

Data Mining - PIMA

Lanpei Li
lilanpei@gmail.com

& Ahmad Alleboudy
ahmad.alleboudy@outlook.com

Overview

Here we present our final paper for the data mining 1 course on the Pima (Diabetes Detection) dataset

Task 1: Data Understanding

Data Semantics

Pregnancies: Number of times the subject got pregnant (discrete variable, integer)

Glucose: Plasma glucose concentration a 2 hours in an oral glucose tolerance test (discrete variable, integer)

Blood Pressure: Diastolic blood pressure (mm Hg) (discrete variable, integer)

Skin Thickness: Triceps skinfold thickness (in mm) (discrete variable, integer)

Insulin: 2-Hour serum insulin (mu U/ml) (discrete variable, integer)

BMI: Body mass index (weight in kg/(height in m)²) (continuous variable, float)

Diabetes Pedigree Function: Diabetes pedigree function (continuous variable, float)

Age: Age (in years, discrete variable, integer)

Outcome: Binary variable (0 or 1)

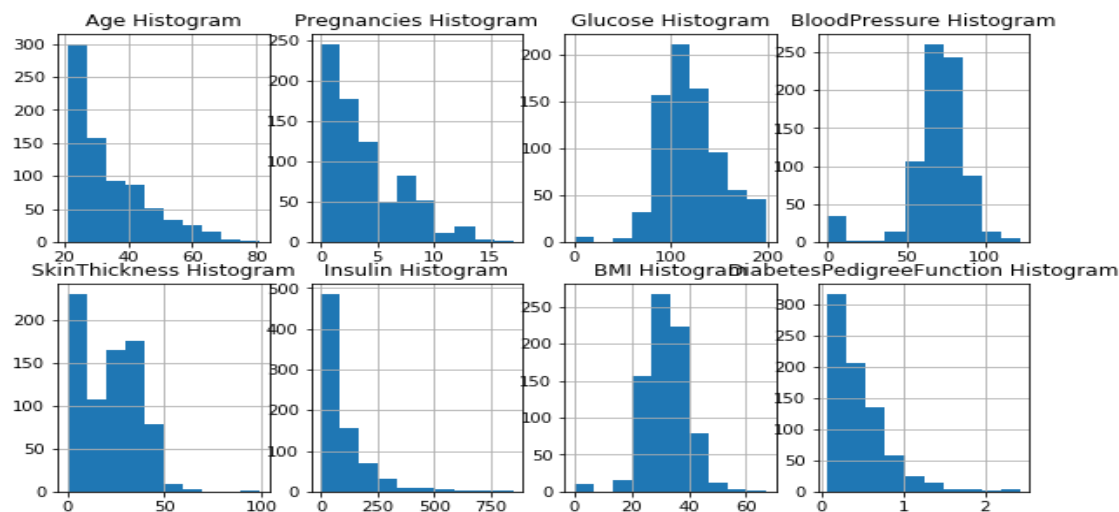
Distributions of Variables and statistics

The following table shows the summary statistics per attribute for the dataset

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
count	768.000000	768.000000	768.000000	768.000000	768.000000	768.000000	768.000000	768.000000	768.000000
mean	3.845052	120.894531	69.105469	20.536458	79.799479	31.992578	0.471876	33.240885	0.348958
std	3.369578	31.972618	19.355807	15.952218	115.244002	7.884160	0.331329	11.760232	0.476951
min	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.078000	21.000000	0.000000
25%	1.000000	99.000000	62.000000	0.000000	0.000000	27.300000	0.243750	24.000000	0.000000
50%	3.000000	117.000000	72.000000	23.000000	30.500000	32.000000	0.372500	29.000000	0.000000
75%	6.000000	140.250000	80.000000	32.000000	127.250000	36.600000	0.626250	41.000000	1.000000
max	17.000000	199.000000	122.000000	99.000000	846.000000	67.100000	2.420000	81.000000	1.000000

We notice min values of 0 in some biological measurements that normally shouldn't allow zero like BMI and Blood pressure, so, it is possible that the 0 values present are missing values

The following plot shows the distributions of the different attributes in the dataset



As the plots show

Age is Right Skewed, meaning more young subject appear than older ones in the dataset

Also, Pregnancies is right skewed

Glucose seems not skewed, however, values of 0 indicate missing readings

Same for the Blood, however values less than 40 indicate outliers and missing values at 0

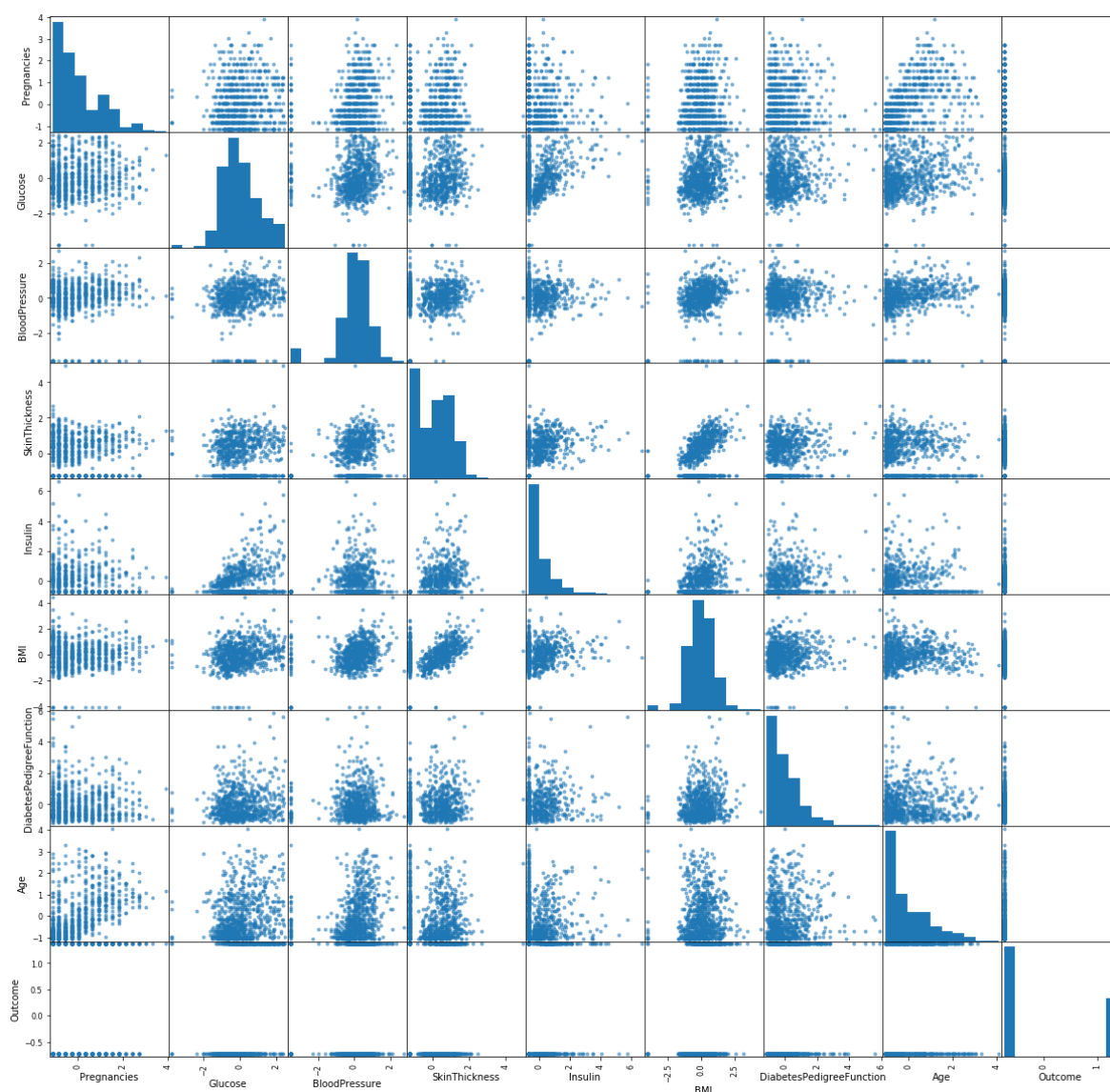
Skin thickness seems to be not skewed, but with possibly outliers near 100

