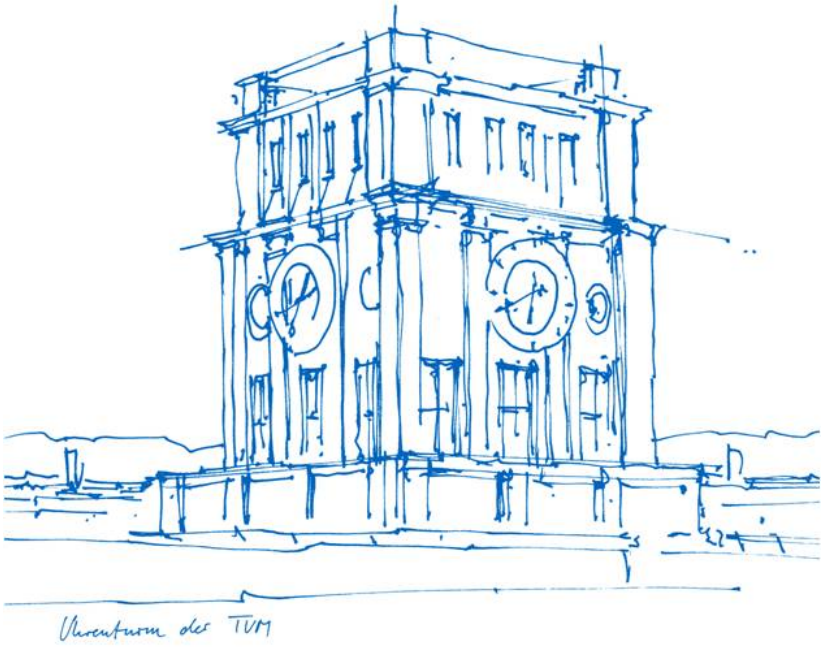
Exercises for Social Gaming and Social Computing (IN2241 + IN0040) – Introduction to Exercise 4



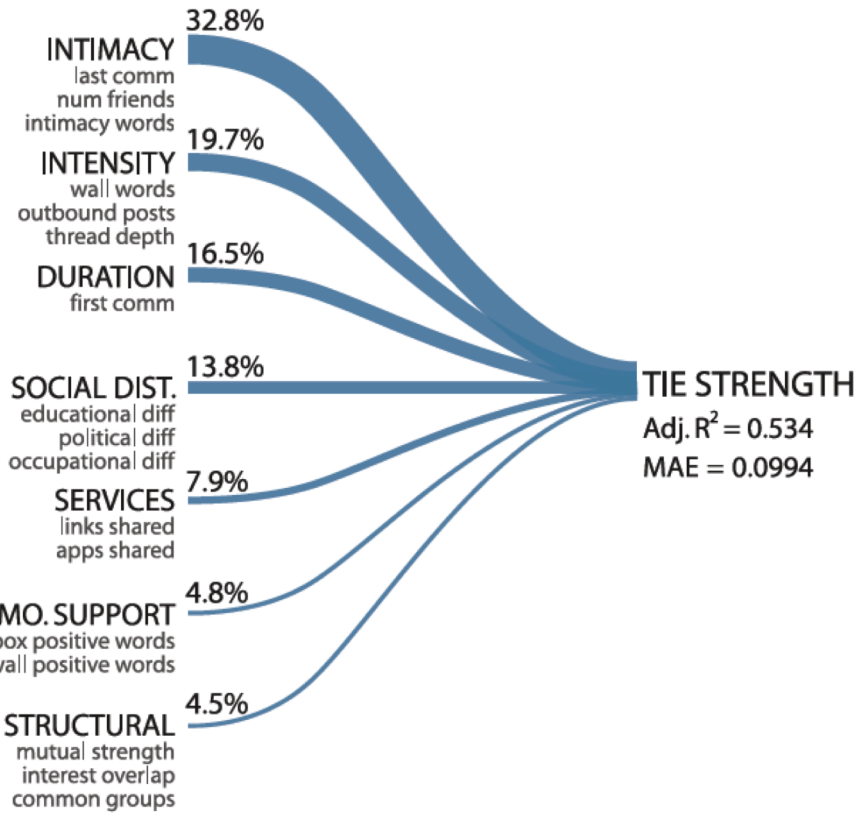
Exercise Content

Sheet Number	Exercise	Working Time
1	• Introduction to Python and Network Visualization	May 24-30
2	• Centrality Measures	May 31 – June 6
3	• Finding Groups & Clustering Methods	June 7 – 13
4	• Predicting Social Tie Strength with Linear Regression	June 14 - 20
5	• Natural Language Processing: Hate Speech detection + Social Context Influence	June 21 - 27
6	• Natural Language Processing: Modern Machine Learning Methods and Explainable AI	June 28 – July 4

