Final Report

Hanna Pylieva, Kateryna Zorina, Yurii Pryima, Yurii Kaminskyi

1. Problem statement

Originally, our problem statement was the following:
"*Find pages in Ukrainian wikipedia that should be translated*".
But this task is very subjective and hard to evaluate. How do we define "better" pages?

So as a result of our discussion we decided to change problem statement to:
"*Prediction of page translation from its historical data*".

So to solve our original task we find pages in Ukrainian wikipedia that most probably will be translated. Here we are making an assumption that page should be translated because similar type of pages were translated before. Also we are making an assumption that page was translated from Ukrainian to English if Ukrainian page was created before the corresponding English page.

2. Data collection

At the beginning we should find all ukrainian pages that have english translation and were created before english page. This is hard task because english wikipedia has more than 40 000 000 pages.

We investigated 3 ways for data collection from wikipedia:

1. Wikipedia rest api
2. PAWS environment
3. Wikimedia dumps

Firstly, we have tried api approach, but it is not appropriate for this problem, because we should run several request for each page to get it`s time creation and translation status. And it takes too much time.

After that we have tried paws notebook. We have faced problem with RAM memory limit on the server. So we decided to run queries for each separate table, download results and do in memory joins to obtain required pages. Code for tables downloding could be found here and tables merging here.

As we received required pages we could collect historical data for them.

2.1 Incoming and outcoming links

One of possible measurements of popularity of the article is the number of incoming and outcoming links. So, we decided to include them into our model.
Luckily, there's a table that helps with that - pagelinks. It contains the information about the pages and the links, that this article could be accessed.

```
+-------------------+--------------------+------+-----+---------+-------+
| Field             | Type               | Null | Key | Default | Extra |
+-------------------+--------------------+------+-----+---------+-------+
| pl_from           | int(10) unsigned   | NO   | PRI | 0       |       |
| pl_from_namespace | int(10)            | NO   | MUL | 0       |       |
| pl_namespace      | int(11)            | NO   | PRI | 0       |       |
| pl_title          | varchar(255) binary | NO  | PRI | NULL    |       |
+-------------------+--------------------+------+-----+---------+-------+
```

So, based on that, we build a queries to access this information.
This queries was runned in PAWS environment.
The whole notebook with code for extracting incoming/outcoming links could be found here.
And here you could find the code that merges the number of incoming/outcoming links to all the other data, that was used for training of our model.

2.2 Users interactions features

We decided that user activity play important role in page translation.
So we decided to include this features:

- **Views :** for each page time series with amount of views per day (*getViews* function)
- **Revisions:** for each page time series with cumulative sum of revisions (*GetRevisions_andAge* function)
- **Amount of contributors:** for each page time series with cumulative sum of contributors (*GetRevisions_andAge* function)
- **Age of page:** for each page time series with number of days since page creation (*GetRevisions_andAge* function)

Data was collected using wikipedia REST api and runed on local machine.
Code for feature collection and time series creation can be found here.

3. Data preprocessing

As a result of data collection step we received time series data for 6 features ('revisions_count', 'contributors_count', 'age_of_page_days', 'num_of_views', 'incoming_links', 'outcoming_links')
for 5.5k translated and 5.5k untranslated pages.

Then we aggregated time series data, so that our final datasets for learning had the next structure:
{ *page_name*, *[features]*, *target_variable* },
where *target_variable* = 1 if an article was translated in the end of 30-days period, else - if it wasn't translated.

