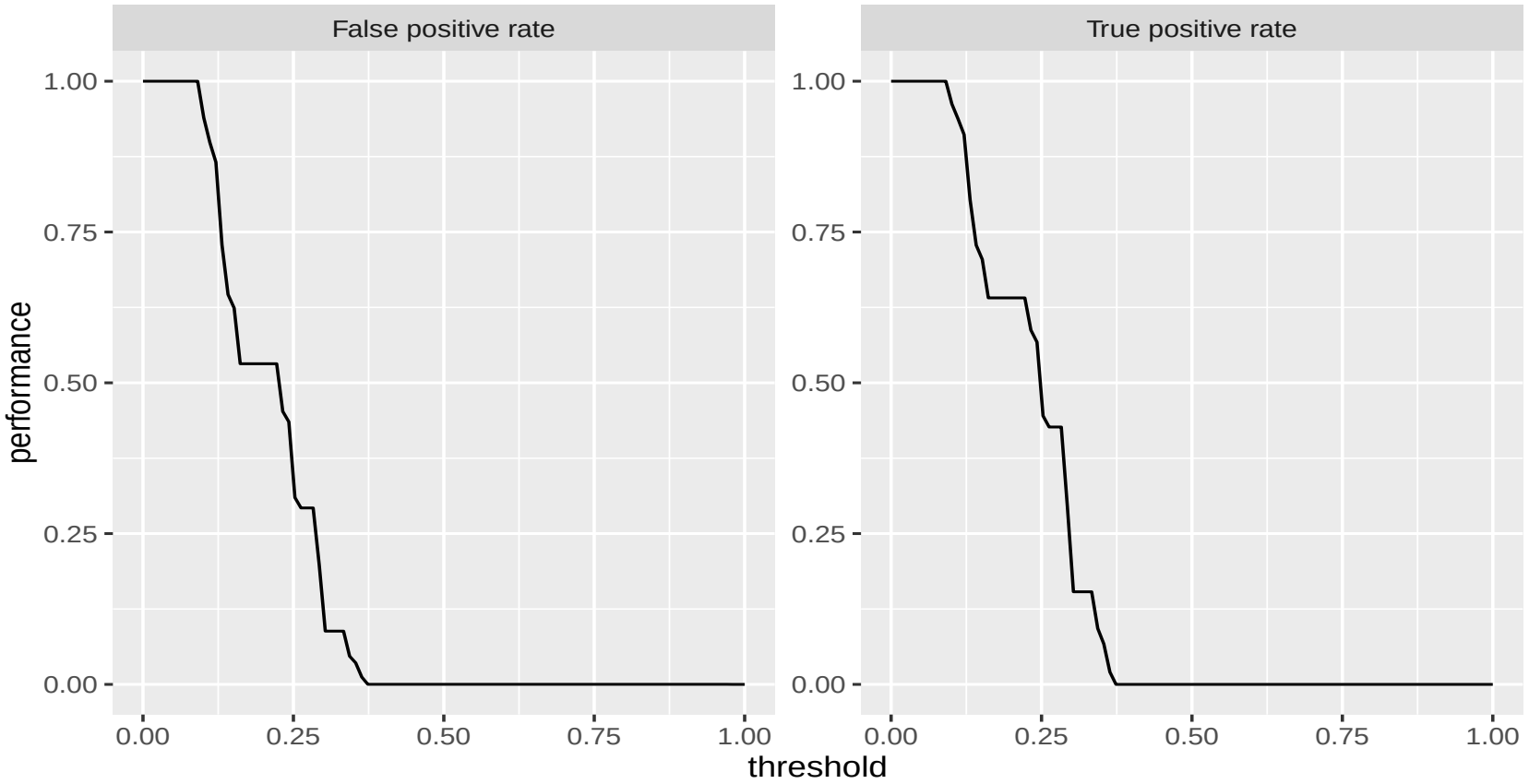
- Due to the class imbalance, SMOTE will be used to oversamlpe the minority class
- Whenever possible, classification algorithms will be set to learn probabilities rather than classes
- This allows thresholding to be used
- Thresholds will be optimized w.r.t. a self-defined "balanced measure" (denoted bm), given by

