Carine Candel, Deniz Ovalıoğlu and Isabel Walter
Machine Learning
Rein van den Boomgaard and Gosia Migut

Perception in dating

Problem Stating

For this project we are using a dataset called the Speed Dating Experiment, that was compiled by the professors Sheena Iyengar and Ray Fisman from the Columbia Business School. They acquired data from participants in speed dating events between 2002 and 2004. The participants went on multiple four minute speed dates with different partners each time. At the end of every date the participants were asked whether they would like to meet this partner again. Participants also rated themselves and the partner they met on six attributes: Attractiveness, Sincerity, Intelligence, Fun, Ambition and Shared Interests. Participants were also given a questionnaire on lifestyle, habits, demographics and other subjects, this is also included in the dataset. So the dataset holds a lot of diverse information that was all needed for the original experiment. Namely, the original experiment was designed to discover if race and gender preferences matter in dating. When we read about this dataset we came up with the following question:

'Does the difference between how you perceive yourself and how someone else perceives you have an influence on whether that someone wants to meet you again?'

Approach

For this project we do not use the whole dataset. We actually only use a small part of it. To answer our question, we highly limited the amount of data we are using from the dataset, since there is a significant number of attributes included in the original data, that are not relevant to our experiment so we excluded it and made our own adapted dataset. This adapted dataset consists only of the information on how participants rated themselves and their partner on different attributes. Note that we did not use all six attributes because almost no-one filled in their ratings of the Shared Interests attribute, so we decided to leave it out to omit huge patches of missing values. Furthermore, we took the feature whether for each date the partner wanted to meet again from the original dataset and added it to the adapted one. After this we took a good look at our dataset and found that there were quite some people who either did not fill in any rating about themselves or filled in only one of the five ratings about their partner. Since almost fifty percent of attribute values in an instance is then missing, we decided to mark these instances as invalid and left these instances out. Also, as we mention later we are using a decision tree algorithm from sklearn and this module does not support missing values. This means in the end we ended up with the following dataset:

The dataset consists of 6949 instances and each instance includes the following attributes/labels:

1. *How person 1 would rate him- or herself according to five attributes, on a scale from 1-10. Attributes are: Attractiveness, Sincerity, Intelligence, Fun, Ambition.*
2. *How person 2 rated person 1 according to the five attributes on a scale from 1-10.*
3. *Whether person 2 wants to meet person 1 again. Value is 1 if yes and 0 if no. This information defines our labels.*

We made the dataset in the form of a matrix with rows being the different instances and the columns being the labels + the different attributes (so 1 + 10 columns in total)

Now that we had the right adapted dataset, we needed to calculate the differences between how someone rated him- or herself and how their partner rated them. To calculate this we calculated the absolute differences between the matching columns and put these into new attribute columns. We also calculated the differences in ratio form to see how this would influence results.

So what classifiers did we decide to use? We decided to first of all use decision trees (using sklearn), because decision trees can be used as a very helpful classification algorithm for our classification problem. Also, interpretation of the output is very clear; we can actually visualize the tree and easily see what attributes are important, what attributes are not and what ratings lead to what results. Furthermore, with decision trees you need to do only little data preparations and keep the data as 'pure' as possible. For the decision tree algorithm we split the dataset in a 80% training set and a 20% test set and then find the best depth by 10-fold cross validation (using the training set). We also used 10-fold cross validation to plot the validation score against the percentage of data used.

After classification by decision trees we wanted to do unsupervised learning by implementing a k-means clustering algorithm from sklearn to see whether we would be able to obtain two remotely homogeneous clusters (so that there is a predominant label in the clusters), as our labels would suggest. Clustering does not make any prior assumptions about the data. So if the clusters are remotely homogeneous with predominantly instances of only one label per cluster , it means it is logical to split the data on those labels, based on the given attributes. In this case we would naturally impose a k-value of 2, since we only wanted two clusters. We used the whole adapted dataset for the clustering, so no splitting into training set and test set.