We took just maximum of *age_of_page_days* feature for each article. For the rest 5 features we used the next aggregation functions:

- Mean
- Standard deviation
- Minimum value
- Maximum value
- Range (max - min)

- Trend - this is p coefficient from ARIMA(p, d, q) fitting
- Average value of the feature for the last 7 days
- Average value of the feature for the all period

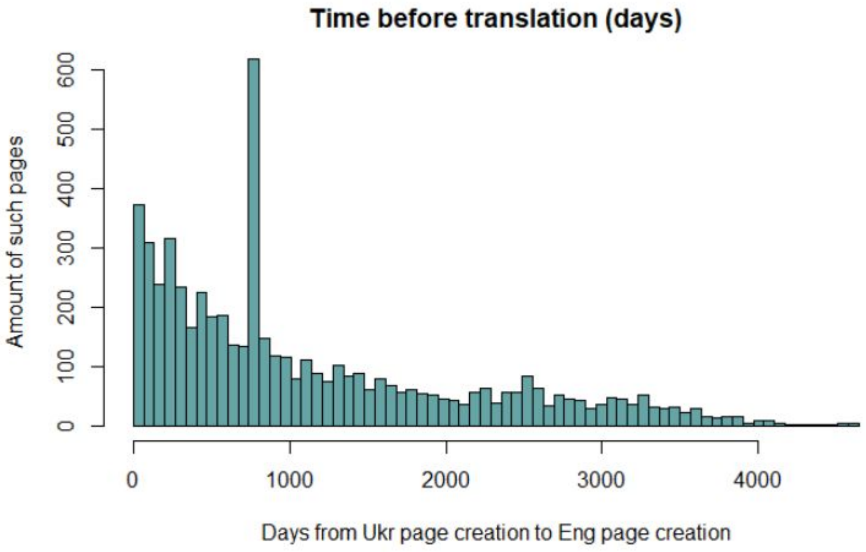
After time series aggregation we noticed that *incoming_links* are stable and don't have neither trend, not standard deviation. That is why we decided to include only *mean* for this feature.

Code for data preprocessing can be found here.