$$\mathrm{bm} = \frac{\mathrm{accuracy} + 2 \cdot \mathrm{F}_1}{3}. \tag{2}$$

Decision tree using the CART algorithm

- Using a default threshold of 0.5, 5-fold stratified cross-validation results in 90 % accuracy, 0.592 area under ROC, and a disappointing F_1 score of almost 0
- Thresholding increases the F_1 score to a solid 0.2, but decreases accuracy to 72 %
- ROC analysis shows flat regions and spikes indicating that the CART algorithm is very sensitive to threshold changes

Threshold dependence curves for CART



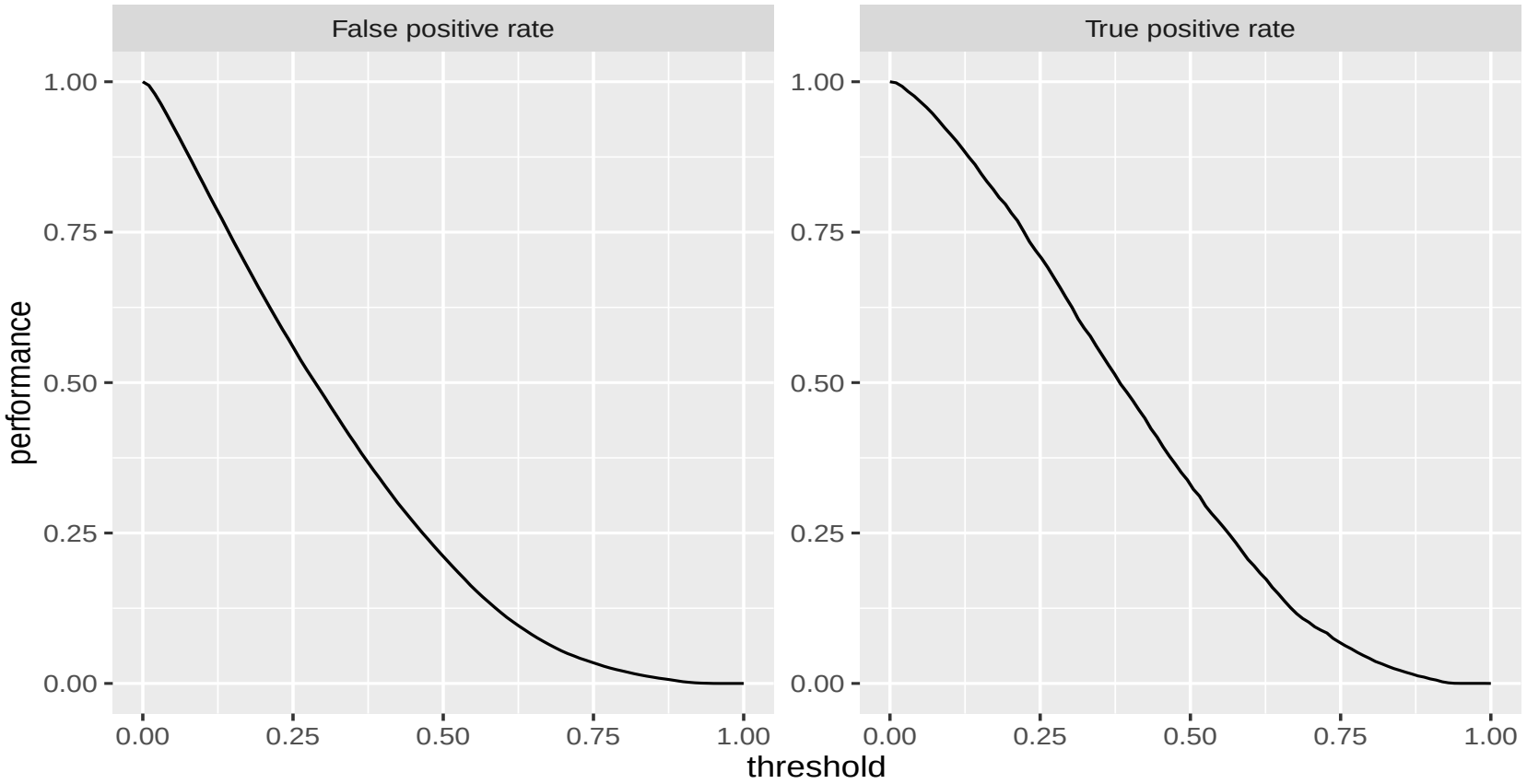
Decision tree using the J48 algorithm

- Using a default threshold of 0.5, 5-fold stratified cross-validation results in 89.6 % accuracy, 0.578 area under ROC, and a disappointing F_1 score of 0.03
- Thresholding increases the F_1 score to 0.154, but decreases accuracy to 81 %
- t-test indicates that J48 performs roughly the same as CART in terms of accuracy
- ROC analysis shows that J48 is less sensitive to threshold changes than cart, i.e., it exhibits smooth ROC curves

Naïve Bayes classification

- "Out of the box," naïve Bayes has the best F_1 score of 0.2, but a relatively low accuracy of 75 %.
- Unfortunately, thresholding lowers the F_1 score in order to account for the low accuracy
- F_1 is lowered to 0.193, but accuracy boosted to 80 %
- Naïve Bayes exhibits the smoothest ROC curves and is least dependent on the threshold

Threshold dependence curves for naïve Bayes



Setting up the kNN algorithm

- Weka's implementation of kNN will be used, also called IBk (instance-based learning)
- Cross-validation will be used to determine the value of $k \in \{1, \ldots, 13\}$
- $k = 4$ ends up being taken as optimum
- k-d trees will be used for nearest-neighbor queries in $\mathcal{O}(\log n)$ time

kNN classification

- Unfortunately, it turns out the kNN is the poorest of all classifiers
- `=== Detailed Accuracy By Class ===`

	ROC Area	F-Measure	Class
	0.532	0.889	Other
	0.532	0.128	Readmitted
Weighted Avg.	0.532	0.814	

- The accuracy is 80 %, but as it can be seen, the area under the ROC curve leaves a lot to be desired

