

*Machine Learning*

Project Report

*Recommendation system of neighborhood in the Сollegium UCU*

**

Student: *Anastasia Vedernikova*

Lecturer:*Sergey Shelpuk*

Lecturer: *Anastasiya Kornilova*

**KEYWORDS**

Recommendation, neighbors, collegium UCU, regression, classification, clustering

**MOTIVATION**

Success of students in the university depends on a combination of many factors: moral support, good physical well-being, basic knowledge, motivation and diligence to explore new material. The first semester is the key period that has the greatest impact on students’ life. A schoolboy just entered the university greatly changes his lifestyle. He becomes more mature, responsible, learns to make his own decisions. The student gets acquainted with new people and chooses friends with whom he will spend next 4 or possibly more years. It is especially difficult for students who study in another city. They have to get used to the new city and neighbors in the dormitory, to put up with their way of life. The atmosphere in which the student lives directly affects his academic achievements. Friendly relationships with a neighbor in the dormitory and a fun time spent outside the classroom motivates and gives more strength for studying. At a time when quarrels and tense relationships make the student think not about learning but about his problems. Getting new friends in new city is vital for academic achievements and psychological health. There are a lot of researches about influence of community, family and friends on success in studying. One of them – “***Friendship and Student Engagement, Achievement, and Persistence in College***” by **Andrew Joseph Mauk**. The most salient conclusion from the study is that friendship quality and friendship types are all positively related to college student engagement. This finding informs us that friendship is an important contributor to the actions and behaviors of college students, and that **friendship is a crucial component of the college student experience.**

**PURPUSE OF THE PROJECT**

To develop an effective settlement system in the dormitory (collegium UCU) to increase the likelihood of a friendly relationships between students, which will affect their success in studying and satisfying student life. Created recommendation system will give student a chance to answer questions and get 3 students’ Facebook links with the highest similarity in habits and interests/probability to become friends with, so they can agree on living together.

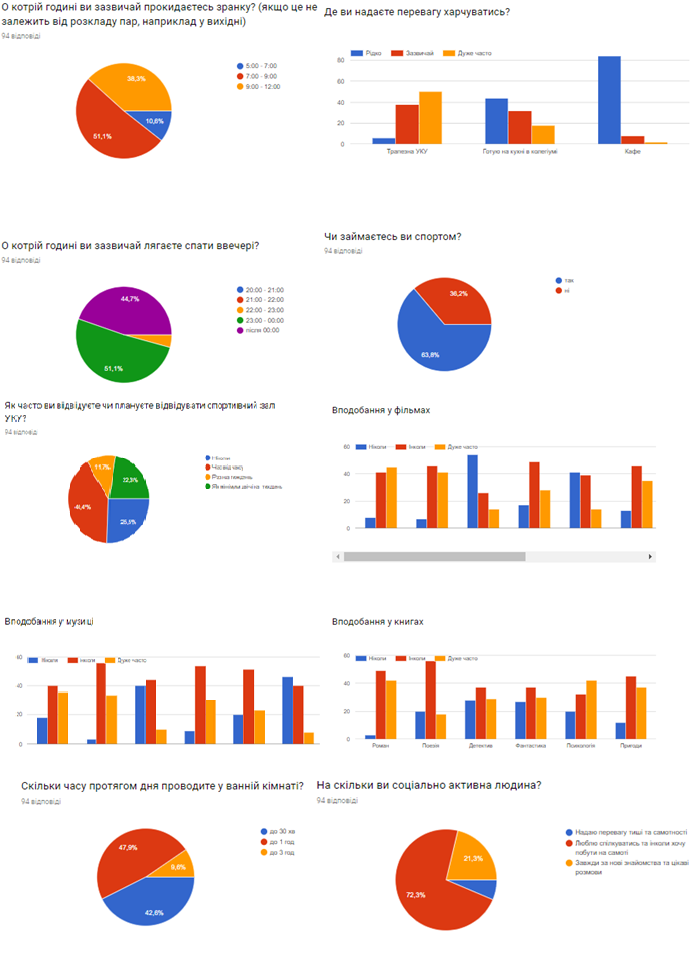
**DATA GATHERING**  
I have been studying several articles on the Internet to select the main issues that affect the ability to have a comfortable life together. Basing on this knowledge I have created questionnaire and agreed it with the master of the collegium.   
Collegium administration sent questionnaires to students living in a colegium. 94 of them have already answered it.