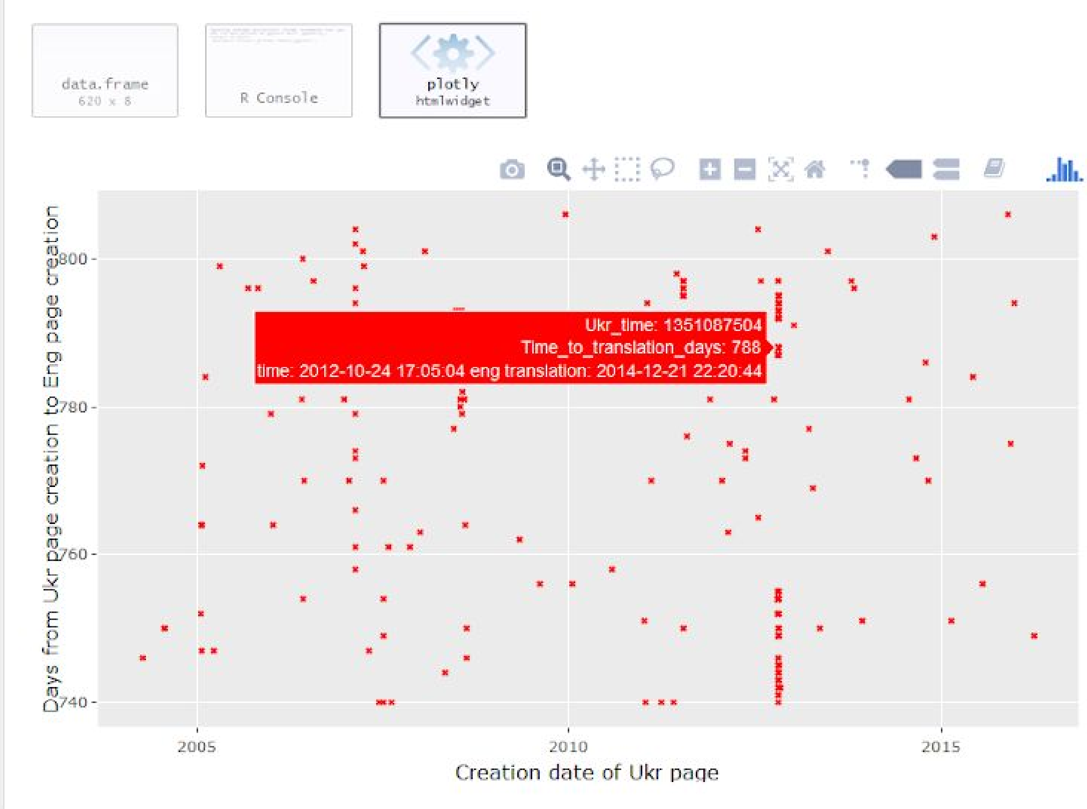
4.Data Analysis

4.1 Time before translation

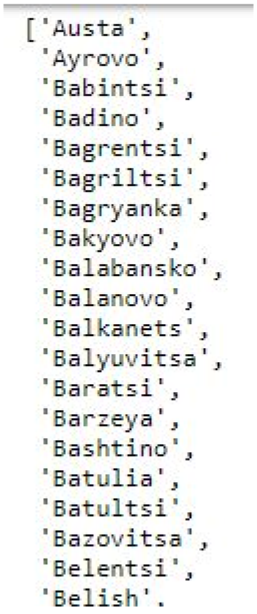
Amount of articles which present both in Ukr and Eng wiki:
132 803
Amount of articles which were created in Ukr before created in Eng wiki:
5 761
Let's investigate, how much time it usually takes from article to be translated from Ukr to Eng:



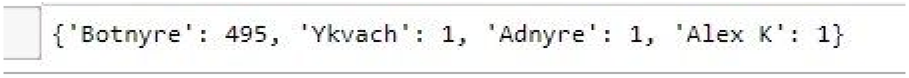
We can see strange peak between 739 and 806 days. If we investigate further:



Most pages from this peak were created in 10-11 month 2012 and translated in 11-12 month of 2014 (498 pages). If we look closer on this pages:



```
['Austa',
 'Ayrovo',
 'Babintsi',
 'Badino',
 'Bagrentsi',
 'Bagriltsi',
 'Bagryanka',
 'Bakyovo',
 'Balabansko',
 'Balanovo',
 'Balkanets',
 'Balyuvitsa',
 'Baratsi',
 'Barzeya',
 'Bashtino',
 'Batulia',
 'Batultsi',
 'Bazovitsa',
 'Belentsi',
 'Belish',
```



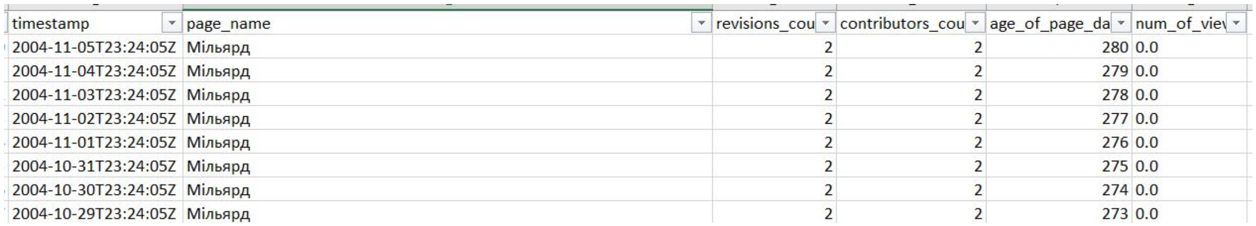
```
{'Botnyre': 495, 'Ykvach': 1, 'Adnyre': 1, 'Alex K': 1}
```

<- creators

Most of them are about some cities and villages of Bulgaria and were written in Ukrainian mostly by **Botnyre** user, which is bot of user Adnyre. And we can see that this bot was very active at this time. Maybe we can exclude this data from further analysis.