Outline

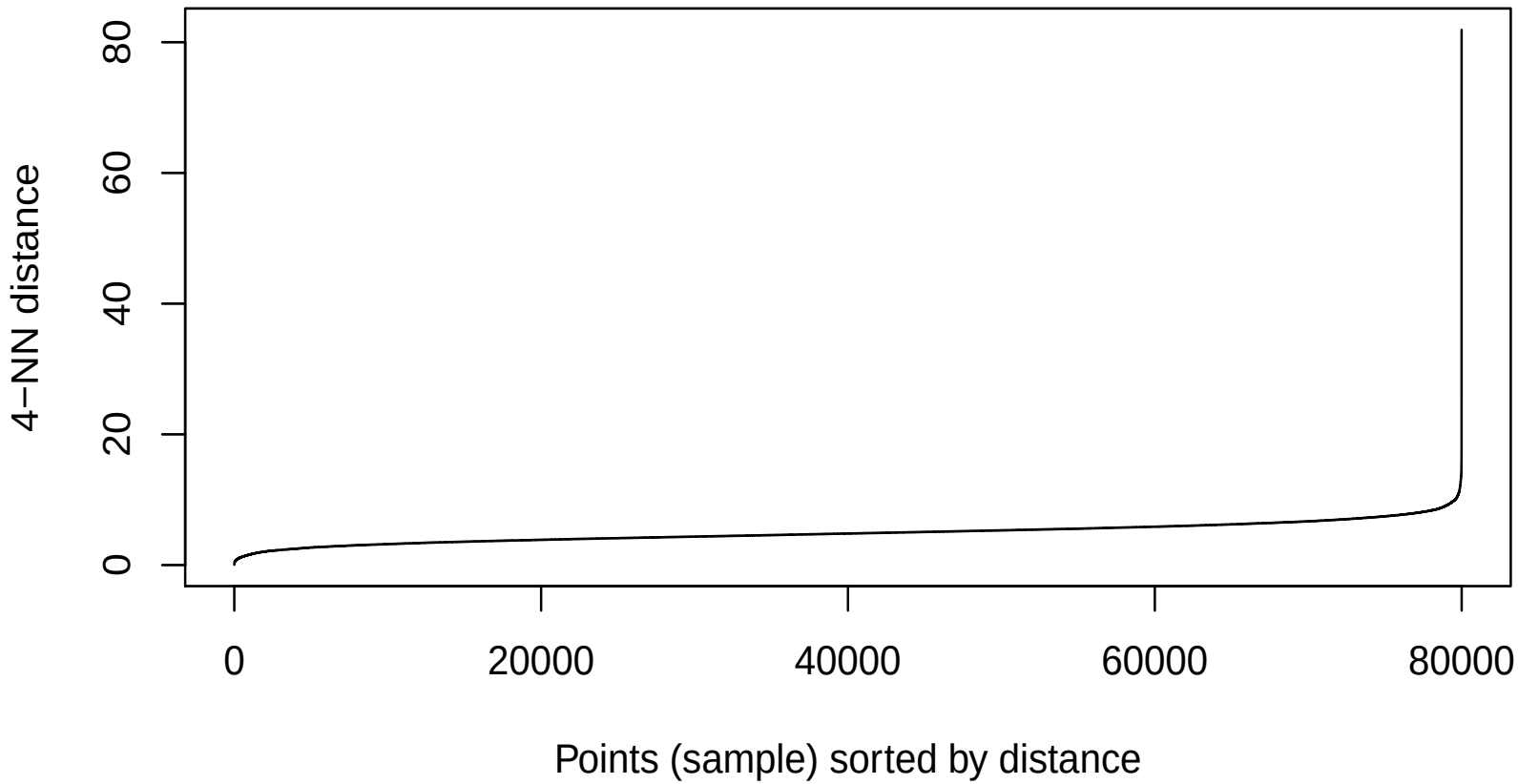
Setting up the DBSCAN algorithm

DBSCAN's parameters will be determined by the following widely accepted strategy:

1. Eps is determined by looking at a knee of a kNN distance plot (I will take $k = 4$);
2. MinPts is set to $\#\{\text{dimensions}\} + 1$.

The figure on the next slide shows a knee at distance ≈ 11, so I will take Eps $= 1$.

