UNIVERSITY OF LEICESTER

Computational Task 1

For Data Mining and Neural Networks MA4022/ MA7022

Gleb Vorobchuk

Leicester

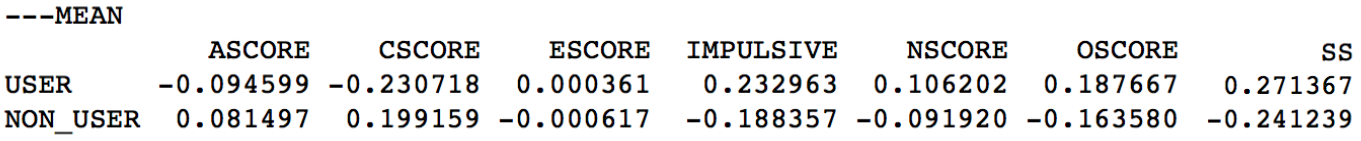
February 2018

Task №1.

Descriptive statistics: For both classes (users and non-users) find the mean values of the 7 attributes and their standard deviations. Evaluate the 95% confidence intervals for mean values.

For both classes and all attributes I found the mean values as[1]:

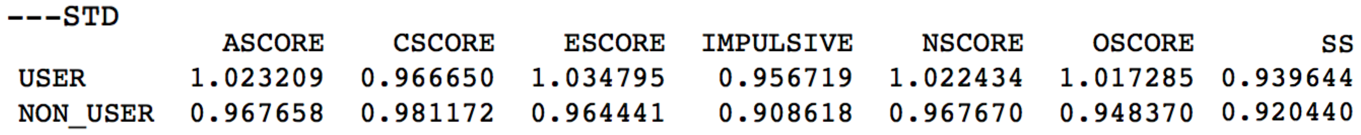
$$\bar{x} = \frac{1}{n}\left(\sum_{i=1}^{n} x_i\right) = \frac{x_1 + x_2 + \cdots + x_n}{n}$$



```
---MEAN
            ASCORE    CSCORE    ESCORE  IMPULSIVE    NSCORE    OSCORE        SS
USER     -0.094599 -0.230718  0.000361   0.232963  0.106202  0.187667  0.271367
NON_USER  0.081497  0.199159 -0.000617  -0.188357 -0.091920 -0.163580 -0.241239
```

Then I found Standard Deviations as[2] :

$$s = \sqrt{\frac{\sum_{i=1}^{N}(x_i - \overline{x})^2}{N-1}}.$$



```
---STD
            ASCORE    CSCORE    ESCORE  IMPULSIVE    NSCORE    OSCORE        SS
 USER     1.023209  0.966650  1.034795   0.956719  1.022434  1.017285  0.939644
 NON_USER 0.967658  0.981172  0.964441   0.908618  0.967670  0.948370  0.920440
```

So now we can define 95% confidence intervals for the mean. The interval estimate gives an indication of how much uncertainty there is in our estimate of the true mean. The narrower the interval, the more precise is our estimate.[3]

$$\bar{Y} = \pm z^{*}\frac{s}{\sqrt{N}}$$

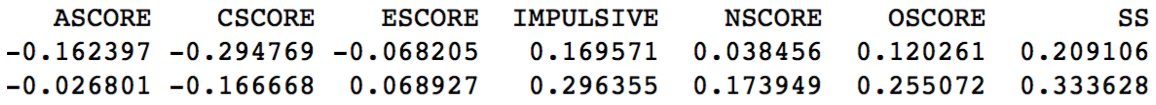
[1] https://en.wikipedia.org/wiki/Mean

[2] https://en.wikipedia.org/wiki/Standard_deviation

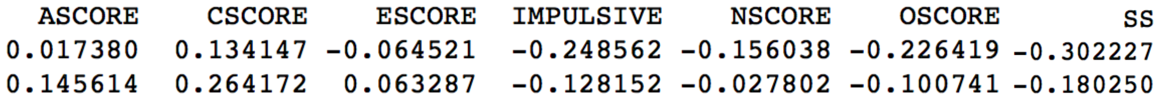
[3] http://www.itl.nist.gov/div898/handbook/eda/section3/eda352.htm

Where $\bar{Y}$ - is the sample mean, s - is the sample standard deviation and z^* value fo 95% interval is taken from z-distribution an its equal to 1.96.

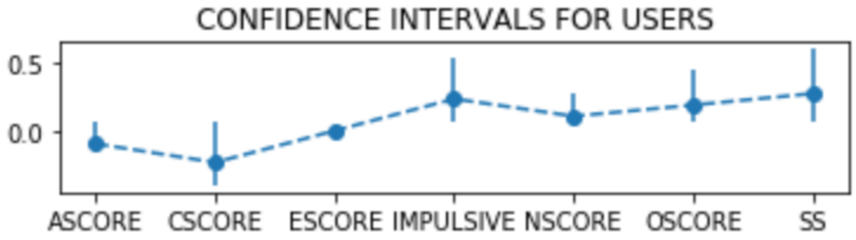
Here is the confidence intervals for users and non-users correspondingly:

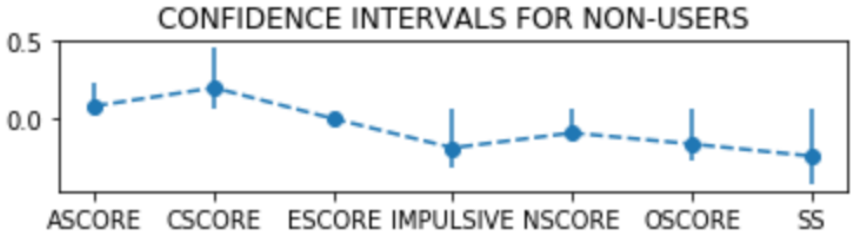


```
  ASCORE    CSCORE    ESCORE  IMPULSIVE    NSCORE    OSCORE        SS
-0.162397 -0.294769 -0.068205  0.169571  0.038456  0.120261  0.209106
-0.026801 -0.166668  0.068927  0.296355  0.173949  0.255072  0.333628
```



```
  ASCORE    CSCORE    ESCORE  IMPULSIVE    NSCORE    OSCORE         SS
0.017380  0.134147 -0.064521 -0.248562 -0.156038 -0.226419 -0.302227
0.145614  0.264172  0.063287 -0.128152 -0.027802 -0.100741 -0.180250
```





Psychological profiles most of the people in average are:

USERS	Less Agreeable	Less Conscientious	More Extravert	More Impulsive	More Neurotic	More Opened to experience	More Sensation-Seeking
NON-USERS	More Agreeable	More Conscientious	Less Extravert	Less Impulsive	Less Neurotic	Less Opened to experience	Less Sensation-Seeking

Task №2.

Report, which differences between these means for users and non-users are significant. For significance evaluation use p-values.

In order to do that we need to perform t-test for means of two independent samples from descriptive statistics.

The t statistic to test whether the means are different can be calculated as follows[wiki]:

$$t = \frac{\bar{X}_1 - \bar{X}_2}{s_{\bar{\Delta}}}$$

Where:

$$s_{\bar{\Delta}} = \sqrt{\frac{s_1^2}{n_1} + \frac{s_2^2}{n_2}}.$$

And with degrees of freedom calculated as:

$$\mathrm{d.f.} = \frac{\left(\frac{s_1^2}{n_1} + \frac{s_2^2}{n_2}\right)^2}{\frac{\left(s_1^2/n_1\right)^2}{n_1-1} + \frac{\left(s_2^2/n_2\right)^2}{n_2-1}}.$$

Usually, the difference between two groups to determine if it is statistically significant. The difference between two groups is statistically significant if it can not be explained by chance alone.

Usually, statistical significance is determined by calculating the probability of error (p value) by the t ratio.

The difference between two groups (such as an experiment vs. control group) is judged to be statistically significant when $p = 0.05$ or less.

At p= 0.05, the differences between the two groups have only a 5% probability of occurring by chance alone.[4]

[4] http://www.mentalhealth.com/dis-rs/rs-effect_size.html

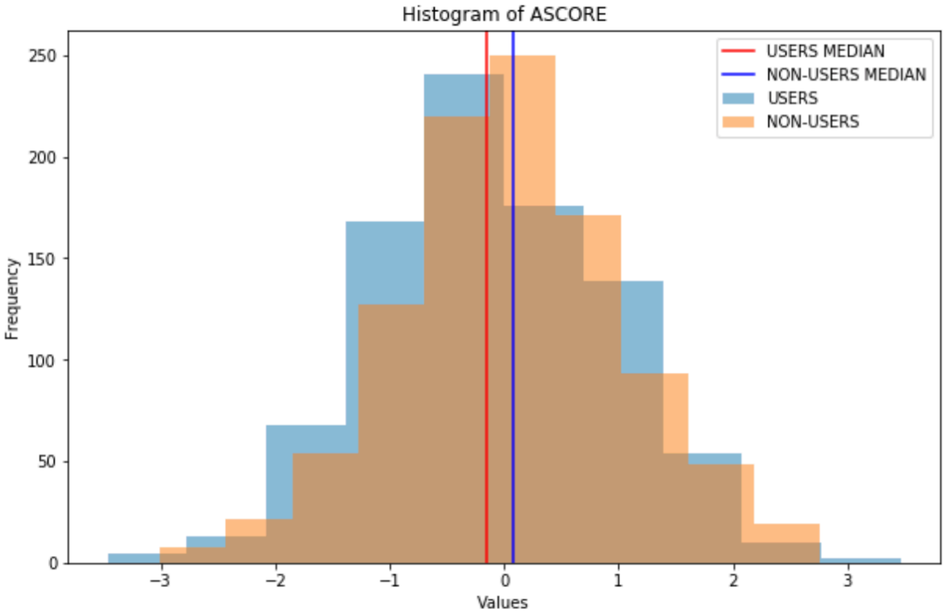
So comparing p-values with 0.05 for all the attributes we have following results:

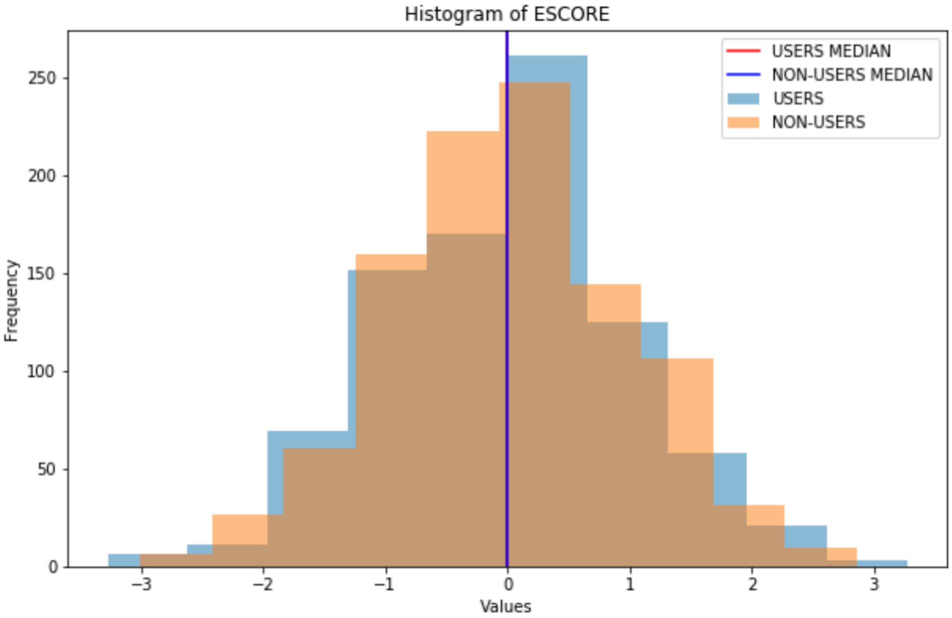
	p-value:	Difference:
ASCORE	0.000223368585288	Is not significant
CSCORE	7.43274657151E-20	Is not significant
ESCORE	0.983689025058	Is significant
IMPULSIVE	1.09274861731E-20	Is not significant
NSCORE	3.29356549511E-05	Is not significant
OSCORE	1.25669047954E-13	Is not significant
SS	1.09839780078E-29	Is not significant

Only differences in means for ESCORE are significant.

Task №3

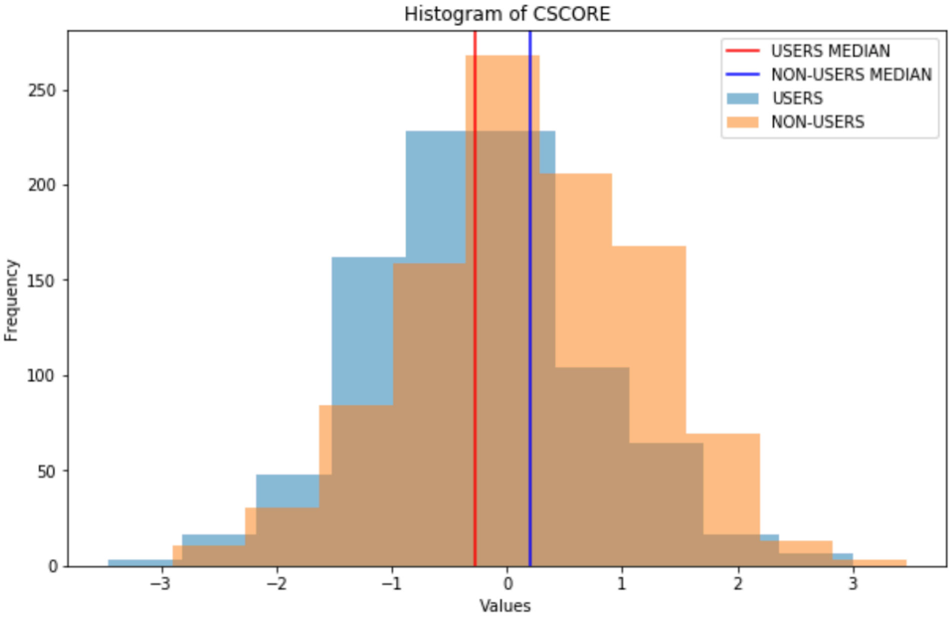
Try to create predictors user/non-user by one attribute (7 such predictors). For this purpose, create histograms for each attribute and each class and select the best threshold for each attribute x for the decision rule: if x>a then one class (users or non-users) and if x<a then