**QUESTIONNAIRE**

|  |  |
| --- | --- |
| Прізвище та ім'я |  |
| Вкажіть будь-ласка посилання на ваш facebook, для того щоб ваші потенційні сусіди могли зв'язатись з вами |  |
| Назвіть імена та прізвища ваших сусідів по кімнаті |  |
| Вкажіть номер кімнати в гуртожитку |  |
| Оцініть ваш рівень задоволення життя разом з ними | Від 1 (не хочу з ним/нею жити) до 5 (найкращі друзі) |
| О котрій годині ви зазвичай прокидаєтесь зранку? (якщо це не залежить від розкладу пар, наприклад у вихідні) | 5:00 - 7:00, 7:00 - 9:00, 9:00 - 12:00 |
| О котрій годині ви зазвичай лягаєте спати ввечері? | 20:00 - 21:00, 21:00 - 22:00, 22:00 - 23:00,  23:00 - 00:00, після 00:00 |
| Де ви надаєте перевагу харчуватись? | Рядки: Трапезна УКУ, Готую на кухні в колегіумі, Кафе  Стовбці: Рідко, Зазвичай, Дуже часто |
| Чи займаєтесь ви спортом? | так, ні |
| Як часто ви відвідуєте чи плануєте відвідувати спортивний зал УКУ? | Ніколи, Час від часу, Раз на тиждень,  Як мінімум двічі на тиждень |
| Вподобання у музиці | Рядки: рок, поп, джаз, реп, класика, контемп, латиноамериканська, електро/металіка  Стовбці: Ніколи, Інколи, Дуже часто |
| Вподобання у фільмах | Рядки: Комедія, Драми, Жахи, Історичні/біографія, Бойовики, Фантастика, Детективи, Мелодрами, Мюзикли, Мультики  Стовбці: Ніколи, Інколи, Дуже часто |
| Вподобання у книгах | Рядки: Поезія, Роман, Пигоди, Детектив,  Фантастика, Психологія, Історичні  Стовбці:Ніколи, Інколи, Дуже часто |
| Як часто ви прибираєте? | Раз на місяць, Раз на тиждень, Двічі на тиждень, Кожного дня |
| Як часто ви приймаєте/ плануєте приймати гостей? | Раз на рік, Раз на місяць, Раз на тиждень,  Декілька разів протягом тижня |
| Релігія | Християнство, Іслам, Юдаїзм, Буддизм, Атеїзм, Інше |
| Чи ви палите? | Так, ні |
| Спосіб навчання | Рядки: В компанії, з музикою, більше 5 год в день, в кімнаті  Стовбці: Ніколи, Інколи, Дуже часто |
| Скільки часу протягом дня проводите у ванній кімнаті? | До 30хв, до 1 год, до 3 год |
| На скільки ви соціально активна людина? | Надаю перевагу тиші та самотності,  Люблю спілкуватись та інколи хочу побути на самоті,  Завжди за нові знайомства та цікаві розмови |
| Чи відноситесь ви до вегетаріанців? | Так, ні |
| Спосіб відпочинку | Рядки: Вечірка в клубі, Пісні під гітару,  Похід в гори, Прогулянка парком, Подорожі до інших міст  Стовбці: Ніколи, Інколи, Дуже часто |

**STATISTICS OF ANSWERS**

****

****

****

**PROCESSING DATA**

I have read questionnaires in dataframe and processed them using pandas library in python. In aim to analyze students’ answers, I have cleaned excessive and dirty data. For example, answers such as: “Це Колегіум, а не гуртожиток!!! 203”, “5 (колегіум на Мушака)” I have converted to just numbers 203 and 5. Also I have generated dummy variables (0/1) for more convenient usage.