Insulin and DiabetesPedigreeFunction are right skewed

BMI shows an outlier at 0 and possibly at 60 as well and is centered around 30

Also, not shown, but the distribution of the outcomes are 500 non diabetic subjects vs 268 diabetic ones

The following is the correlation plots of the attributes



As the figure shows We can easily notice some intuitive positive correlations between some attributes:

Age and Pregnancies - BMI and skin thickness - BMI and blood pressure - Insulin and Glucose

Some of which might be candidate for elimination(please see Assessing Data quality in the following page)

Assessing data quality (missing values, outliers)

• Missing Values

Although there are no null values, 376 is the number of examples where at least a 0 existed in their attributes other than Pregnancies and Outcome, that is about 47% of the examples, we shouldn't exclude such big number of examples, so we decided to substitute for the missing values.

For each attribute we computed two averages 1- the average of the non zero values that had Outcome 0

2- the average of the zero values that had Outcome 1

We substitute for the missing values with their corresponding outcome average.

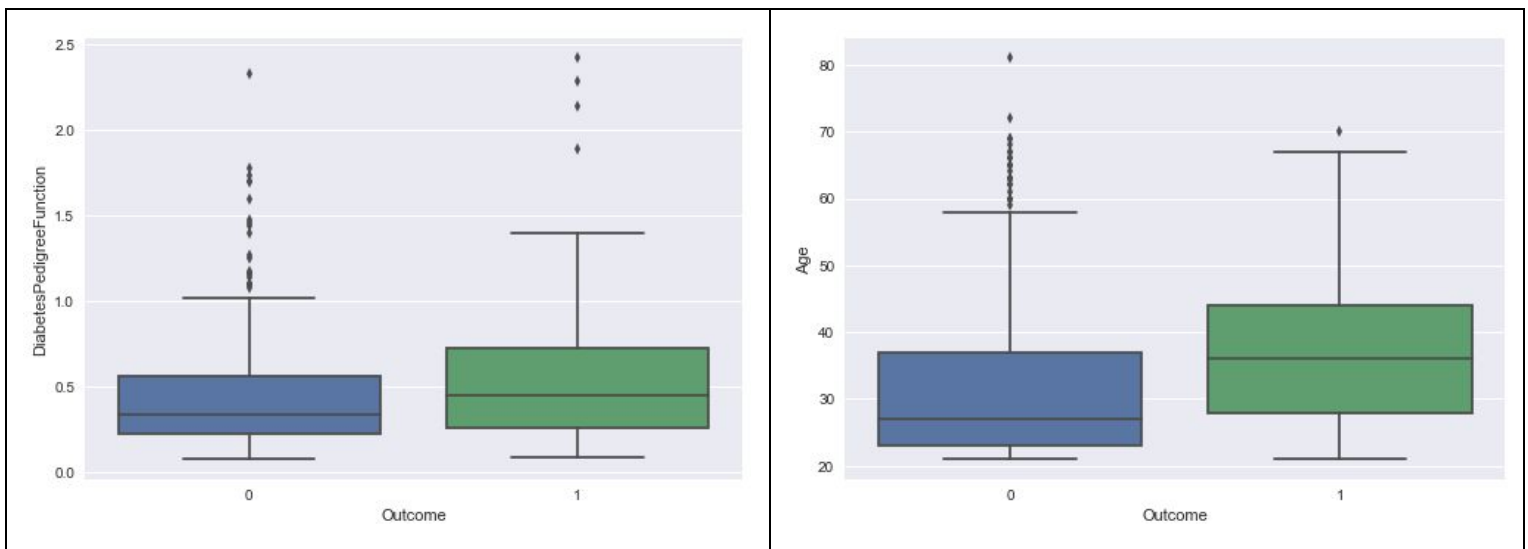
Also, we decided to drop the features Insulin and Skin thickness since they have the most missing values

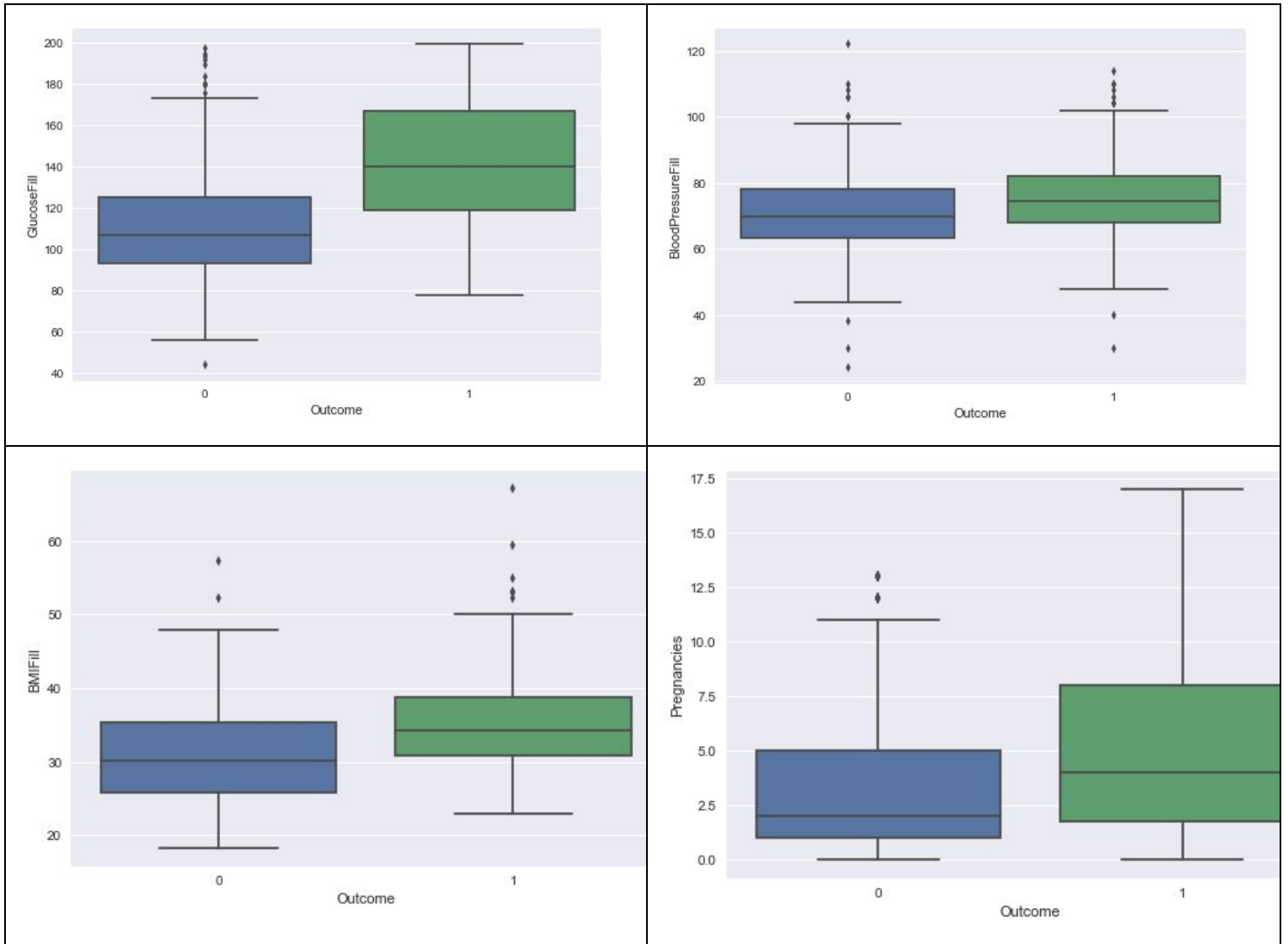
And they already have strong positive correlations with BMI and Glucose respectively as shown in the correlations plot