Exercise Sheet 4: Predicting Tie Strength

- Goals:
  - Being able to use linear regression for predicting one social context variable from other social context variable
  - Getting to know a classic social computing paper
- Data:
  - An anonymized dataset from a social networking site

Paper on which exercise is based: „Predicting Tie Strength with Social Media“ (Gilbert and Karahalios) [1]



Top 15 Predictive Variables	β	F	p-value
Days since last communication	-0.76	453	< 0.001
Days since first communication	0.755	7.55	< 0.001
Intimacy × Structural	0.4	12.37	< 0.001
Wall words exchanged	0.299	11.51	< 0.001
Mean strength of mutual friends	0.257	188.2	< 0.001
Educational difference	-0.22	29.72	< 0.001
Structural × Structural	0.195	12.41	< 0.001
Reciprocal Serv. × Reciprocal Serv.	-0.19	14.4	< 0.001
Participant-initiated wall posts	0.146	119.7	< 0.001
Inbox thread depth	-0.14	1.09	0.29
Participant's number of friends	-0.14	30.34	< 0.001
Inbox positive emotion words	0.135	3.64	0.05
Social Distance × Structural	0.13	34	< 0.001
Participant's number of apps	-0.12	2.32	0.12
Wall intimacy words	0.111	18.15	< 0.001

Predicting Tie Strength

- input („predictive“) variables x:

1) Number of friends 2) Friends' number of friends 3) Days since last communication 4) Shared appearances in photos 5) Wall intimacy words 6) Inbox intimacy words	Intimacy variables
7) Days since first communication	Duration variables
8) Number of mutual friends	Structural variables
9) Age distance 10)Educational distance	Social distance variables

- output („dependent“) variable y:

tie strength

Predicting Tie Strength

- model: (linear) linear regression:

$$y_i = \alpha + \beta \boldsymbol{X}_i + \epsilon_i$$

- Recommended reading: C. Bishop: *Pattern Recognition and Machine Learning*. 2006, chapter 3 [2] on Moodle

- dataset: SocialGraph.gml with
  - input data X (incomplete → please complete ☺) for each edge and
  - ground truth tie strength y (for 0.7 of the edges → predict rest 0.3 of edges)

Tasks:

Task 4.1: Preparations

a) Imports and Visualization

First, needed libraries and the graph's .gml file have to be imported. The social graph is visualized in order to get an idea what the network actually looks like. Inspect the plotted graph. **Describe** shortly, what the graph's visualization is telling you, and if there are any problems with this representation. **Any ideas** on how to improve the visualization?

n [ ]:

```
import networkx as nx, numpy as np, pandas as pd, statsmodels.api as sm, matplotlib.pyplot as plt

# read in the structure
g = nx.read_gml('SocialGraph.gml', label='id')


# formatting the graph and applying spring layout
fig=plt.figure(figsize=(18, 16))

pos=nx.spring_layout(g, k=0.4, iterations=5)

visual_style = {
    "node_size": 300,
    "node_color": "#4089EF",
    "bbox" : (700,700),
    "with_labels" : False
}

nx.draw(g, pos, **visual_style)

```

TODO: Write your observations and ideas here

Tasks:(cont.)

b) Complete and convert the data

To further work with our data set, we will now convert it to a Pandas [4] dataframe. Some of our predictive variables are not yet computed in the *gml* file, therefore you have to **calculate the missing variables** from the graph's attributes. You can take a look at the *gml* file as it is human-readable to see what variables are available for you.

```
    # TODO: Compute the missing values
    age_dist = #TODO
    edu_diff = #TODO
    num_friends = #TODO
    friends_num_friends = #TODO
    num_mutual_friends = #TODO

```

c) The Variance Inflation Factor (VIF)

Multiple linear regression can hold some pitfalls if you do not evaluate your data beforehands. Such a pitfall is containing multicollinearity in your predictive variables.

Find out and **explain** in your own words what multicollinearity is, why it forms a danger to linear regression models and how the VIF is linked to that. **Create** a temporary dataframe containing only the predictive variables and **add a constant value** to the dataframe for the VIF to produce representative values. Then **compute the VIFs** for them. Statsmodels `variance_inflation_factor()` and `add_constant()` will help you with that.