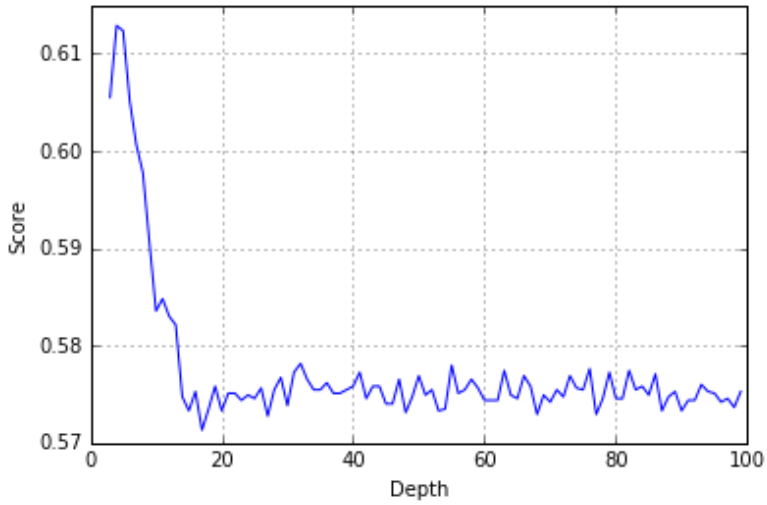
Choices made and justifications

1. We highly trimmed the original dataset to get rid of unnecessary attributes regarding finding an answer to our research question.
2. We removed about 1000 instances because of missing values (sklearn decision tree cannot handle missing values).
3. We use sklearn functions for all the classifying and the clustering and installed the pydotplus module for python (pydotplus is needed to visualise the decision tree).
4. We indicated the differences both by using absolute values of differences and by using ratios to see whether there are differences in the results (ratios could give 'smoother' results).
5. With using absolute differences we calculated the absolute values of how someone rates him- or herself minus how someone else rates that person.
6. With using ratio to indicate the differences we divided how someone rates you by how someone rates him- or herself, because in 'the how someone rates you' attribute people actually rated each other with a 0 and you cannot divide by 0.
7. We started searching for an optimum depth from a depth of 3 to 200. 3 because we do not want a too small decision tree. 200 is just a randomly picked large number to make sure that we did not miss the best depth.

Results

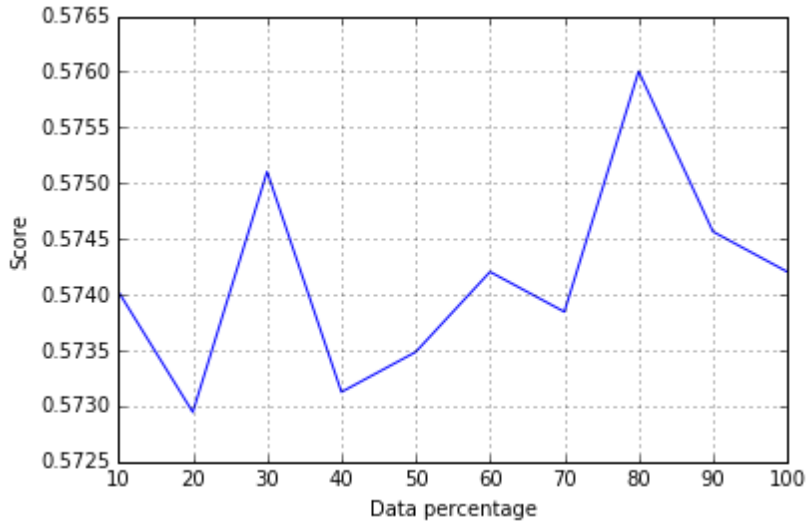
Decision tree classification (using absolute differences)

Learning curve for validation score vs depth:



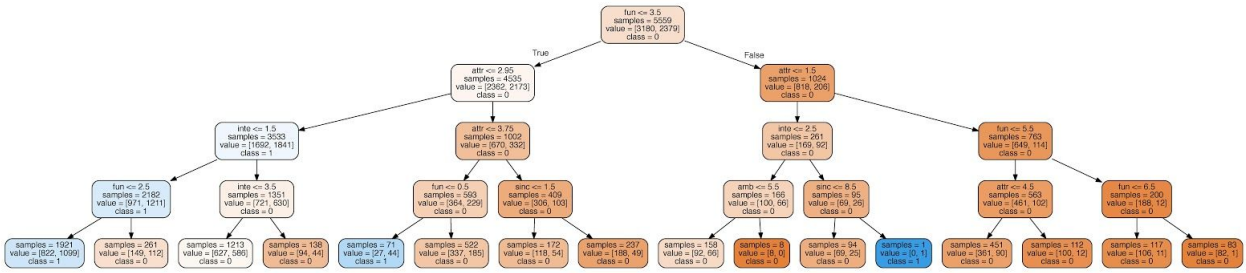
From this graph we can conclude that the best depth is at depth = 4 because then the cross-validation score is the highest. When greater depth values are used the score rapidly decreases.