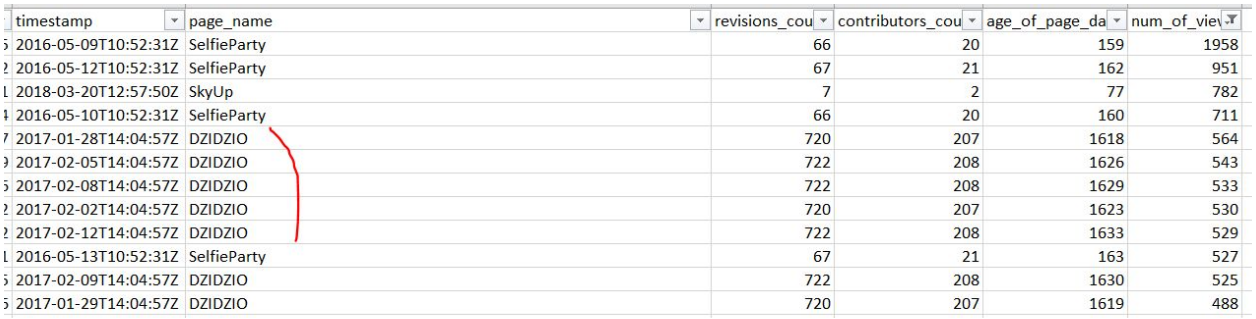
4.2 Timeseries

Mostly data for 30 days is very static, 0 views or 1-2 views, small change in revisions etc.



timestamp	page_name	revisions_cou	contributors_cou	age_of_page_da	num_of_vie
2004-11-05T23:24:05Z	Мільярд	2	2	280	0.0
2004-11-04T23:24:05Z	Мільярд	2	2	279	0.0
2004-11-03T23:24:05Z	Мільярд	2	2	278	0.0
2004-11-02T23:24:05Z	Мільярд	2	2	277	0.0
2004-11-01T23:24:05Z	Мільярд	2	2	276	0.0
2004-10-31T23:24:05Z	Мільярд	2	2	275	0.0
2004-10-30T23:24:05Z	Мільярд	2	2	274	0.0
2004-10-29T23:24:05Z	Мільярд	2	2	273	0.0

We were afraid, that we pulled data wrong, but after further investigation we find out that there are pages with high activity before translation, like this one:



timestamp	page_name	revisions_cou	contributors_cou	age_of_page_da	num_of_vie
2016-05-09T10:52:31Z	SelfieParty	66	20	159	1958
2016-05-12T10:52:31Z	SelfieParty	67	21	162	951
2018-03-20T12:57:50Z	SkyUp	7	2	77	782
2016-05-10T10:52:31Z	SelfieParty	66	20	160	711
2017-01-28T14:04:57Z	DZIDZIO	720	207	1618	564
2017-02-05T14:04:57Z	DZIDZIO	722	208	1626	543
2017-02-08T14:04:57Z	DZIDZIO	722	208	1629	533
2017-02-02T14:04:57Z	DZIDZIO	720	207	1623	530
2017-02-12T14:04:57Z	DZIDZIO	722	208	1633	529
2016-05-13T10:52:31Z	SelfieParty	67	21	163	527
2017-02-09T14:04:57Z	DZIDZIO	722	208	1630	525
2017-01-29T14:04:57Z	DZIDZIO	720	207	1619	488

DZIDZIO is Ukrainian singer, who is popular among young Ukrainians.