Missing values per attribute: [Glucose:5, Blood Pressure:35, Skin thickness: 227, Insulin: 374, BMI:11, DiabetesPedigreeFunction:0, Age: 0]

• Outliers

We plotted the box plots of all of the attributes grouped by the Outcome

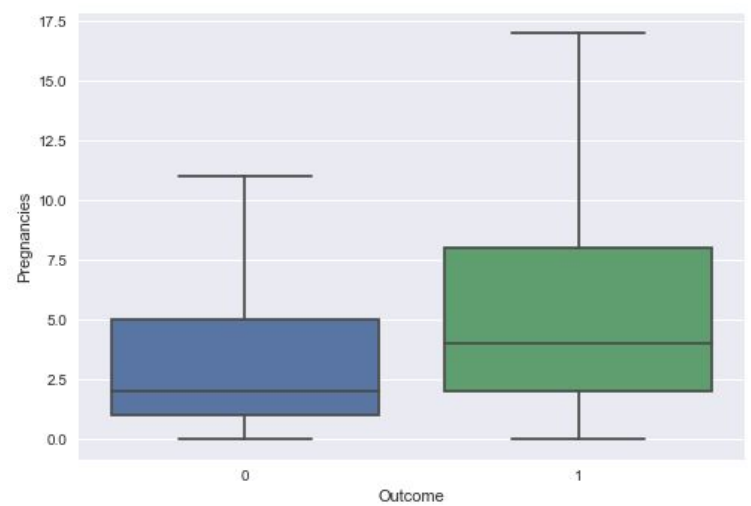
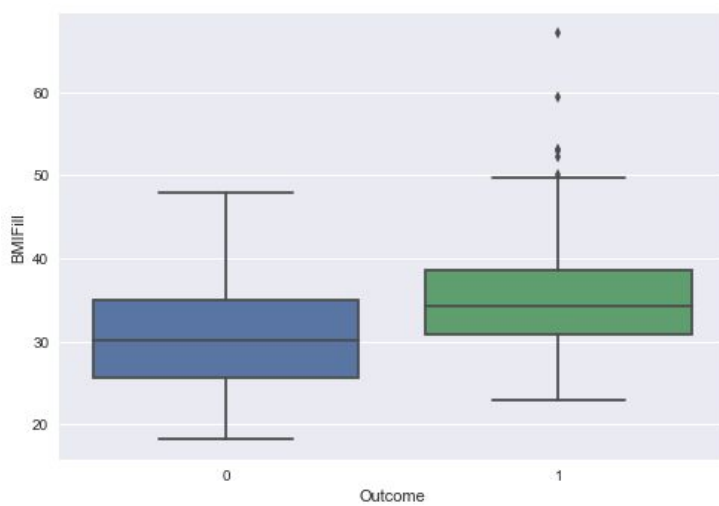
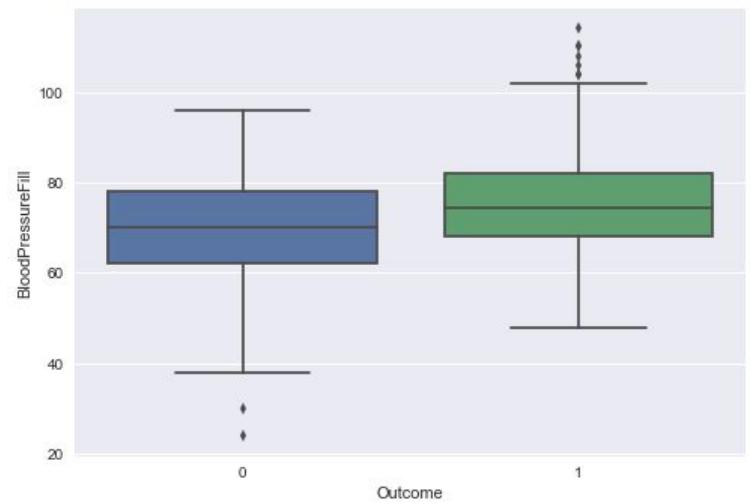
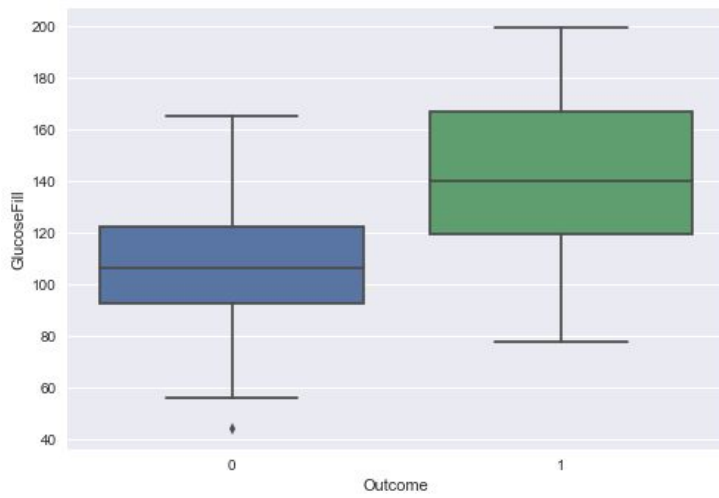
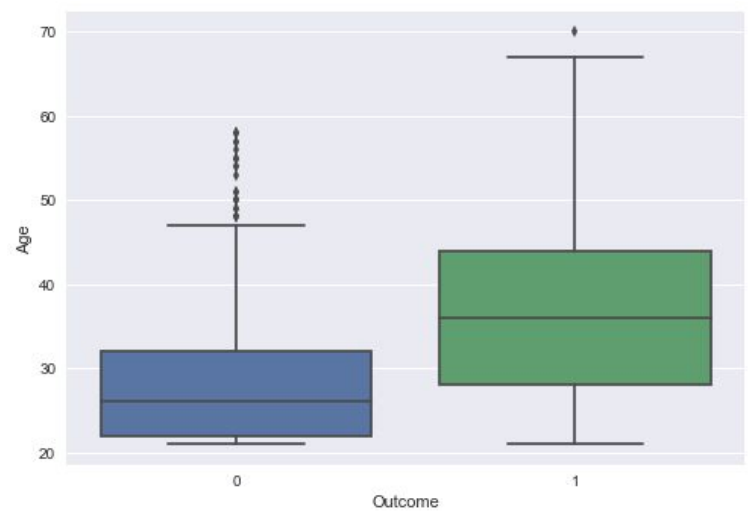
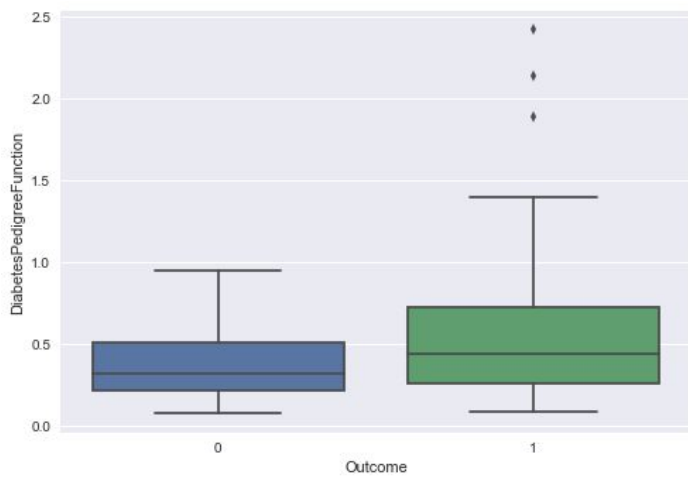