Learning curve for validation score vs percentage of data used:



From this graph we can derive that the cross-validation score does not show a significant difference when different amounts of data is used,however, it seems to get a bit better in general when more data is used and the best cross-validation score is observed when only 80% of the data is used.

Visualised decision tree:



The difference in 'fun' rating is the most important indicator of whether someone wants to meet again. The darker the color of the node, the higher certainty the leaf has. A blue node indicates that someone would like to meet again, a orange nod indicates that someone would not like to meet again. In general someone is more likely to want to meet again if the difference in the scores they gave are small. As you can see at the leaves of the tree there are still a lot of very light colored nodes, so not a lot of certainty. This is because the depth value to get the best results out of new data is not large.

Accuracy: 0.594244604317

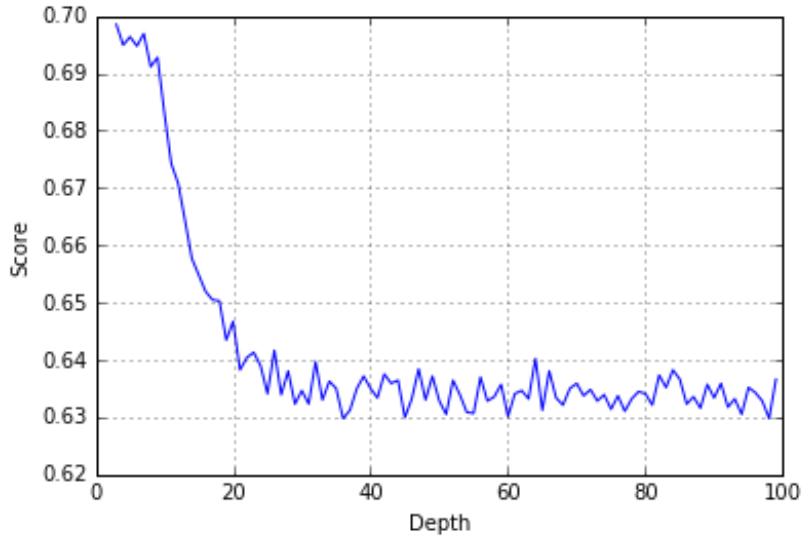
The accuracy is approximately 10% higher than 50%, therefore the classification is only a bit better than a random guess.

Confusion matrix:

Actual values\Predicted values	True	False
True	TP = 561	FN = 224
False	FP = 340	TN = 265

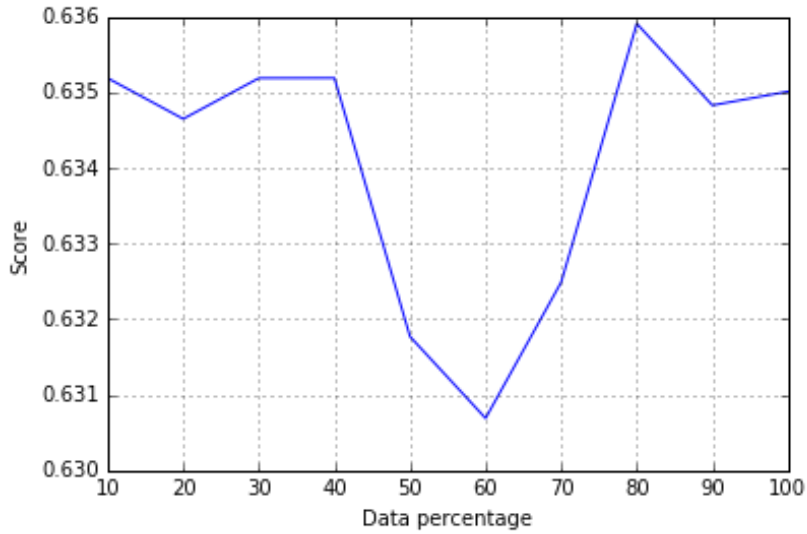
Decision tree classification (using ratio differences)

Learning curve for validation score vs depth:



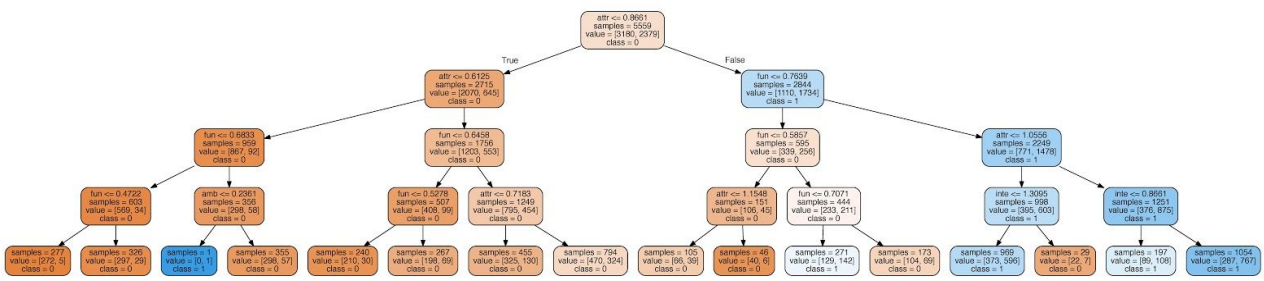
From this graph we can conclude that the best depth is at depth = 4 because then the cross-validation score is the highest. When greater depth values are used the score rapidly decreases.

Learning curve validation for score vs percentage of data used:



From this graph we can derive that the cross-validation score is highest when we use 80 percent of the data and lowest when we use 60 percent of the data. However, the fluctuations are in a very small interval from 0.63075 to 0.636.

Visualised decision tree:



The difference in 'attractiveness' rating is the most important indicator of whether someone wants to meet again. In general if the ratio is closer to 1 someone is more likely to want to meet again. In the tree based on the ratios, 'sincerity' does not play a role in the decision process and moreover, 'ambition' and 'intelligence' do not play a significant role as well. As you can see at the leaves of the tree there are still a lot of very light colored nodes, so not a lot of certainty. This is because the depth value to get the best results out of new data is not large.

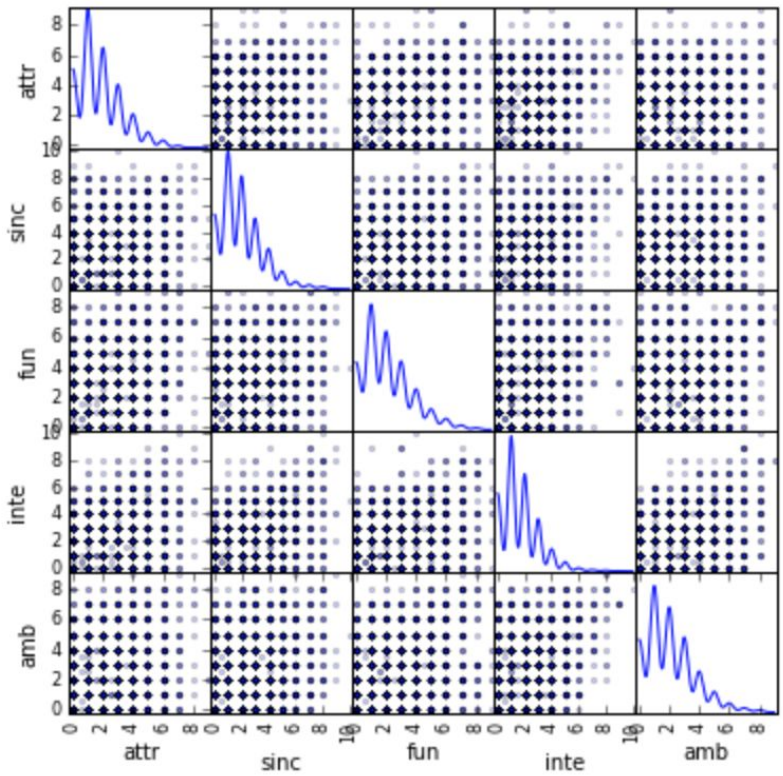
Accuracy: 0.689928057554
Accuracy is about 70 percent, which indicates it is a not completely random classification.