Additionally **explain**: What do the results tell you? Do we have to make any adaptions deriving from them?

```
from statsmodels.stats.outliers_influence import variance_inflation_factor
from statsmodels.api import add_constant


# TODO: Creat a dataframe, add a constant & compute VIF


```

TODO: Write your explanaitions here

Tasks:(cont.)

d) Normalisation-Transformation

Normalization is a technique often applied as part of data preparation for machine learning. The goal of normalisation is to change the values of numeric columns in the dataset to a common scale, without distorting differences in the ranges of values. Not every dataset does require normalization, however as our dataset has variables with different ranges we need to normalize it.

With the help of sklearn's preprocessing functionality, **apply the normalisation-transformation on each feature vector for the training table (but not the Tie Strength)**. You can find more information about the preprocessing functionality here. [5] Again, output the first ten entries of your dataframe.

]:
```
# TODO: Apply normalisation transformation
```

Tasks:(cont.)

Task 4.2: The Regression Model

a) Building the model

1. Finally, the regression can be applied on the dataframe. For this purpose, **split** the dataframe into `y`: the target variable and `X`: the predictive variables. As you have read above, our model contains a bias/intercept named α. This will be realized in the model by adding a constant (1.0), that gets multiplied with its own coefficient and therewith forms the intercept. It represents the target value when all explanatory variables are zero. Once again `add_constant(X)` will be of use.

Split the dataframe, **add** the constant and then **apply** a multiple linear regression on the training table, the statsmodels functions `OLS()` and `fit()` will help you with that. Output the summary with `model.summary()`.

```
# TODO: Add constant & build the regression model
```

2. As you can see the model's summary provides us with a multitude of informations about its performance. Now we need to evaluate our model based on these values. Find out what the meaning of the following statistics are: `R-squared`, `Adj. R-squared`, `Prob (F-statistic)`, the predicitve variables' significances `P>|t|`. This site [6] does a good job explaining them intuitively.

Evaluate our model's performance by giving a short comment on the obtained values for them. Don't write more than 5 sentences!

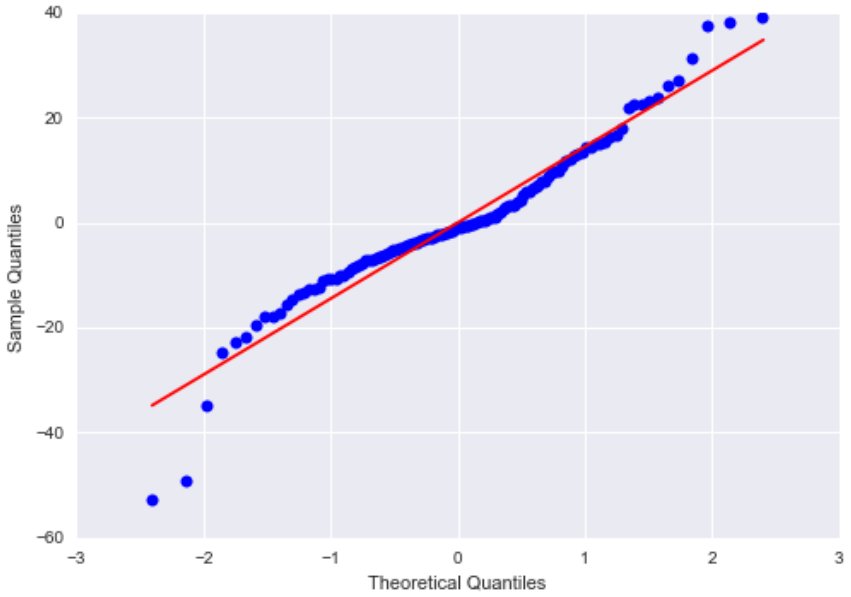
TODO: Write your evaluation here!

3. Now additionally **compare** the obtained coefficients `coef` for our predictive variables to the findings of the paper referenced in [1]. Wich kind of variables (Intimacy, Duration, Structural, Social distance) have the most influence on the Tie Strength according to our regression? You can also comment on specific predicitive variables' values. Keep in mind that the paper's coefficients are already standardized regarding the variabe's values, while ours do not yet compensate for them. Don't write more than 5 sentences.

TODO: Write your observations here!