We found that the median of the Outcome 1 are always higher than the median for the Outcome 0 for all of the attributes so, we excluded values larger than $1.5 \times$ the third quantile if the outcome is 0 or less than the first quantile/1.5 for all of the attributes if the outcome is 1

We ended up with 697 examples

Box plots after outliers elimination:



Task 2: Clustering

Clustering Analysis by K-means

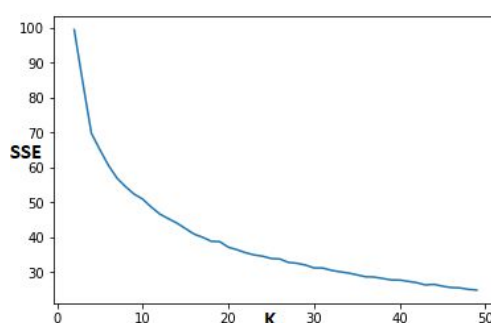
- **Choice of attributes and distance function**

We used all of the attributes after applying the missing values estimation except for Insulin and Skin Thickness as discussed in the Data understanding task.

Since our data are all discrete and continuous numerical values we chose the Euclidean distance as our distance function

- **Identification of the best value of k ,(we picked 11)**

We tried a range of values of k from 2 to 50, each time computing the SSE as shown in the plot



As shown, a suitable value for K is about 11 (the knee of the plot)

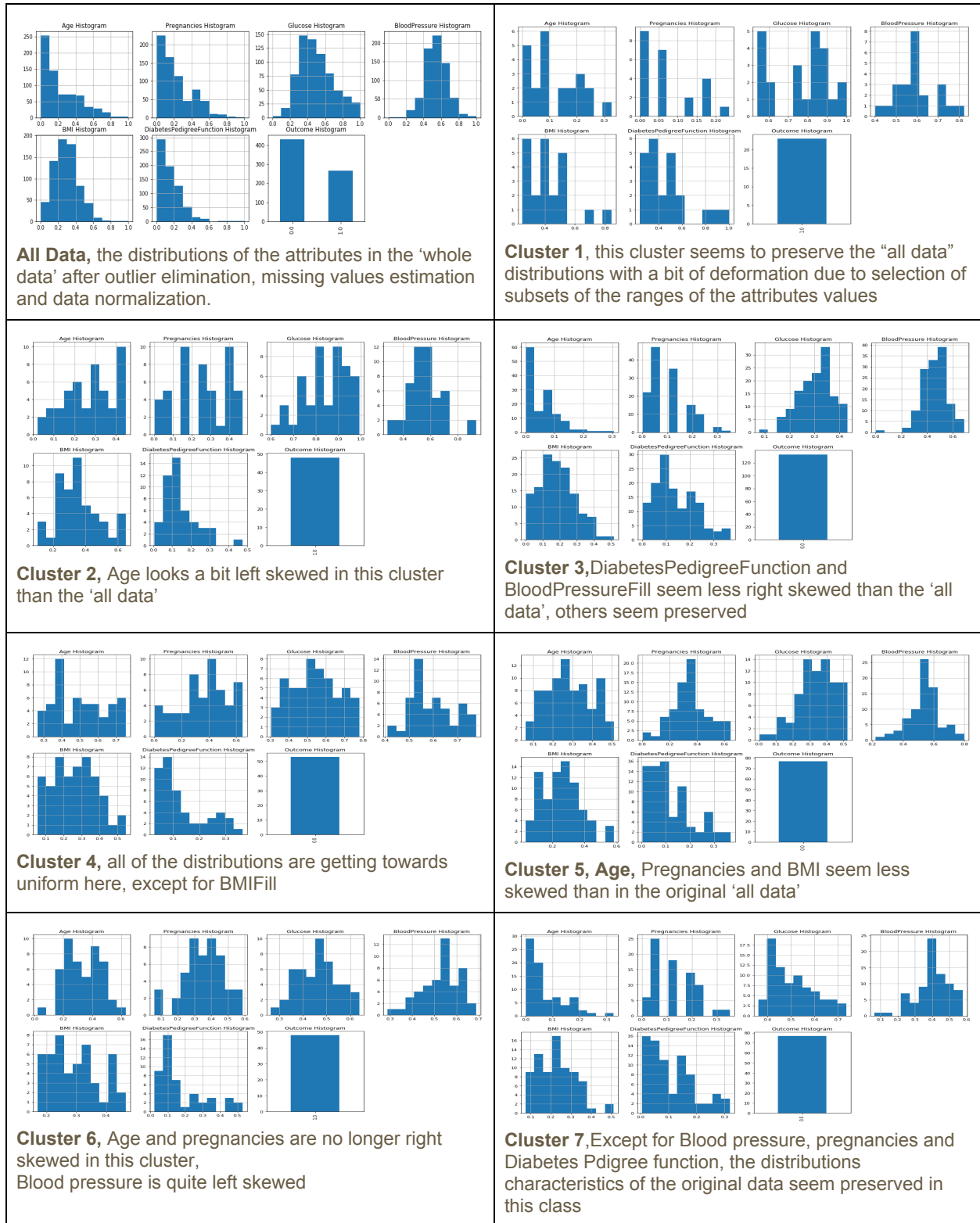
- **Characterization of the obtained clusters**