Confusion matrix:

Actual values\Predicted values	True	False
True	TP = 567	FN = 218
False	FP = 213	TN = 392

2-Means clustering (using absolute differences)

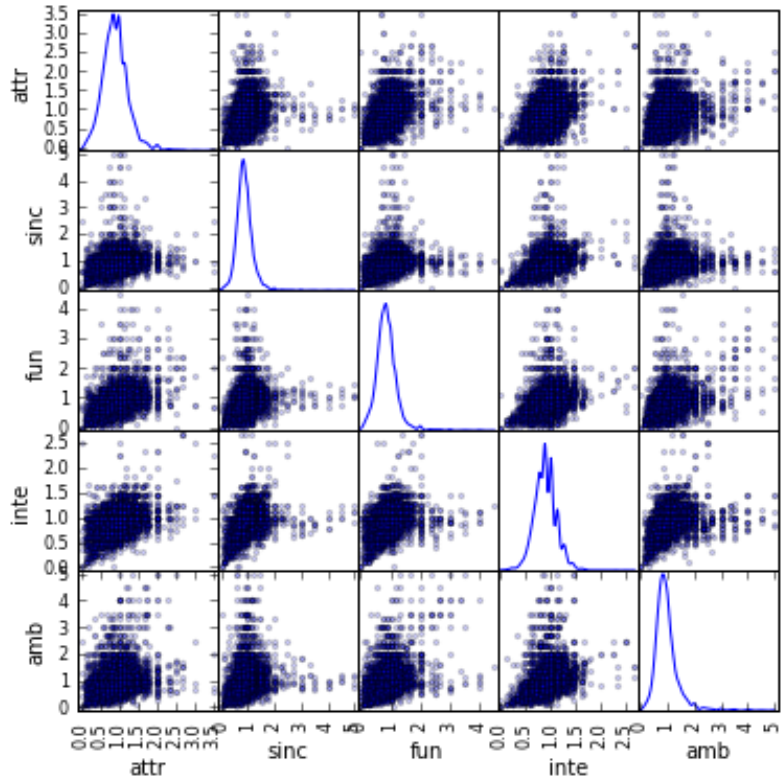
Visualised clustering:



Homogeneity score: 0.037
So the homogeneity score is very low, which means there is absolutely no predominant label in one cluster. Also in the graphs you can see that there are no nicely formed clusters and the distributions of the attributes are far from normally distributed.

2-Means clustering (using ratio differences)

Visualised clustering:



Homogeneity score: 0.038

Homogeneity is still very very low, so again absolutely no predominant label in the clusters. However, attributes are approximately normally distributed and the clustering looks quite nice.

Evaluation of results

For a better designed project we would need more attributes to answer the question properly, such that people rate each other on more than five attributes. Also, rather than taking the ratio of the ratings, the ratio could be calculated from both participants and then the average of the ratios could be used when the data is processed. The second approach would be less biased (in a social context) compared to the current approach where the ratio of other's rating to self rating is taken. To answer the question whether the difference in perception of someone's character has an influence on the decision if they want to meet again, only the "absolute difference" is important. In our approach of taking the ratio, the value you get would be different if you divided your own ratings with the others' ratings.

Conclusion

'Does the difference between how you perceive yourself and how someone else perceives you have an influence on whether that someone wants to meet you again?'

Based on our results the larger the difference between a person's perceptions of you and how you perceive yourself, the lower the chance that you will want to meet again. With accuracy values 0.59 and 0.69 it is clear that the decision is not totally random because it shows correlation

with the differences in ratings. However, there does not seem to be a very strong correlation between differences in ratings and the decision of meeting again.

Comparing the two different approaches to calculate the difference in ratings, the one where we used ratios is clearly more accurate compared to the approach where we used absolute values. This might be the case because in the ratio approach less data is discarded compared to taking the absolute values of subtractions. For instance, when the absolute value is taken, -3 is converted to 3 which means it is not important anymore if you rated yourself higher than the other or vice versa, while by using ratios this aspect is still taken into account.

Also worth mentioning is that the decision tree algorithm gives a significant lower score on the cross-validation set than on the test set. This could be caused by differences in size of the two sets, as the cv set consists of only 10 percent of 80 percent of the data and the test set consists of 20 percent of the data.

Comparing the different machine learning algorithms, clustering and decision trees, clustering does not give a significant result, because it gives no clusters with one predominant label. The homogeneity scores of the clusters are therefore very low, which suggest that it is not logical to split the data in 0's and 1's based on these attributes. However, the decision tree algorithm gives accuracies that are significantly above 50% suggesting that you actually can predict it from the attributes. Still, these percentages are not very high, which could mean that this dataset is just not very appropriate for predicting the answer to our question.

Afterall, if you want a second date: "Ya olduğun gibi görün, ya da göründüğün gibi ol" (Mevlana) meaning either seem as you are or be as you seem...

Appendix

- The code can be found in Deniz Ovalıoğlu's *github* account.
- PDF's of the visualized decision trees (for absolute differences and for ratio differences) can also be found there.
- The dataset we used in the code can be found there too.