Result of processing csv data is saved to “**average\_neigh\_rate**\ProcessedCollegium.csv” in the form:

Середній рівень задоволення життя разом (сума оцінок двох сусідів /2), відповіді на питання першого студента (0 або 1- dummy variables)(55 questions), відповіді на питання його сусіда ( теж dummy variables)(55 questions).

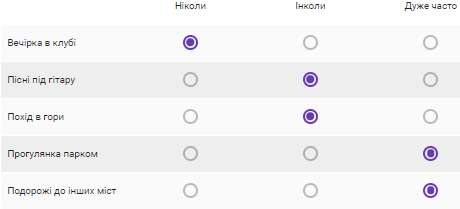
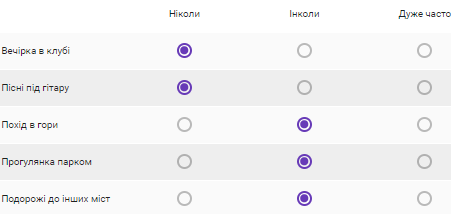
*General schema for first row in ProcessedCollegium.csv*

|  |  |
| --- | --- |
| Середній рівень задоволення життя разом | 4.0 |
| О котрій годині ви зазвичай прокидаєтесь зранку? (якщо це не залежить від розкладу пар, наприклад у вихідні)\_7:00 - 9:00 | 0 |
| О котрій годині ви зазвичай прокидаєтесь зранку? (якщо це не залежить від розкладу пар, наприклад у вихідні)\_9:00 - 12:00 | 1 |
| О котрій годині ви зазвичай лягаєте спати ввечері? \_після 00:00 | 0 |
| Де ви надаєте перевагу харчуватись? [Трапезна УКУ]\_Зазвичай | 1 |
| Де ви надаєте перевагу харчуватись? [Трапезна УКУ]\_Рідко | 0 |
| Де ви надаєте перевагу харчуватись? [Готую на кухні в колегіумі]\_Зазвичай | 0 |
| Де ви надаєте перевагу харчуватись? [Готую на кухні в колегіумі]\_Рідко | 1 |
| ... | 0/1 |
| О котрій годині ви зазвичай прокидаєтесь зранку? (якщо це не залежить від розкладу пар, наприклад у вихідні)\_Сусід\_1\_7:00 - 9:00 | 0 |
| …\_Сусід\_1\_... | 0/1 |

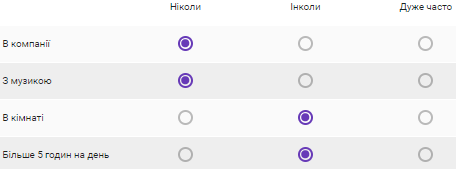
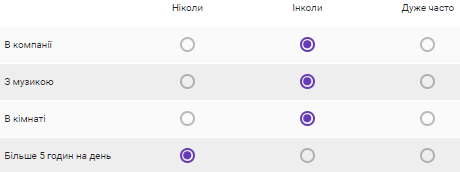
Another similar schema was created with using the level of satisfying of **first neighbor** instead of average level and was saved to the directory “first\_neigh\_rate\ProcessedCollegium.csv”

While modeling I have found that only one student evaluated his neighbor with 1 mark and decided to analyze this **outlier**. Кадиляк Святослав lives with Ігор Базан and Артур Федаш in 230 room. But only Ігор Базан have filled the questionnaire. He evaluated level of satisfying living together in 4 score. After reviewing their answers I can make conclusion that they have quite similar interests. They both chose Christianity as a religion, they have same sleeping time and cleaning habits, students are not vegetarians and don’t smoke. I can conclude that they are quite sociable but have different preferences in books, music, relax time spending.