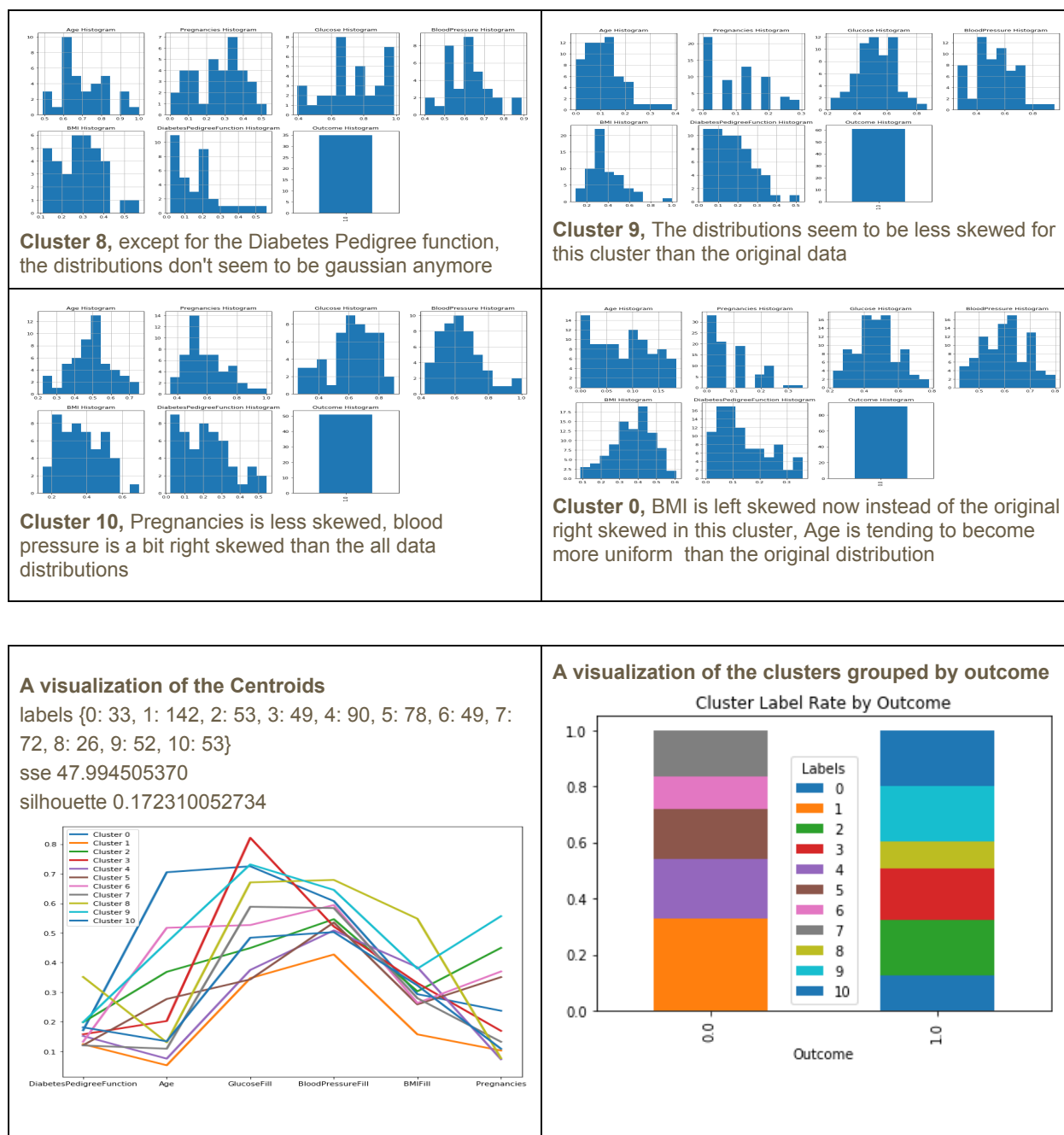
First, analysis of the k centroids As shown in the following table, the centroids of the clusters' attributes are quite different from the all data attribute averages, this is due to that the k means clustering managed to untangle the data into quite homogenous clusters based on the attributes values combinations

Cluster	Age	BMIFill	BloodPressureFill	DiabetesPedigreeFunction	GlucoseFill	Outcome	Pregnancies
0	0.70408	0.29007	0.608333333	0.169593296	0.717137097	0	0.22977941
1	0.0511	0.14838	0.425062657	0.121180406	0.361096289	1	0.09774436
2	0.46977	0.38037	0.649056604	0.193480818	0.739257456	0	0.54827969
3	0.5102	0.27187	0.584486373	0.12573514	0.518928789	1	0.37402886
4	0.19825	0.34272	0.533106576	0.151919692	0.821198157	0	0.16326531
5	0.36698	0.3077	0.544146825	0.201682018	0.456451613	1	0.45063025
6	0.107	0.30288	0.596996997	0.126237681	0.583173496	1	0.12082671
7	0.09262	0.46248	0.581196581	0.556559154	0.765260546	0	0.05882353
8	0.07209	0.36825	0.493357488	0.152814391	0.351542777	1	0.08120205
9	0.13703	0.37329	0.546296296	0.179239017	0.495340502	0	0.09897292
10	0.26169	0.24907	0.535021097	0.122247565	0.340955492	1	0.34102755
ALL Data	0.22467	0.2914	0.533397	0.156078	0.497913	0.381636	0.219259

Second, comparison of the distribution of variables within the clusters and that in the whole dataset

The plots labels are the attributes names, they are from the top left to right bottom :[Age,Pregnancies, GlucoseFill, BloodPressureFill, BMIFill, DiabetesPedigreeFunction, Outcome]





Analysis by density-based clustering

- **Choice of attributes and distance function**

Using a similar logic as mentioned before, we dropped Insulin and skin thickness, and used all of the other attributes.

For the distance function since the attributes are all numerical we used the euclidean distance

- **Study of the clustering parameters**

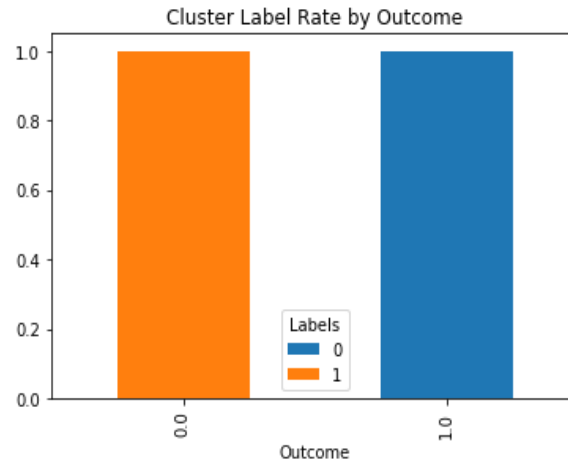
We conducted a grid search for eps and the minimum number of samples per cluster and concluded with eps= 0.6,min samples = 5

- **Characterization and interpretation of the obtained clusters**

We acquired two clusters, without any outlier points

labels {0: 266, 1: 431, -1: 0} , **silhouette 0.578484488776**, each cluster representing one of the outcomes we have

Shown is the visualization of clusters by grouped by outcome

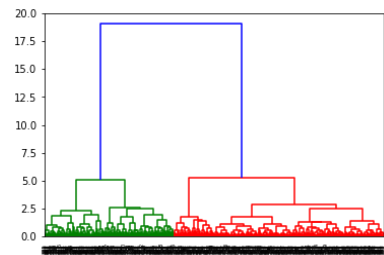
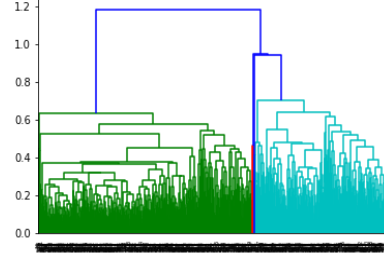
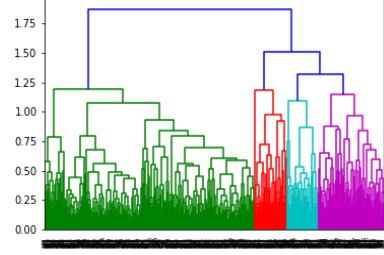


Analysis by hierarchical clustering

- **Choice of attributes and distance function**

Following the results from the k means search, we used 11 clusters, same logic per the distance metric and attributes is used.

The following table shows the results and dendrograms obtained from different objective functions

'ward'	'Average'	'Complete'
labels {0: 64, 1: 161, 2: 48, 3: 40, 4: 18, 5: 23, 6: 83, 7: 105, 8: 82, 9: 31, 10: 42}	labels {0: 111, 1: 369, 2: 7, 3: 55, 4: 22, 5: 3, 6: 110, 7: 1, 8: 4, 9: 8, 10: 7}	labels {0: 39, 1: 75, 2: 86, 3: 21, 4: 410, 5: 13, 6: 1, 7: 1, 8: 35, 9: 3, 10: 13}
silhouette 0.142751609462	silhouette 0.236855407563	silhouette 0.255366719541
		

Conclusion for the different clustering approaches used:

The density based clustering seems to introduce the best **silhouette score of 0.578484488776**

And it follows a very intuitive clustering that it eventually clustered the samples into the desired outcome 0 vs outcome 1 which can be useful for classification of new examples.