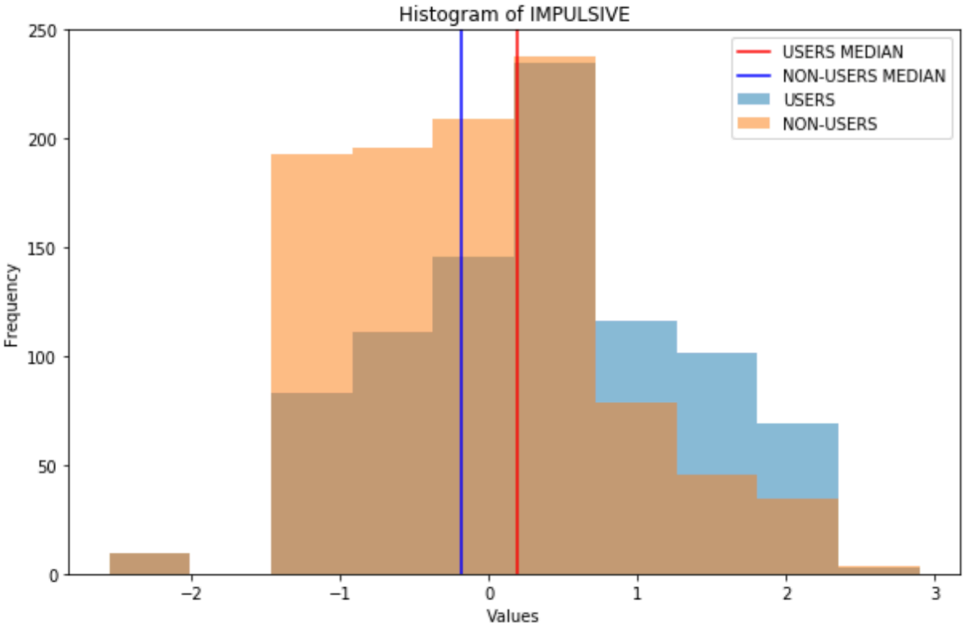




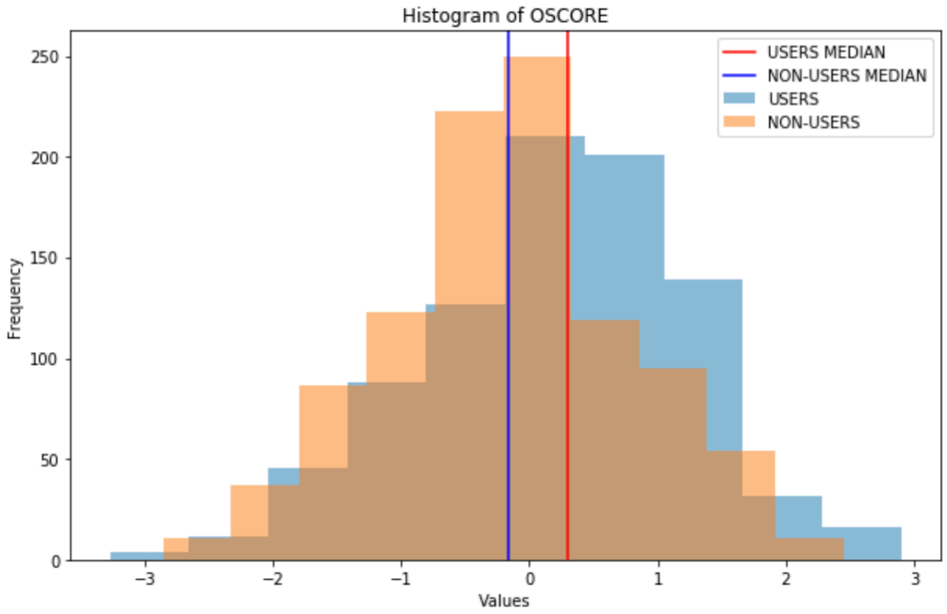
another

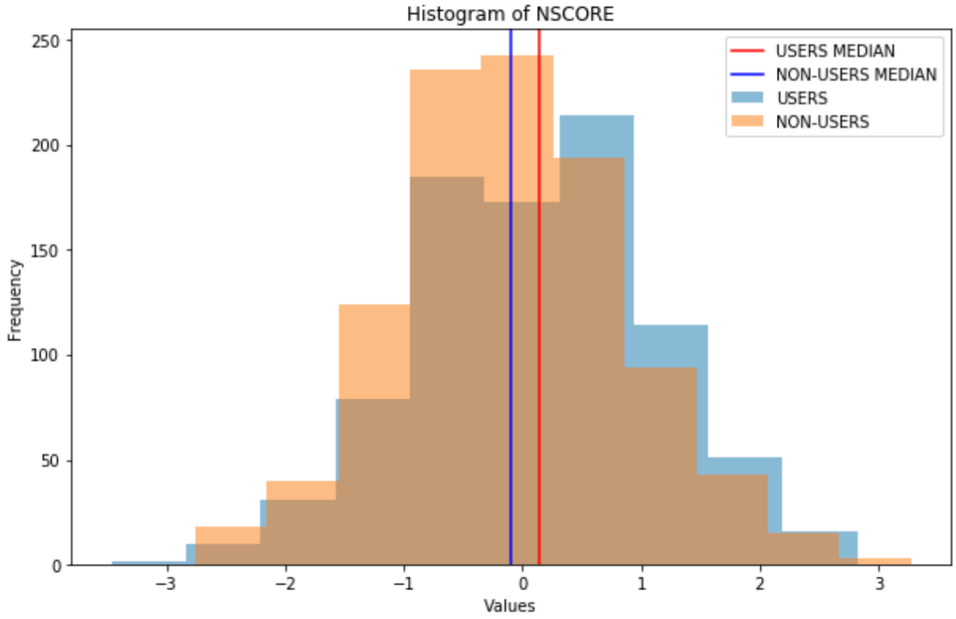
class (non-users or users) (the optimal cut). Find the classification error for each attribute. Which attribute gives the best prediction? Arrange the attributes in their prediction ability.

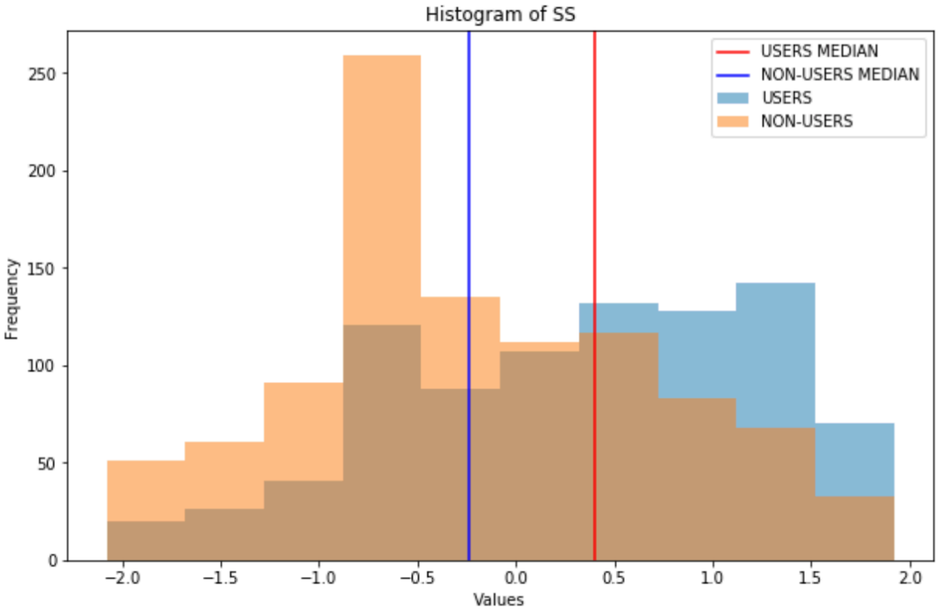




Here are Histograms of each attribute:


Histogram of OSCORE
USERS MEDIAN
NON-USERS MEDIAN
USERS
NON-USERS
Frequency
0
50
100
150
200
250
−3
−2
−1
0
1
2
3
Values

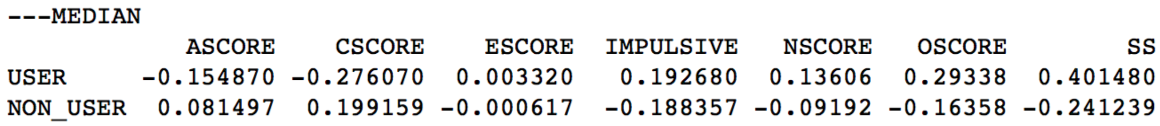

Histogram of NSCORE
USERS MEDIAN
NON-USERS MEDIAN
USERS
NON-USERS
Frequency
0
50
100
150
200
250
−3
−2
−1
0
1
2
3
Values



Now we need to define threshold for the classification rule.

In order to do that I decided to take an intervals between median values of each attribute with 1000 values in them.

First of all we need to find Medinas for all the attributes:



```
---MEDIAN
             ASCORE    CSCORE    ESCORE  IMPULSIVE   NSCORE   OSCORE        SS
USER      -0.154870 -0.276070  0.003320   0.192680  0.13606  0.29338  0.401480
NON_USER   0.081497  0.199159 -0.000617  -0.188357 -0.09192 -0.16358 -0.241239
```

After that we setting the classification rule for each attribute. As we can see for ASCORE and CSCORE User median lays on the left and Non-User on the right.

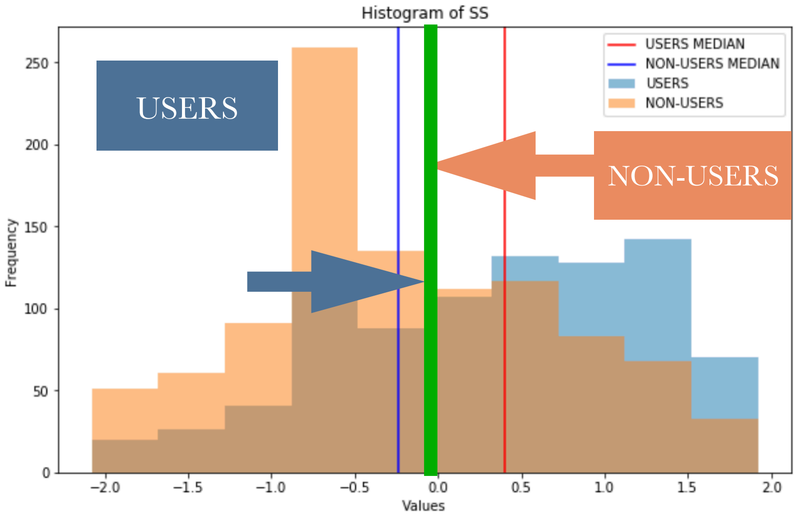
For OSCORE, IMPULSIVE, SS and NSCORE User median lays on the right and Non-User median lays on the left.

For ESCORE median values are close but as we can see a majority of values for User lay on the right and for Non-User on the left.

The idea is that if the values of the attribute lie to the right/left, then we classify the person as the user/non-user. Depending on the median of attribute, the classification rule will change.

Varying threshold through the calculations will help us find minimal error.

We are using separate data for Users and Non-users to calculate mispredictions.



For example:

Figure shows histograms and Median values of SS score for both classes.

Let Green line be a threshold. Then we assume two rules for both classes:

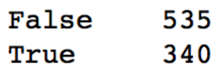
If SS score of a person is greater than threshold - we classify this person as user.

If SS score of a person is less than or equal to threshold - we classify this person as non-user.

Here we have a table of all the rules:

	USER	NON-USER
ASCORE THRESHOLD	Less than	Greater than or equal
CSCORE THRESHOLD	Less than	Greater than or equal
ESCORE THRESHOLD	Greater than or equal	Less than
IMPULSIVE THRESHOLD	Greater than	Less than or equal
NSCORE THRESHOLD	Greater than	Less than or equal
OSCORE THRESHOLD	Greater than	Less than or equal
SS THRESHOLD	Greater than	Less than or equal

Then we count how many users misclassified in user dataset and how many non-users misclassified in non-users dataset:



```
False     535
True      340
```

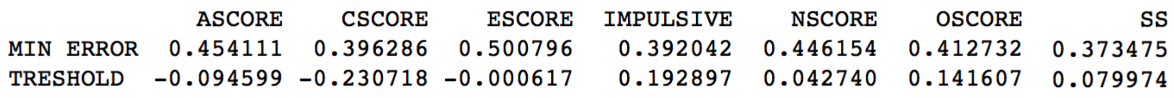
After that we assume that error of classification is:

$$E = \frac{False_{User} + False_{Non-user}}{All_{Users} + All_{Non-users}}$$

Where All_{Users} and $All_{Non-users}$ is the lengths of corresponding datasets.

Then we need to find thresholds for all attributes where error is minimised.