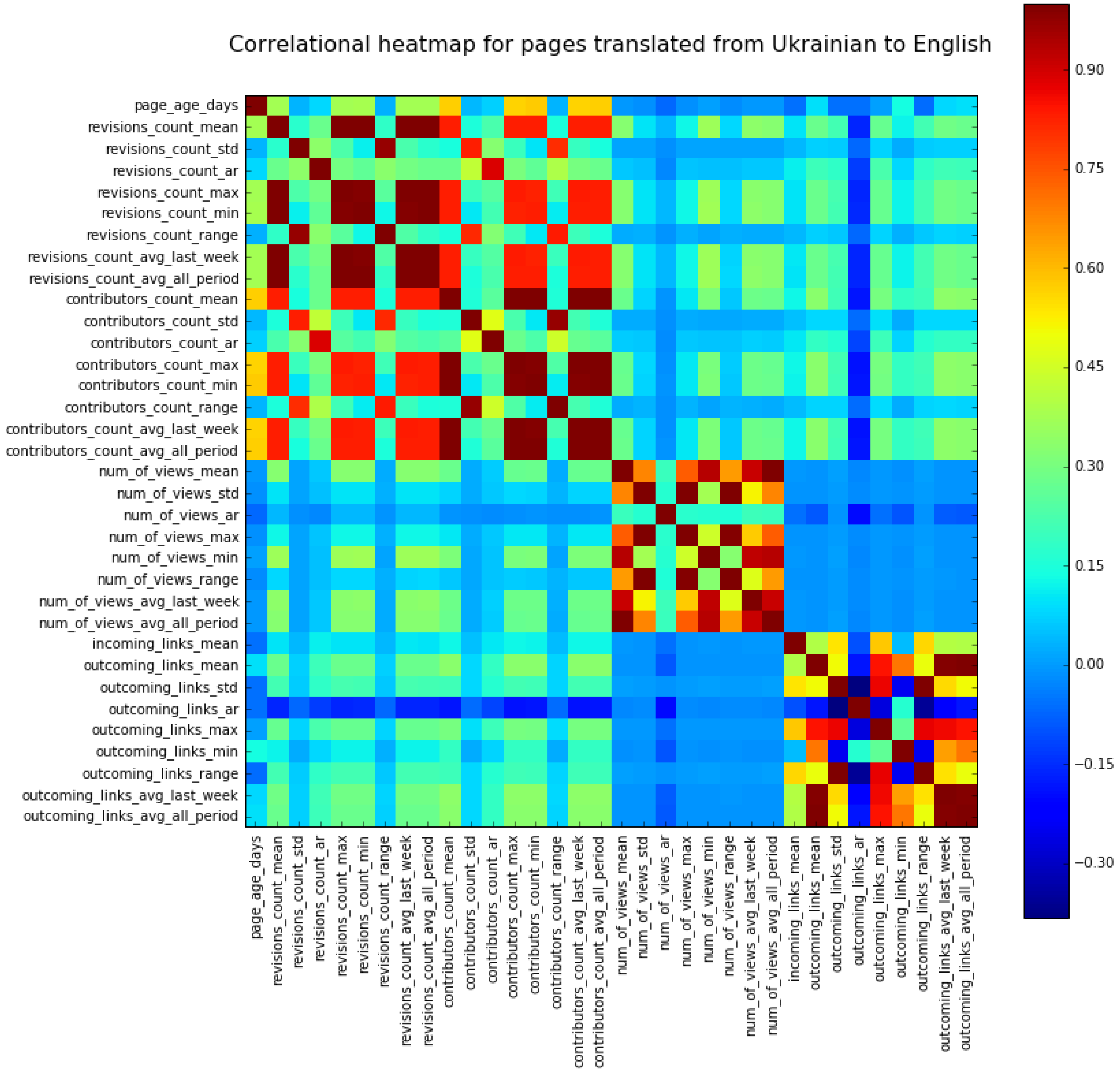
Also some more analysis of pages with high activity before translation:

- SelfieParty - Ukrainian movie, released in on March 31, 2016. (translated in June 2016)
- SkyUp - first Ukrainian low-cost carrier
- O.Torvald - Ukrainian band, participated in Eurovision
- Jinjer - Ukrainian band
- Oxxxymiron - Russian rapper
- MBAND - Russian band
- Skinhate - Ukrainian band
- Petcube - Ukrainian startup
- Badcomedian - Russian blogger, was in tour (including Ukrainian cities) on summer-autumn of 2017 (translated in November 2017)

So we can see, that pages that were popular and have high activity before translation are mostly about people who are in some way connected to Ukraine and also some new movies/startups.