Task 3: Association Rules Mining

Frequent patterns extraction with different values of support and different types

We discretized the attributes values as follows:

Age bins=range(from 20, to 90, step 10), Glucose bins=range(from 40, to 210, step 10), Blood pressure bins=range(from 20, to 121, step 10), BMI bins=range(from 15, to 80, step 10), DiabetesPedigreeFunction bins=[0.0,0.5,1.0,1.5,2.0,2.5], Pregnancies bins=range(from 0, to 17, step 3)

We fixed the minimum number of items in each rule to 3, since 2 resulted into too many frequent patterns and more than three got too strict producing very few patterns

Minimum Support	Frequent	Closed	Maximal
2	812 frequent patterns	756 frequent patterns	283 frequent patterns
10	43 frequent patterns	43 frequent patterns	23 frequent patterns
20	7 frequent patterns	7 frequent patterns	3 frequent patterns

As shown in the table above, 2 min support produces quite too many frequent patterns and 20 is quite too strict producing too little patterns, so, we decided to use 10 minimum supports and we used the type of 'Frequent', since it is quite similar to close and maximal is quite strict producing less patterns than both of them.

Discussion of the most interesting frequent patterns

[Pregnancies <= 2] [20 < Age <= 30] [outcome = negative]	Support = 30.99
For most of the yielded patterns when the sample had a low number of pregnancies and a quite young age[20:30] the outcome was negative, which is quite intuitive.	
[25 < BMI <= 35] [outcome = negative] [0.0 < DiabetesPedigreeFunction <= 0.5]	Support = 24.6772
For a quite low BMI and DiabetesPedigreeFunction, the outcome is negative	
[20 < Age <= 30] [outcome = negative] [0.0 < DiabetesPedigreeFunction <= 0.5]	Support = 30.99
For a quite young age and DiabetesPedigreeFunction the outcome was negative	

Association rules extraction with different values of confidence

Minimum Confidence	Number of Association rules
15	983
25	595
45	258
65	113
90	8

We decided that a quite reasonable choice would be 25, since lower than this is too relaxed and higher gets too strict quite quickly

Discussion of the most interesting rules

Sample rules Top 3 with a negative outcome :

<p> $([60 < \text{Blood pressure} \leq 70] [20 < \text{Age} \leq 30] [0.0 < \text{DiabetesPedigreeFunction} \leq 0.5])$ $\Rightarrow ([\text{outcome} = \text{negative}])$ </p>	<p>Lift = 150.302</p> <p>Conf = 92.9412</p>
A low blood pressure for a young age sample with a small DiabetesPedigreeFunction yields a negative outcome	
<p> $([15 < \text{BMI} \leq 25] [0.0 < \text{DiabetesPedigreeFunction} \leq 0.5])$ $\Rightarrow ([\text{outcome} = \text{negative}])$ </p>	<p>Lift = 150.64</p> <p>Conf = 93.1507</p>
A quite low BMI and DiabetesPedigreeFunction yields a negative outcome	
<p> $([\text{Pregnancies} \leq 2] [20 < \text{Age} \leq 30] [25 < \text{BMI} \leq 35] [0.0 < \text{DiabetesPedigreeFunction} \leq 0.5])$ $\Rightarrow ([\text{outcome} = \text{negative}])$ </p>	<p>lift = 147.204</p> <p>Conf = 91.0256</p>
A small number of pregnancies a young age and a small BMI and DiabetesPedigreeFunction yields a negative outcome, which is very intuitive	

Sample rules Top 3 with a positive outcome :

<p> $([40 < \text{Age} \leq 50] [0.0 < \text{DiabetesPedigreeFunction} \leq 0.5])$ $\Rightarrow ([\text{outcome} = \text{positive}])$ </p>	<p>Lift = 144.806</p> <p>Conf = 55.2632</p>
A quite old age and despite a small value for DiabetesPedigreeFunction yields a positive outcome	
<p> $([30 < \text{Age} \leq 40] [70 < \text{Blood pressure} \leq 80])$ $\Rightarrow ([\text{outcome} = \text{positive}])$ </p>	<p>Lift = 134.654</p> <p>Conf = 51.3889</p>
A quite middle age with a quite high blood pressure yields a positive outcome	

$([35 < \text{BMI} \leq 45] [70 < \text{Blood pressure} \leq 80])$ $\Rightarrow ([\text{outcome} = \text{positive}])$	Lift = 131.015 Conf = 50.0
A high BMI with a quite high blood pressure yields a positive outcome	

Use the most meaningful rules to replace missing values and evaluate the accuracy

We joined back the two columns we dropped, first Skin Thickness then 'Insulin' into our data, to try and compute their missing values using the rules we acquired from the previous step.

Starting with Skin thickness, we discretized it into bins=range(from 0, to 110,step 10)

We split our data into two part, a train set where no missing values for Insulin and skin thickness and a test set where missing values for insulin and skin thickness existed.

We used the train split to produce the rules then applied them on the test split to get their missing values for skin thickness.

We used the same procedure for the choice of the frequent patterns type choice (frequent), the min items = 3, the min support 10, the we used the resulting patterns to get the rules with a confidence ≥ 25

And we picked the top 100 rules with skin thickness in the consequence, sorted by the confidence and applied them to the test split.

Unfortunately, due to the lack of enough examples for training to generate the rules, the outcome was always in the range from 30 to 40 for the skin thickness in the test set

Same procedure was carried out for 'Insulin', discretized as bins=range(from 0, to 900, step 100). And the resulting values were between 0 and 100.

The procedure for applying the rules was, for each sample we take the rules that had the largest intersection between the attributes and the anticipation of the rules, for each set of resulting rules we made a vote for the rules outcome and took the majority as the outcome for the missing value we are trying to estimate.

We restricted the number of acceptable intersections to 2

A problem that occurred was that, some samples didn't didn't have any intersections with any of the rules,such examples were skipped.

Since we didn't know the target values for the missing values,it was not possible to evaluate the accuracy of applying the rules.

Use the most meaningful rules to predict if the diabetes is detected and evaluate the accuracy

The same procedures as estimating the missing values part was carried out, but with the 'Outcome' omitted and used as a target.

Due to the problem that not all of the samples had an intersection with the anticipation of the rules larger than the threshold we defined of 2 We eventually acquired 551 examples, 236 of them were correct (accuracy is quite low 0.42)

Task 4: Classification

Learning of different decision trees with different parameters and gain formulas with the object of maximizing the performances

Searching for the best performance with different parameters (Gain formulas, and min_samples_leaf and depth)

- Gini gain**

Depth\minSamplesInLeaf	1	5	10	15
3	Precision : 0.790285407087 Recall : 0.790530846485 F1 : 0.784318055182 Accuracy : 0.790530846485	Precision : 0.787508211846 Recall : 0.787661406026 F1 : 0.781074373831 Accuracy : 0.787661406026	Precision : 0.787508211846 Recall : 0.787661406026 F1 : 0.781074373831 Accuracy : 0.787661406026	Precision : 0.787508211846 Recall : 0.787661406026 F1 : 0.781074373831 Accuracy : 0.787661406026
6	Precision : 0.87480930032 Recall : 0.875179340029 F1 : 0.874947132826 Accuracy : 0.875179340029	Precision : 0.858781180198 Recall : 0.859397417504 F1 : 0.858970564002 Accuracy : 0.859397417504	Precision : 0.8473673351 Recall : 0.847919655667 F1 : 0.847577900587 Accuracy : 0.847919655667	Precision : 0.838251543945 Recall : 0.83787661406 F1 : 0.838048063073 Accuracy : 0.83787661406
9	Precision : 0.955495600091 Recall : 0.955523672884 F1 : 0.955507561543 Accuracy : 0.955523672884	Precision : 0.897853078778 Recall : 0.898134863702 F1 : 0.897945361271 Accuracy : 0.898134863702	Precision : 0.859171915501 Recall : 0.859397417504 F1 : 0.857560538322 Accuracy : 0.859397417504	Precision : 0.84123014286 Recall : 0.842180774749 F1 : 0.841443210864 Accuracy : 0.842180774749
No limit	Precision : 1.0 Recall : 1.0 F1 : 1.0 Accuracy : 1.0	Precision : 0.899197318136 Recall : 0.899569583931 F1 : 0.899100225095 Accuracy : 0.899569583931	Precision : 0.859171915501 Recall : 0.859397417504 F1 : 0.857560538322 Accuracy : 0.859397417504	Precision : 0.84123014286 Recall : 0.842180774749 F1 : 0.841443210864 Accuracy : 0.842180774749