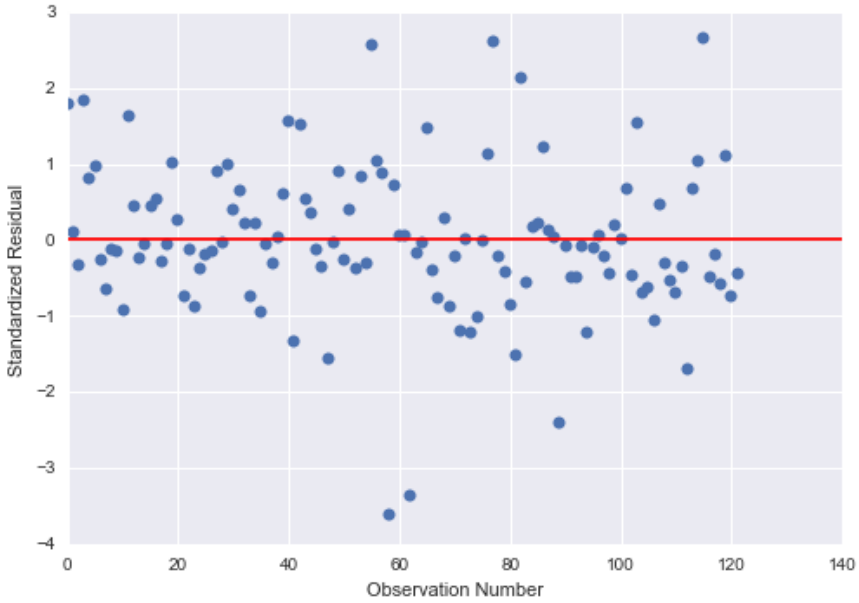
Q-Q Plot & Residuals Plot

- **QQ Plot**:
  compare two distributions
  (here: $\{f_i\}_{i\in[1:N]}$ and $\{y_i\}_{i\in[1:N]}$)
  by plotting quantiles against each other



[4]

- **Residuals Plot**



[4]

Tasks:(cont.)

b) OPTIONAL: Goodness of Fit

After you have now analyzed some of the statistics of our model, there are some additional methods of analyzing the Goodness of Fit of our model. There are several methods to evaluate the Goodness of Fit of a regression. In this exercise, you will work with two of them: the Q-Q Plot and the Residual Plot.

1.: Q-Q Plot

Create a Q-Q Plot and evaluate what the result means for your fit. Plot the model's residuals on one axis and the normal distribution on the other axis, `scipy.stats` will provide it to you. What does the result tell you regarding your fit? Don't write more than 4 sentences.

Hint: Statsmodles offers a function for Q-Q Plots.

```
import scipy.stats as stats

# TODO: Create the QQ-Plot
```

TODO: Write your interpretation here!

2.: Residual Plot

Now evaluate your fit by plotting the residuals with matplotlib. The plot should show the standardized residuals for each entry. What does the result tell you regarding your fit? Don't write more than 4 sentences.

Hint: The standardized residuals can be accessed via `model.resid_pearson`.

```
# TODO: Create the Residual-Plot
```

TODO: Write your interpretation here!

Tasks:(cont.)

Task 4.3: Prediction of Tie Strengths

As a last step, we want to compare the predicted tie strenghts with the true values.

a) **Use the regression model to predict the Tie Strength values, for previously unseen data.** Statsmodels will be of help with that. **Remember** that we normalized the training data, so this needs to be done here as well.

```
# TODO: Perform normalisation transformation, add constant & predict the Tie Strengths



# An example for queries:
# pred_table[pred_table['Tie Strength'] > 0.7].head(5)
```

b) Now we want to compare the predictions with the actual model. **Hint:** Execute the code snippet to use the MSE (Mean Squared Error) from sklearn to compare the two outputs. Please write down in one sentence how this output can be interpreted.

```
# Compare your prediction with the true tie strenghts using i.e. the MSE
mean_squared_error(true_tie_strength, predicted_tie_strength, squared=False)
```

Are the predictions in line with the observations above? Pick a few entries to back up your observations. If you would like to talk about other than the first ten entries, you can query a pandas dataframe similar to SQL. More information on how to do this is available in the pandas documenation [4].

TODO: Write your observation and explanation here!

Submitting Your Solution

- work by expanding the .ipynb iPython notebook for the exercise that you downloaded from Moodle.
- save your expanded .ipynb iPython notebook in your working directory.
  Submit your .ipynb iPython notebook via Moodle (nothing else)
- remember: working in groups is not permitted.
  Each student must submit her own ipynb notebook!
- we check for plagiarism. Each detected case will have the consequence of 5.0 for the whole exercise grade.
- deadline: please check Moodle

Citations

[1] E. Gilbert and K. Karahalios: *Predicting Tie Strength With Social Media*. Proceedings of the SIGCHI Conference on Human Factors in Computing Systems. ACM, 2009.

[2] C. Bishop: *Pattern Recognition and Machine Learning*. 2006, chapter 3

[3] Günnemann et al: Machine Learning 1 lecture slides 2017