Identifying a knee in a 4NN plot



DBSCAN clustering and analysis

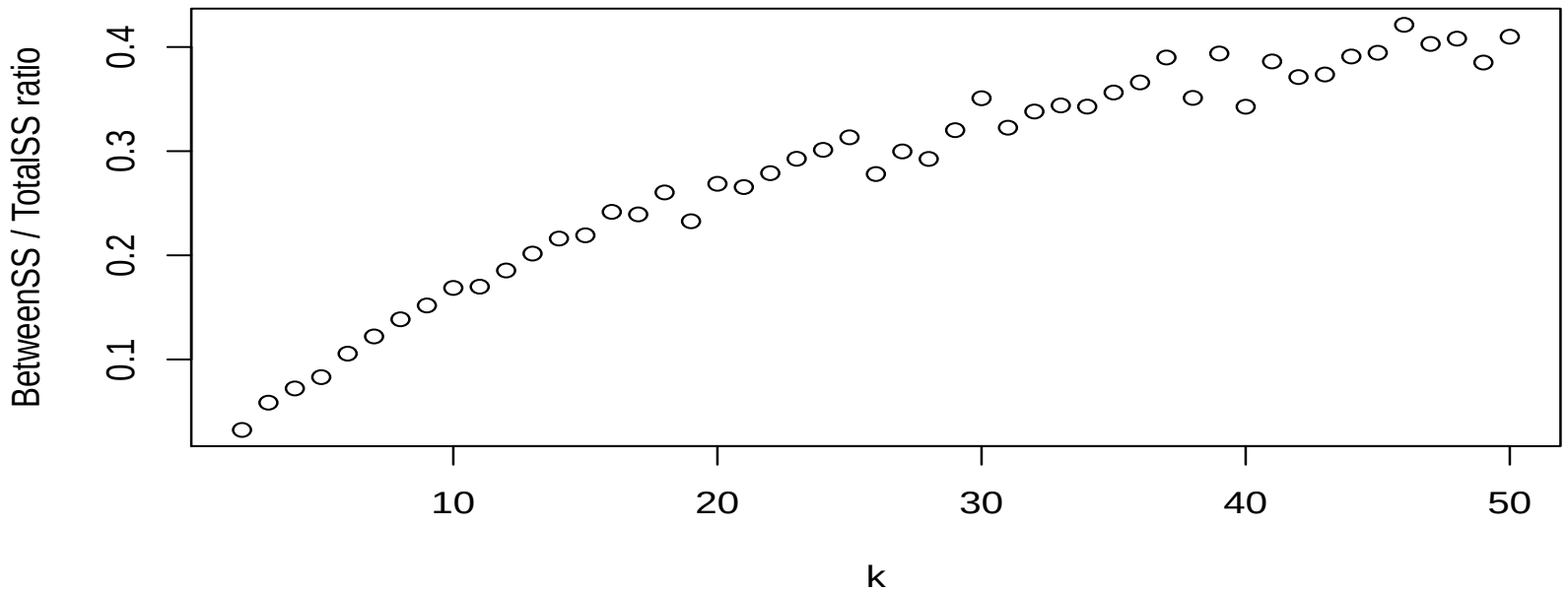
- DBSCAN discovers 4 clusters and 179 noise points

```
## DBSCAN clustering for 20000 objects.
## Parameters: eps = 11, minPts = 73
## The clustering contains 4 cluster(s) and
## 179 noise points.
##
##     0     1     2     3     4
##   179 19374   152   141   154
```

- These clusters do not correspond to our `readmitted` attribute

k-means clustering and analysis

I will try k values ranging from 2 to 50. The BetweenSS to TotalSS ratios can be seen below. They are poor even for $k = 50$, leading me to believe that there are no well-separated clusters in terms of k-means.