- Entropy gain

Depth\minSamplesInLeaf	1	5	10	15
3	Precision : 0.788232596574 Recall : 0.787661406026 F1 : 0.780480362586 Accuracy : 0.787661406026	Precision : 0.788232596574 Recall : 0.787661406026 F1 : 0.780480362586 Accuracy : 0.787661406026	Precision : 0.788232596574 Recall : 0.787661406026 F1 : 0.780480362586 Accuracy : 0.787661406026	Precision : 0.788232596574 Recall : 0.787661406026 F1 : 0.780480362586 Accuracy : 0.787661406026
6	Precision : 0.869414458046 Recall : 0.860832137733 F1 : 0.855824497998 Accuracy : 0.860832137733	Precision : 0.854508957573 Recall : 0.855093256815 F1 : 0.854710840861 Accuracy : 0.855093256815	Precision : 0.84405737448 Recall : 0.845050215208 F1 : 0.844194482409 Accuracy : 0.845050215208	Precision : 0.838196810363 Recall : 0.83931133429 F1 : 0.837691909227 Accuracy : 0.83931133429
9	Precision : 0.937473025187 Recall : 0.9368723099 F1 : 0.937044242372 Accuracy : 0.9368723099	Precision : 0.905177962504 Recall : 0.90243902439 F1 : 0.903047751948 Accuracy : 0.90243902439	Precision : 0.866151618606 Recall : 0.866571018651 F1 : 0.86516169026 Accuracy : 0.866571018651	Precision : 0.850009439752 Recall : 0.850789096126 F1 : 0.850215440018 Accuracy : 0.850789096126
No limit	Precision : 1.0 Recall : 1.0 F1 : 1.0 Accuracy : 1.0	Precision : 0.913649050828 Recall : 0.913916786227 F1 : 0.913655447348 Accuracy : 0.913916786227	Precision : 0.866151618606 Recall : 0.866571018651 F1 : 0.86516169026 Accuracy : 0.866571018651	Precision : 0.850009439752 Recall : 0.850789096126 F1 : 0.850215440018 Accuracy : 0.850789096126

We see that a no limit for the depth combined with a min samples in a leaf of 1 gives a 100% performance, however, this is definitely overfitting the data.

We will decide the choice for the parameters later, when doing the training and validation of the model. But we can conclude from the table above that the Entropy formula is slightly better than Gini for our data.

Decision trees validation with test and training set

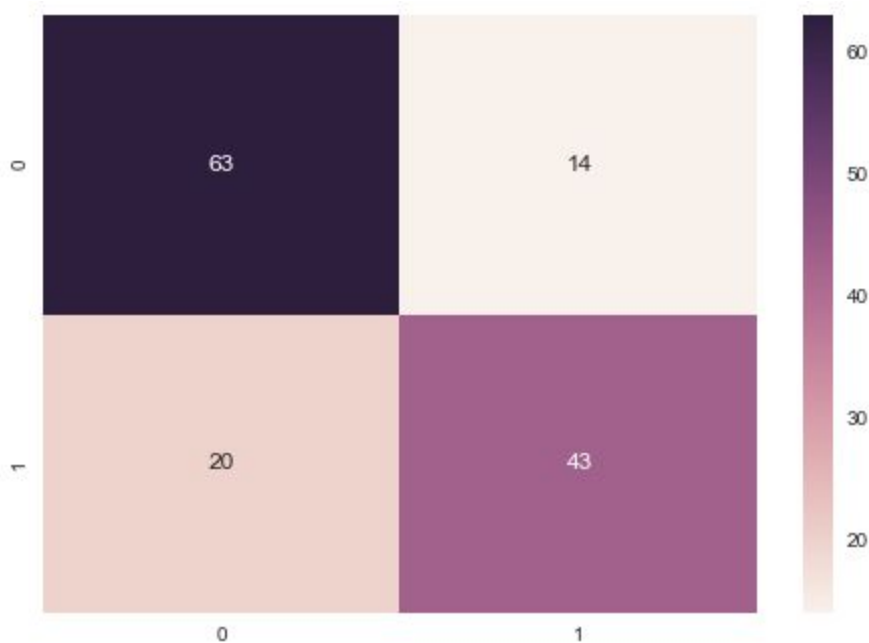
We kept 20% of the data for validation with 140 validation samples and 557 training samples

Entropy gain

Depth\minSamplesInLeaf	5	10	15
3	Train Accuracy : 0.800718132855 Validation Accuracy : 0.7	Train Accuracy :0.800718132855 Validation Accuracy : 0.7	Train Accuracy : 0.800718132855 Validation Accuracy : 0.7
6	Train Accuracy : 0.867145421903 Validation Accuracy : 0.714285714286	Train Accuracy : 0.858168761221 Validation Accuracy : 0.714285714286	Train Accuracy : 0.856373429084 Validation Accuracy : 0.75
9	Train Accuracy : 0.899461400359 Validation Accuracy : 0.685714285714	Train Accuracy : 0.867145421903 Validation Accuracy : 0.757142857143	Train Accuracy : 0.856373429084 Validation Accuracy : 0.75

We decided to use Depth of 9 with min samples per leaf = 10

In the following table we report the confusion matrix of the model for the validation set



In the following we report the performance metrics of the validated model

	precision	recall	f1-score	support
Non-diabetic	0.76	0.82	0.79	77
Diabetic	0.75	0.68	0.72	63
avg / total	0.76	0.76	0.76	140

Applying K-fold with k=10 cross validation for the model we acquired Accuracy: 0.78 (+/- 0.14) [Fig CV] shows the model interpretation.

Discussion of the best prediction model

We tried KNN against Decision Trees and here we report our results

- **KNN**

K	10-Folds cross validation Accuracy
5	0.76 (+/- 0.11)
10	0.76 (+/- 0.10)
15	0.77 (+/- 0.09)
20	0.77 (+/- 0.10)
25	0.78 (+/- 0.09)
30	0.77 (+/- 0.09)

- **Decision Trees**

We tried a random search for the best tree model with the following candidate values for 50 different trees:

max_depth	[2,3,4,5,6,7,8,9,10,11,12,None]
criterion	["entropy", "gini"]
Max_features used	Randomly picked 1 to all of the features
min_samples_leaf	Randomly picked From 10 to 51