*Ігор Базан Кадиляк Святослав*

 VS 

*Спосіб відпочинку & Споіб навчання*

** VS **

**MODELS FOR SCHEMA WITH AVERAGE LEVEL**

Linear Regression, Multinomial Logistic Regression, KNN Classifier

**LINEAR REGRESSION** is used to model relationship between a scalar depended variable *y (average level of satisfying for 2 neighbors)* and more explanatory variables denoted as X *(answers for questions of those 2 neighbors).* Trained model predicts continuous value - average level of satisfying for 2 new neighbors.

**MULTINOMIAL** **LOGISTIC REGRESSION** is a [classification](https://en.wikipedia.org/wiki/Statistical_classification) method that generalizes [logistic regression](https://en.wikipedia.org/wiki/Logistic_regression) to [multiclass problems](https://en.wikipedia.org/wiki/Multiclass_classification), with more than two possible discrete outcomes. That is, it is a model that is used to predict the probabilities of the different possible outcomes of a [categorically distributed](https://en.wikipedia.org/wiki/Categorical_distribution) [dependent variable](https://en.wikipedia.org/wiki/Dependent_variable), given a set of [independent variables](https://en.wikipedia.org/wiki/Independent_variable).

**KNN CLASSIFIER** is a non-parametric method used for classification, the output is a class membership. An object is classified by a majority vote of its neighbors, with the object being assigned to the class most common among its *k* nearest neighbors.

For building multinomial logistic regression and KNN classifier, I have created ***3 main classes of marks***(levels): class **1** – (1, 1.5, 2, 3 points), class **2** – (3.5, 4 points), class **3** – (4.5, 5 points). And processed labels according to this class table.

I was using only dummy variables (0/1) for training/testing models as X, and average level of satisfying of living together as Y.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **SCORES FOR SCHEMA WITH AVERAGE LEVEL**   |  |  |  |  | | --- | --- | --- | --- | | Model | Train data in % | Test data in % | Score | | Linear Regression | 70 | 30 | 0.156455665677 | | Logistic Regression | 70 | 30 | 0.47619047619 | | KNN classifier | 70 | 30 | 0.52380952381 | | Linear Regression | 90 | 10 | 0.0253472506046 | | Logistic Regression | 90 | 10 | **0.571428571429** | | KNN classifier | 90 | 10 | 0.428571428571 | |
| ***Multinomial Logistic Regression gives the best score – 0.571428571429 on 10% of test data.***  **MODELS FOR SCHEMA WITH FIRST LEVEL**  Linear Regression, Multinomial Logistic Regression, KNN Classifier,  Random Forest Classifier, XGBoostRegressor, XGBoostClassifier, PCA  **RANDOM FOREST CLASSIFIER**  A random forest is a meta estimator that fits a number of decision tree classifiers on various sub-samples of the dataset and use averaging to improve the predictive accuracy and control over-fitting.  **XGBOOST**  XGBoost is an implementation of gradient boosted decision trees designed for speed and performance that is dominative competitive machine learning. XGBoost has 3 types of parameters: general (booster, silent, nthread), booster (min\_child\_weight, max\_leaf\_nodes, max\_depth, eta, subsample, etc.), learning task (eval\_metric, objective, seed). To tune them, I was using **GridSearchCV** from sklearn. It gave me -  The best parameters set found on development set: {'n\_estimators': 11, 'max\_depth': 1, 'learning\_rate': 0.1101}.  **SCORES FOR SCHEMA WITH FIRST LEVEL**  *(after deleting outlier without PCA)* |

|  |  |  |  |
| --- | --- | --- | --- |
| яModel | Train data in % | Test data in % | Score |
| Linear Regression | 70 | 30 | -1.09359245507 |
| Logistic Regression | 70 | 30 | 0.2 |
| KNN classifier | 70 | 30 | 0.3 |
| Linear Regression | 90 | 10 | -2.14341025019 |
| Logistic Regression | 90 | 10 | 0.428571428571 |
| KNN classifier | 90 | 10 | **0.571428571429** |
| Random Forest classifier | 70 | 30 | 0.45 |
| Random Forest classifier | 90 | 10 | **0.571428571429** |
| XGBoost classifier | 70 | 30 | 0.4 |
| XGBoost classifier | 90 | 10 | **0.5714285714285714** |
| XGBoost regressor | 70 | 30 | -0.220602744373161 |
| XGBoost regressor | 90 | 10 | -0.552887291825936 |

**PRINCIPAL COMPONENT ANALYSIS**

PCA is a statistical procedure that uses an orthogonal transformation to convert a set of observations of possibly correlated variables into a set of values of linearly uncorrelated variables called principal components.