5. Modeling

Firstly, we build a correlational matrix for all the features we had:



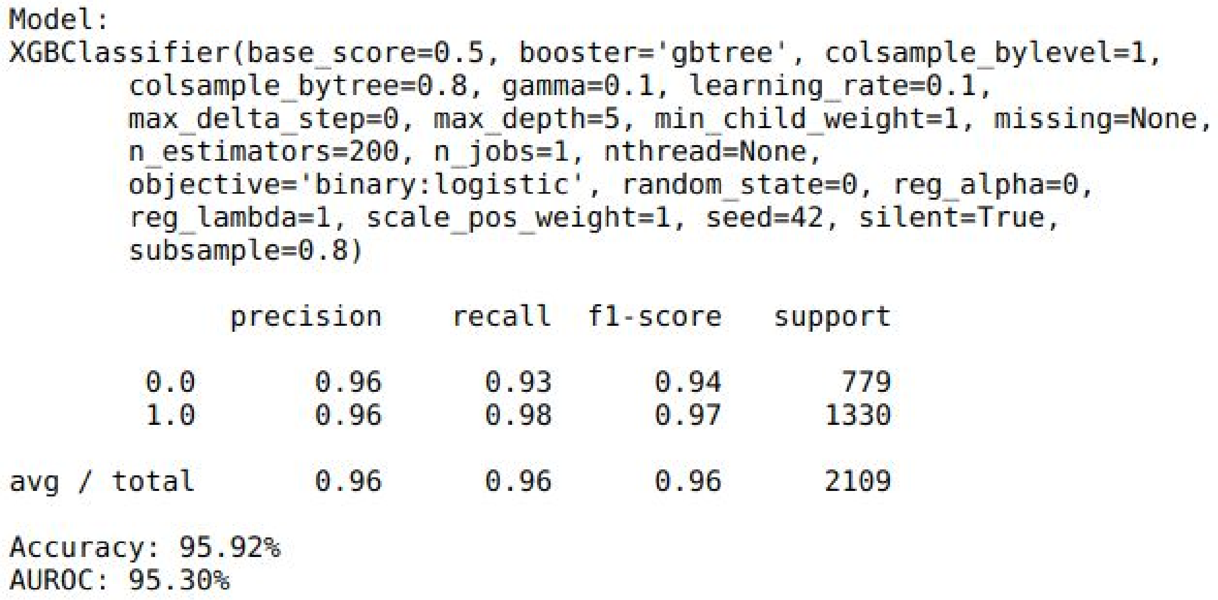
Mostly this correlational map was used to exclude some highly correlated features (like with the case of *incoming_links* time series). A deeper analysis of correlational map can give valuable findings, but due to the lack of time we didn’t spend a lot of time on this.

The data (aggregated features describing translated and untranslated articles with mark of translation) received from preprocessing step was shuffled and splitted into train (80%) and test.

Initially we tried to use One CLass Classification (One Class SVM) only on translated articles to build a decision boundary of translated articles from train set and identify then whether each article from the test set will be within this article. But this model gave bad accuracy, so we decided to use binary classification.

For binary classification we tried SVC and XGBoost, the second gave better results, so we opted for it as final model.