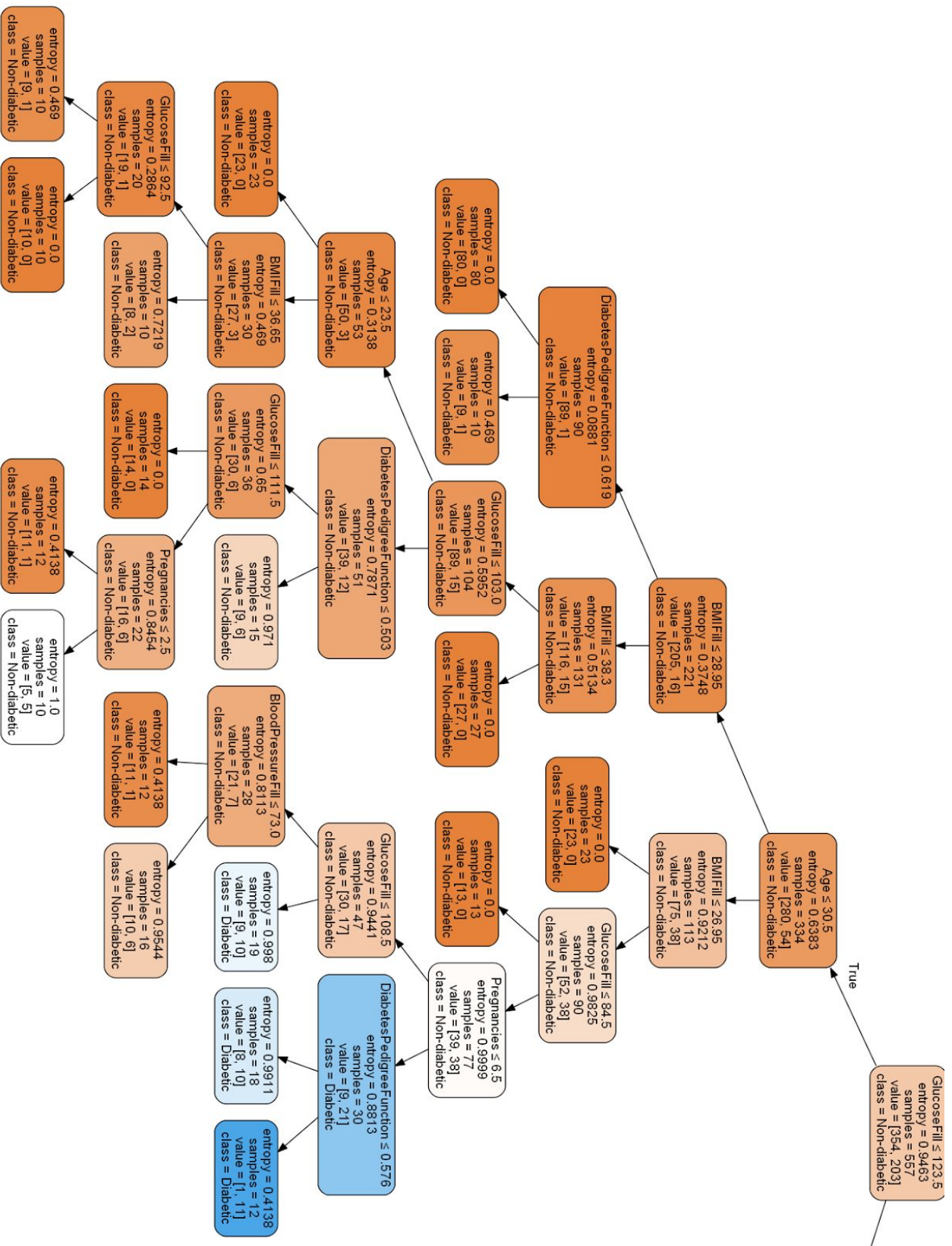
And the yielded best model is:

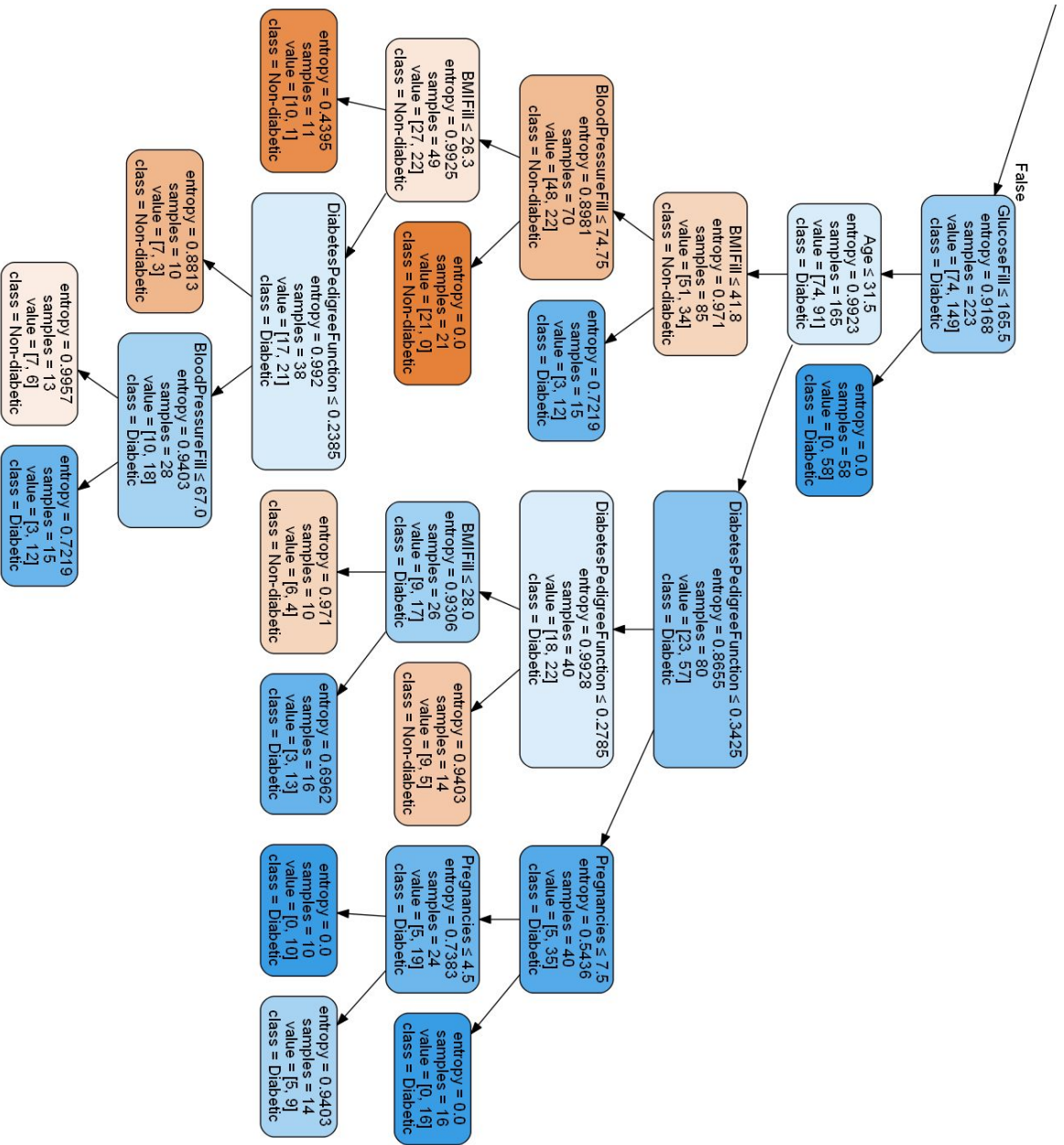
criterion='gini', max_depth=9,max_features=all, min_impurity_split=1e-07,min_samples_leaf=1
With mean 10-folds cross validation accuracy of 0.803 (+/- 0.025)

We concluded by choosing the Tree model as it had better accuracy than the KNN model

Decision trees interpretation

Following is [Fig CV] is the detailed interpretation of the Validated tree model





Summary

- The given data has lots of missing values which made it very noisy for the given tasks even after providing a means for estimating such missing values, 'insulin' and 'skin thickness' were still difficult to benefit from.
- A density Based Clustering was the best approach to tackle the clustering of the dataset, other clustering algorithms tried performed bad.
- For rules mining, it was crucial to balance the number of attributes in the anticipates of the rules, fixing a high confidence for the rules and applying rules where at least more than two attributes of the samples were present in the rule anticipate.
- For classification, Decision trees performed better than KNN and it was crucial not to overfit the data by performing proper cross validation.