Outline

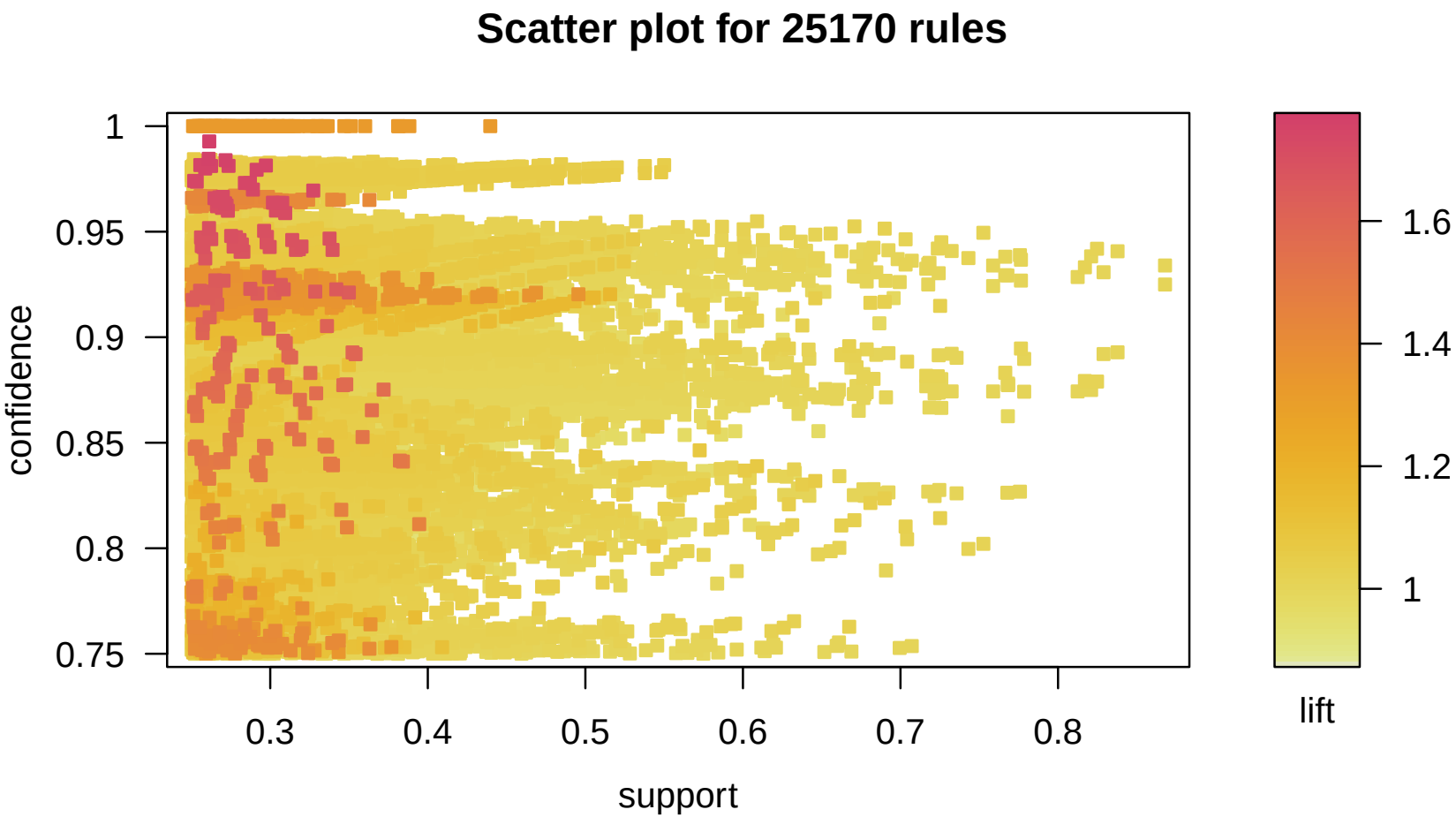
The setup for mining association rules

- The apriori algorithm will be used
- Numeric attributes will be discretized by clustering
- There appear to be many frequent items making up for good left-hand side candidates
- The minimum support will be set to 0.25, the minimum confidence to 0.75, and the maximum number of items in a rule to 15
- Confidence and lift will be used as "goodness" measures

Running the apriori algorithm

- The apriori algorithm generates 27 868 rules
- Rules with the highest lift are those which tell us whether a patient should be administered a change in medicine
- Rules with the highest confidence are those which tell us whether a patient should be administered any medicine
- These rules are, fortunately, not trivial

Visualizing the generated rules



Outline

Classification insights

- Classification is hard due to the imbalance, but thresholding helps
- SMOTE may lead to overfitting
- Thus, it might be worthwhile to explore ensemble-based algorithms, which are known to be less susceptible to overfitting
- Adding pairwise interactions is likely to help due to the added nonlinearity

Clustering and association rules insights

- There are no well-defined clusters in terms of DBSCAN or k-means
- However, meaningful association rules *do exist*
- Additional data and/or attributes might be needed in order to look for clusters or improve on other concepts