So we have following minimal errors with corresponding thresholds :



```
              ASCORE    CSCORE    ESCORE  IMPULSIVE    NSCORE    OSCORE        SS
MIN ERROR   0.454111  0.396286  0.500796   0.392042  0.446154  0.412732  0.373475
TRESHOLD   -0.094599 -0.230718 -0.000617   0.192897  0.042740  0.141607  0.079974
```

As we can see SS score gives the best prediction at threshold 0.079974 with minimal error 0.373475

Here is the arrangement of attributes in their prediction ability:

ATTRIBUTE	SS THRESHOLD	IMPULSIVE THRESHOLD	CSCORE THRESHOLD	OSCORE THRESHOLD	NSCORE THRESHOLD	ASCORE THRESHOLD	ESCORE THRESHOLD
ERROR	0.37347	0.39204	0.39628	0.41273	0.44615	0.45411	0.50079

Task №4.

Test 1NN and 3NN classification rules. Present the classification errors. Which rule is better?

The KNN algorithm is a robust and versatile classifier that is often used as a benchmark for more complex classifiers such as Artificial Neural Networks (ANN) and Support Vector Machines (SVM). Despite its simplicity, KNN can outperform more powerful classifiers and is used in a variety of applications such as economic forecasting, data compression and genetics. For example, KNN was leveraged in a 2006 study of functional genomics for the assignment of genes based on their expression profiles.[5]

In the classification setting, the K-nearest neighbour algorithm essentially boils down to forming a majority vote between the K most similar instances to a given "unseen" observation. Similarity is defined according to a distance metric between two data points. A popular choice is the Euclidean distance given by:

$$d(x, x') = \sqrt{(x_1 - x_1')^2 + (x_2 - x_2')^2 + \ldots + (x_n - x_n')^2}$$

More formally, given a positive integer K, an unseen observation *x* and a similarity metric *d*, KNN classifier performs the following two steps:

It runs through the whole dataset computing d between x and each training observation. We'll call the K points in the training data that are closest to *x* the set *A*.

It then estimates the conditional probability for each class, that is, the fraction of points in *A* with that given class label.[6]

$$P(y = j | X = x) = \frac{1}{K} \sum_{i \in A} I(y^i = j)$$

In order to avoid overfitting - the process when is incapable to predict new observations we will use the method of k-fold cross-validation.

In a nutshell we are going to divide original dataset into k equal sized subsamples.

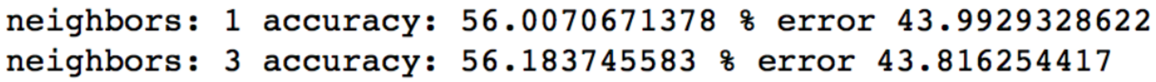
Then we will use first subset as validation set for model testing and remaining k-1 subsets as training sets. We perform this process k-times and each one of the subsamples will be used once for validation. Usually 10-folds cross-validation is used. Let' us perform 1-NN and 3-NN prediction with 10 fold cross-validation.

[5] https://kevinzakka.github.io/2016/07/13/k-nearest-neighbor/

[6] https://kevinzakka.github.io/2016/07/13/k-nearest-neighbor/

It seems that ESCORE is not relevant attribute because difference in mean values is small. And as we know KNN is sensitive to irrelevant features, I've decided to remove this column from classification.

Results of 1-NN and 3-NN prediction with 10-fold cross-validation:

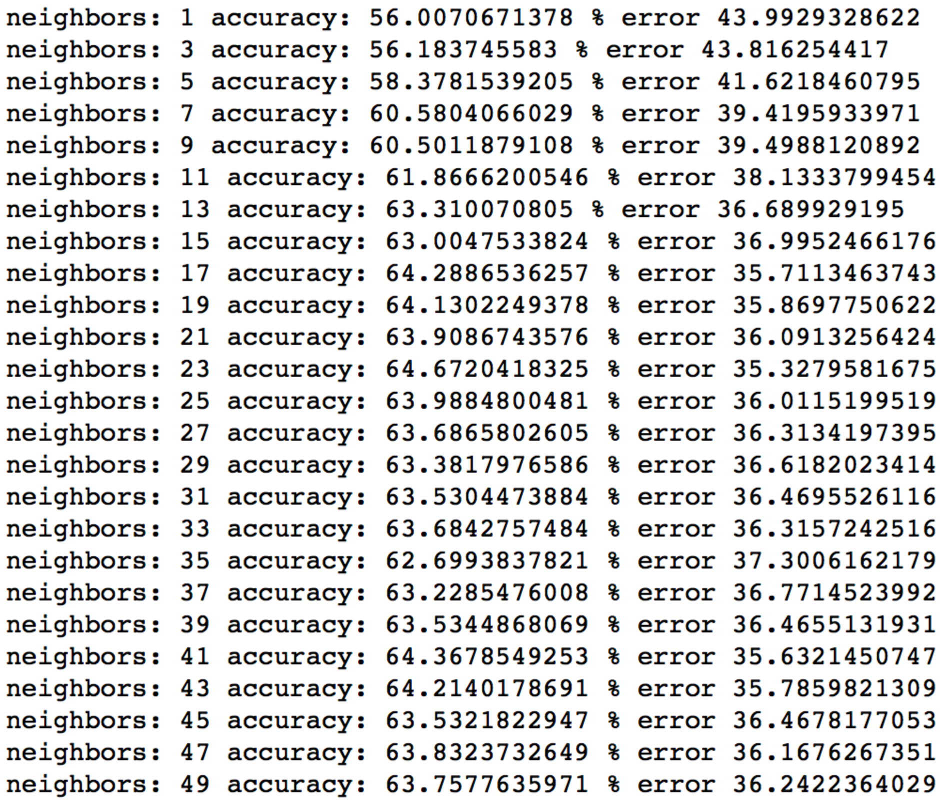


```
neighbors: 1 accuracy: 56.0070671378 % error 43.9929328622
neighbors: 3 accuracy: 56.183745583 % error 43.816254417
```

As we can see 3-NN rule gives better prediction accuracy, but 1-NN rule is easier to perform.