**SCORES FOR SCHEMA WITH FIRST LEVEL AFTER PCA** *(after deleting outlier)*

|  |  |  |  |
| --- | --- | --- | --- |
| Model | Train data in % | Test data in % | Score |
| Linear Regression | 70 | 30 | -0.0372916003557 |
| Logistic Regression | 70 | 30 | 0.4 |
| KNN classifier | 70 | 30 | 0.25 |
| Linear Regression | 90 | 10 | -0.271570389748 |
| Logistic Regression | 90 | 10 | **0.714285714286** |
| KNN classifier | 90 | 10 | 0.428571428571 |
| Random Forest classifier | 70 | 30 | 0.45 |
| Random Forest classifier | 90 | 10 | 0.5 |
| XGBoost classifier | 70 | 30 | 0.45 |
| XGBoost classifier | 90 | 10 | 0.42857142857142855 |
| XGboost regressor | 70 | 30 | 0.25 |
| XGBoost regressor | 90 | 10 | 0.14285714285714285 |

*PCA has changed results, especially for logistic regression (from 0.42 to 0.71)*

**SCORES FOR FIRST SCHEMA BEFORE DELETING OUTLIER** *(without PCA)*

|  |  |  |  |
| --- | --- | --- | --- |
| Model | Train data in % | Test data in % | Score |
| Linear Regression | 70 | 30 | -0.214695696182 |
| Logistic Regression | 70 | 30 | 0.190476190476 |
| KNN classifier | 70 | 30 | 0.285714285714 |
| Linear Regression | 90 | 10 | -0.563838862615 |
| Logistic Regression | 90 | 10 | 0.285714285714 |
| KNN classifier | 90 | 10 | **0.857142857143** |

*Before deleting outlier, I got the best score of 0.857 using KNN classifier*

**FEATURES ENGENEERING USING K-MEANS CLUSTERING**

One more approach that I used to find a correlation between my data – K-Means clustering of all single neighbors into 15, 7 and 3 clusters. Then I added the number of cluster to which particular neighbor belong as new feature. So, as a result I have 3 more columns, after converting to dummy variables - 25 columns. With parameter “drop\_first=*True”,* I have got 22 features*.* New pairs with K-Means features were saved to “Kmeans\PairsWithKmeansFeatures.csv” and were used for building regressions and classifications, results of which you can see below.

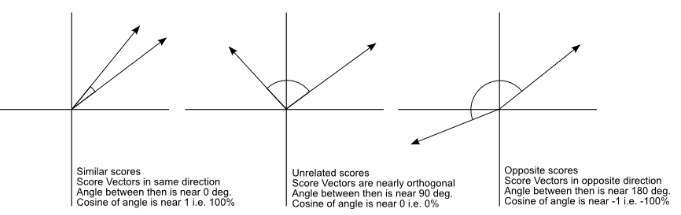
**SCORES FOR FIRST SCHEMA USING FEATURES GENERATED BY K-MEANS CLUSTERING**

|  |  |  |  |
| --- | --- | --- | --- |
| Model | Train data in % | Test data in % | Score |
| Linear Regression | 70 | 30 | -0.2676104217134676 |
| Logistic Regression | 70 | 30 | 0.2857142857142857 |
| KNN classifier | 70 | 30 | 0.3333333333333333 |
| Linear Regression | 90 | 10 | -0.5302380833381815 |
| Logistic Regression | 90 | 10 | 0.2857142857142857 |
| KNN classifier | 90 | 10 | 0.5714285714285714 |
| Random Forest classifier | 70 | 30 | 0.6666666666666666 |
| Random Forest classifier | 90 | 10 | **0.8571428571428571** |

There are different methodologies to calculate difference between vectors representing students’ answers (as two neighbors). I have calculated **cosine similarity –** measure of orientation (not magnitude) that gives cosine of angle between 2 vectors of any dimension.