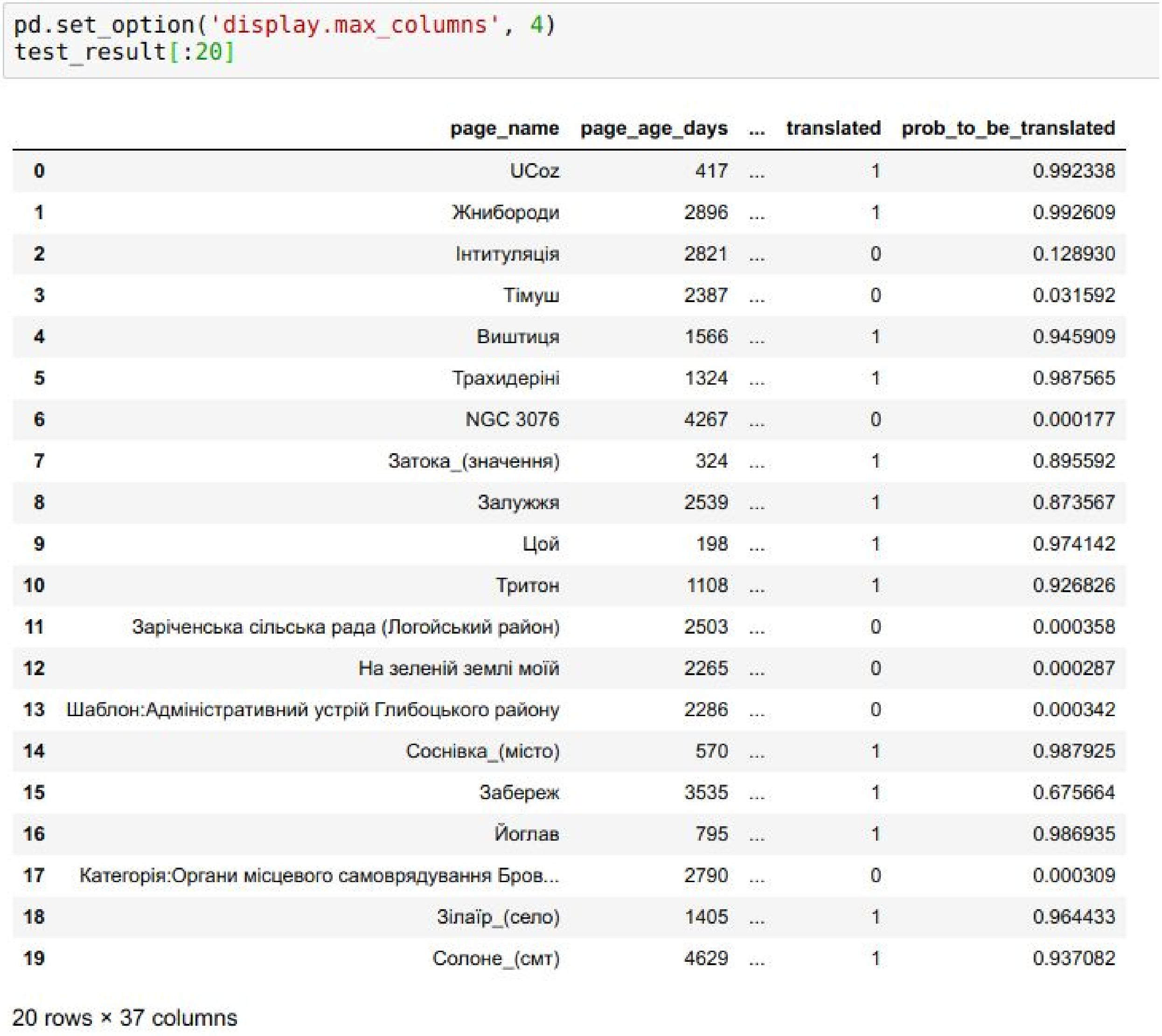
After tuning XGBoost hyperparameters with grid search we received the next model:



```
Model:
XGBClassifier(base_score=0.5, booster='gbtree', colsample_bylevel=1,
       colsample_bytree=0.8, gamma=0.1, learning_rate=0.1,
       max_delta_step=0, max_depth=5, min_child_weight=1, missing=None,
       n_estimators=200, n_jobs=1, nthread=None,
       objective='binary:logistic', random_state=0, reg_alpha=0,
       reg_lambda=1, scale_pos_weight=1, seed=42, silent=True,
       subsample=0.8)

             precision    recall  f1-score   support

        0.0       0.96      0.93      0.94       779
        1.0       0.96      0.98      0.97      1330

avg / total       0.96      0.96      0.96      2109

Accuracy: 95.92%
AUROC: 95.30%
```

According to accuracy and recall it seems like our model performs kind of nice on our test dataset.