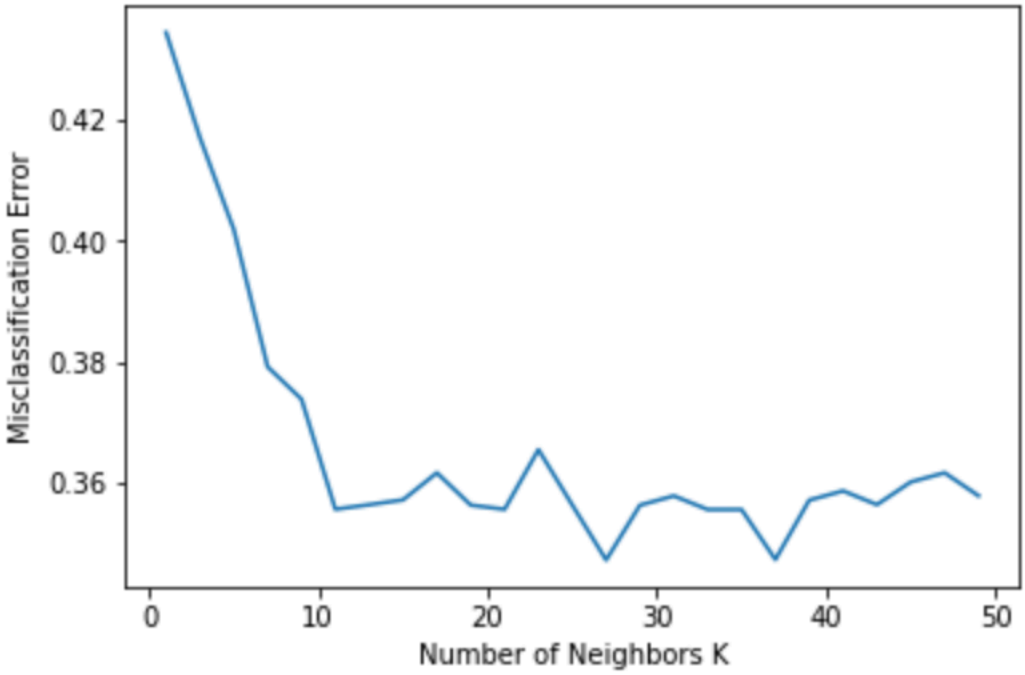
At this point we have difference in accuracy less than 2% and we can't say that one's better.

Variation of K over the calculation will help us find the optimal number of neighbours for this problem.



```
neighbors: 1 accuracy: 56.0070671378 % error 43.9929328622
neighbors: 3 accuracy: 56.183745583 % error 43.816254417
neighbors: 5 accuracy: 58.3781539205 % error 41.6218460795
neighbors: 7 accuracy: 60.5804066029 % error 39.4195933971
neighbors: 9 accuracy: 60.5011879108 % error 39.4988120892
neighbors: 11 accuracy: 61.8666200546 % error 38.1333799454
neighbors: 13 accuracy: 63.310070805 % error 36.689929195
neighbors: 15 accuracy: 63.0047533824 % error 36.9952466176
neighbors: 17 accuracy: 64.2886536257 % error 35.7113463743
neighbors: 19 accuracy: 64.1302249378 % error 35.8697750622
neighbors: 21 accuracy: 63.9086743576 % error 36.0913256424
neighbors: 23 accuracy: 64.6720418325 % error 35.3279581675
neighbors: 25 accuracy: 63.9884800481 % error 36.0115199519
neighbors: 27 accuracy: 63.6865802605 % error 36.3134197395
neighbors: 29 accuracy: 63.3817976586 % error 36.6182023414
neighbors: 31 accuracy: 63.5304473884 % error 36.4695526116
neighbors: 33 accuracy: 63.6842757484 % error 36.3157242516
neighbors: 35 accuracy: 62.6993837821 % error 37.3006162179
neighbors: 37 accuracy: 63.2285476008 % error 36.7714523992
neighbors: 39 accuracy: 63.5344868069 % error 36.4655131931
neighbors: 41 accuracy: 64.3678549253 % error 35.6321450747
neighbors: 43 accuracy: 64.2140178691 % error 35.7859821309
neighbors: 45 accuracy: 63.5321822947 % error 36.4678177053
neighbors: 47 accuracy: 63.8323732649 % error 36.1676267351
neighbors: 49 accuracy: 63.7577635971 % error 36.2422364029
```

Let's plot the misclassification error against number of neighbours:



As we can see fo this particular problem optimal number of neighbours is 27.

Task №5.

Find in the literature description and explanation of Fisher's linear discriminant. Read, understand and write a comprehensive description of the algorithm with main formulas and explanation.

Linear discriminant analysis (**LDA**) is a generalisation of **Fisher's linear discriminant**, a method used in statistics, pattern recognition and machine learning to find a linear combination of features that characterises or separates two or more classes of objects or events. The resulting combination may be used as a linear classifier, or, more commonly, for dimensionality reduction before later classification.[7]

The main idea of Fisher's method is that we projecting data from d-dimensions onto a line.