Cosine similarity formula - 

Interpretation of results: similarity depends on angle between vectors.



Results in degrees:

Pairs with **2** score - 69.7, 74.7, 56.1, 65.4, 65.7 – *mean*: 66.32

Pairs with **3** score - 74.7, 53.7, 61.4, 24.2, 60.4, 65.4, 65.5, 53.4, 53.7, 65.5, 67.3, 58.6, 60.7 – *mean*: 58.8

Pairs with **4** score - 58.0, 63.3, 67.3, 56.0, 58.0, 54.0, 60.4, 64.6, 24.2, 59.8, 61.4, 66.9, 62.6, 56.1, 71.6, 62.8, 66.7, 64.8, 67.3, 69.7, 63.3, 62.8, 61.1, 70.2, 65.4, 63.7, 58.9, 58.9, 70.6, 70.6 – *mean*: 62.0

Pairs with **5** score - 54.0, 66.9, 73.0, 73.0, 71.6, 64.8, 65.7, 55.9, 55.9, 51.2, 56.8, 53.4, 63.7, 55.2, 55.2, 51.2, 58.6, 56.0 – *mean*: 60.1

As we can see, there are several examples of expected results (3score – 74.7, 4score – 24.2). But there are also pairs that are almost uncorrelated with 73.0 degrees of angle but have 5 score. Means are very close, that means that cosine similarity will not be a best way to create pairs.

**CONCLUSION**

I was experimenting with different models: KNN classifier, Linear and Logistic regressions, XGBoost, Random Forest classifier changing processing of data. Approaches: different schemas (average vs first score), k-means clustering of all students to 3, 7, 15 clusters and adding numbers of clusters as new features, calculating cosine similarity between neighbors (comparing it with real scores, tried to use as additional feature), PCA with different numbers of components, analysis of outliers, tuning parameters for XGBoost. I have tried different combination of steps:

1. Adding k-means features + Model (For each model)
2. Adding cosine similarity as feature + Model (Foe each model)
3. Adding k-means features + Adding cosine similarity as feature + Model (For each model)
4. Adding k-means features + PCA + Model (For each model)
5. Adding cosine similarity as feature + PCA + Model (Foe each model)
6. Adding k-means features + Adding cosine similarity as feature + PCA + Model (For each model)

**As a result the best combination for my dataset was: K-Means features + Random Forest classifier with score of 0.857 on 10% of test data and 0.(6) on 30% of test data.**

The main **challenge** in this project was - very small dataset of only 94 students’ answers. But I was trying to get maximum of it. I have created one more questionnaire for new students (not gathering history information as above), program for processing answers and recommendation system (using combination above) that gives 3 best neighbors’ Facebook links to agree on living together in collegium UCU. While modeling recommendation system I have faced with problem of converting data to **dummies**, because get\_dummies func from pandas creates variables according to the variety of values in column (it does not know what options where proposed in the form), so train data (history) and new data were of different size. To make them of same size, I have wrote own program for creating dummies. <https://docs.google.com/forms/d/e/1FAIpQLSdbViF9EvK3vp3ToGSHuwxg6W2tSA5n1Wb5cQGl76xv1BV9bQ/viewform?usp=sf_link> - google form for new students, available for accounts in domain Ucu.edu.ua

<https://github.com/AnastasiaVedernikova/NeighborsInCollegium>-experiments & recommendation system

**REFERENCES**

Models - <http://cs229.stanford.edu/syllabus.html>

K-Means clustering for feature engineering: <https://habr.com/post/270367/>

Cosine similarity: <http://blog.christianperone.com/2013/09/machine-learning-cosine-similarity-for-vector-space-models-part-iii/>