The final result looks like the following - to each page from the test set we’ve found the probability to be translated (in the last column on the screen):



```
pd.set_option('display.max_columns', 4)
test_result[:20]
```

	page_name	page_age_days	...	translated	prob_to_be_translated
0	UCoz	417	...	1	0.992338
1	Жнибороди	2896	...	1	0.992609
2	Інтитуляція	2821	...	0	0.128930
3	Тімуш	2387	...	0	0.031592
4	Виштиця	1566	...	1	0.945909
5	Трахидеріні	1324	...	1	0.987565
6	NGC 3076	4267	...	0	0.000177
7	Затока_(значення)	324	...	1	0.895592
8	Залужжя	2539	...	1	0.873567
9	Цой	198	...	1	0.974142
10	Тритон	1108	...	1	0.926826
11	Заріченська сільська рада (Логойський район)	2503	...	0	0.000358
12	На зеленій землі моїй	2265	...	0	0.000287
13	Шаблон:Адміністративний устрій Глибоцького району	2286	...	0	0.000342
14	Соснівка_(місто)	570	...	1	0.987925
15	Забереж	3535	...	1	0.675664
16	Йоглав	795	...	1	0.986935
17	Категорія:Органи місцевого самоврядування Бров...	2790	...	0	0.000309
18	Зілаїр_(село)	1405	...	1	0.964433
19	Солоне_(смт)	4629	...	1	0.937082

20 rows × 37 columns

The Jupyter Notebook where the modeling was performed is here.

6. Results

We obtained pretty good accuracy for prediction of page translation problem.

Whereas we are not sure that our model will generalize well to all Ukrainian wikipedia pages, as there is a rather small amount of pages which were originally created in Ukrainian and later in English Wiki.

Moreover, we understand that our data set contains a lot of unreliable data, like some villages(not Ukrainian), scientific names for insects etc. “Translation” of those pages looks more like a coincidence that page in English appeared later than in Ukrainian.

7. Further work

- Exclude from model unreliable data (for example, pages translated by bots)
- Use only pages in ukrainian wikipedia that was created not only before english translation, but before all other wikipedias
- We could train our model on other wiki languages with more original articles and after transfer that knowledge for our wikipedia
- Use some flag for pages that are usually popular before translation (people, ukrainian popular bands, movies)
- Add more features

Appendix - Responsibilities in team

Team Member	**Part of work**
Yurii Pryima	Gathering the page_ids, timestamp of article creation on Ukrainian Wiki, timestamp of article creation on English Wiki of 1. ~5500 translated articles - ones created on Ukrainian Wiki earlier than on English Wiki 2. 5500 random not translated articles - ones which were created on English Wiki earlier than in Ukrainian Wiki
Kateryna Zorina	Based on data provided by Yurii Pryima collected daily data for 30 days period before date of translation for translated articles and 30 days period before the day of data extraction. Kateryna collected the next data: - Page age in days - Number of revisions - Number of distinct contributors in page revisions - Number of page views Kateryna also performed analysis of time before translation and data in general.
Yurii Kaminskyi	Yurii collected the next data: - Number of incoming links to the page - Number of outcoming links from the page Yurii also merged gathered data into main dataframe, that was used for model training.
Hanna Pylieva	Performed: - data preprocessing - aggregation of time series so that each article is characterized by single row of data. - sketchy correlational analysis of final features (aggregated ones) for articles which were translated - modeling by creation and tuning (with grid search) ML models (XGBoost and SVM)