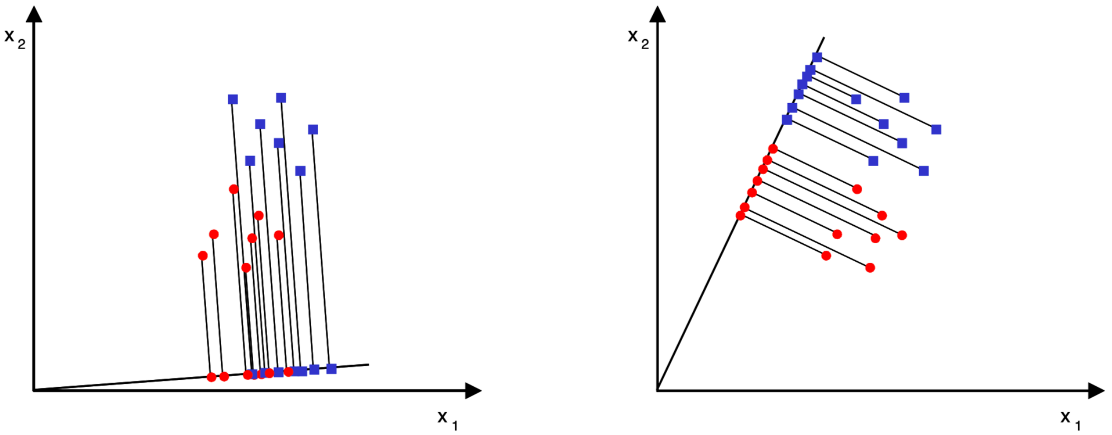
Assume the set D-dimensional samples $\{x^{(1}, x^{(2}, \ldots, x^{(N}\}$

N_1 belongs to class ω_1, and $N_2$to class ω_2.

We seek to obtain a scalar y by projecting the samples x onto a line

$$y = w^T x.$$

Among all the possible lines we need to find one that maximises the separability of the scalars.



On the left figure we see bad projection with mixed classes. On the right figure we see good projection where classes are separated.

In order to find a good projection vector, we need to define a measure of separation

The mean vector of each class in x-space and y-space is:

$$\mu_i = \frac{1}{N_i}\sum_{x\in\omega_i} x, \text{ and } \bar{\mu}_i = \frac{1}{N_i}\sum_{y\in\omega_i} y = \sum_{x\in\omega_i} w^T x = w^T\mu_i, \ i = 0{,}1$$

[7] https://en.wikipedia.org/wiki/Linear_discriminant_analysis#cite_note-Fisher:1936-1

Fisher suggested to use the separation between these two distributions to be the ratio of the variance between the classes to the variance within the classes.

For each class we define the scatter such that:

$$\hat{s}_i^2 = \sum_{y \in \omega_i} (y - \hat{\mu}_i)^2.$$

Then the Fisher's LDA we can define as the linear function $w^T x$ that maximises the function:

$$J(w) = \frac{|\hat{\mu}_0 - \hat{\mu}_1|^2}{\hat{s}_1^2 + \hat{s}_2^2}.$$

$S_W = \hat{s}_1^2 + \hat{s}_2^2$ - is Within-Class Scatter Matrix.

$S_B = (\hat{\mu}_0 - \hat{\mu}_1)(\hat{\mu}_0 - \hat{\mu}_1)^T$. - is Between-Class Scatter Matrix.

S_B measures separation between the means of two classes.

Now we can rewrite Fisher's LDA as:

$$J(w) = \frac{w^T S_B w}{w^T S_W w}$$

i.e generalised Rayleigh quotient[8][9].

Vector w that maximises $J(w)$ should satisfy:

$$S_B w = \lambda S_W w.$$

If S_w has full rank we can rewrite it as a generalised eigenvalue problem:

$$S_W^{-1} S_B w = \lambda w$$

But $S_B w$ is always in direction of $\hat{\mu}_0 - \hat{\mu}_1$ so we can solve eigenvalue immediately:

$$w = S_W^{-1}(\hat{\mu}_0 - \hat{\mu}_1)$$

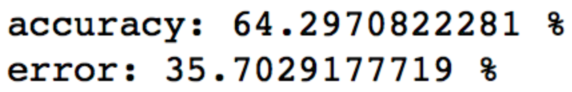
[8] http://www.cedar.buffalo.edu/~srihari/CSE555/Chap3.Part6.pdf

[9] https://en.wikipedia.org/wiki/Rayleigh_quotient

Task №6.

Apply Fisher's linear discriminant to the prepared data set. Analyse the quality of classification. Compare to 1NN and 3NN methods.

After application of Fisher's Linear Discriminant on data we have following results:



```
accuracy: 64.2970822281 %
error: 35.7029177719 %
```

Now we compare accuracy of Fisher's linear discriminant, 1-NN and 3-NN rules accuracy.

	Accuracy	Error
Fisher's LDA	64.297%	35.712%
3-NN	56.18%	43.81%
1-NN	56.01%	43.99%

As we can see Fisher's Linear Discriminant gives us better accuracy.

References.

1. https://en.wikipedia.org/wiki/Standard_deviation
2. https://en.wikipedia.org/wiki/Mean
3. http://www.itl.nist.gov/div898/handbook/eda/section3/eda352.htm
4. http://www.mentalhealth.com/dis-rs/rs-effect_size.html
5. https://kevinzakka.github.io/2016/07/13/k-nearest-neighbor/
6. https://kevinzakka.github.io/2016/07/13/k-nearest-neighbor/
7. https://en.wikipedia.org/wiki/Linear_discriminant_analysis#cite_note-Fisher:1936-1
8. http://www.cedar.buffalo.edu/~srihari/CSE555/Chap3.Part6.pdf
9. https://en.wikipedia.org/wiki/Rayleigh_quotient

Appendix.

Source code:

You can find source code attached and submitted with report in file Computational Task 1.ipynb.