Модель предсказания балла сочинения в формате ЕГЭ по отдельным критериям

(на материале оцененных сочинений сайтов Kritika24 и Mogu-pisat)

12 критериев
Макс. кол-во баллов – 24 (до 2023 – 25)

- **К1 – проблема**
- **К2 – комментарий к проблеме**
- **К3 – позиция автора**
- ***К4 – отношение к позиции автора***
- **К5 – цельность, связность, последовательность**
- **К6 – точность и выразительность**
- **К7 – орфография**
- **К8 – пунктуация**
- **К9 – грамматика**
- **К10 – речь**
- **К11 – этика**
- **К12 – фактическая точность**

Этапы работы

1. Парсинг и составление датасетов.
2. Составление моделей.
3. Работа над усовершенствованием моделей.

main.py
Главный файл для считывания и запуска команд

```python
if __name__ == '__main__':
    print(f'This is final project v.{VERSION}')
    args = [INPUT_FILE, INPUT_FILE2, BEST_PARAMETERS_FILE, ]
    while EXECUTE_FLAG:
        print_commands()
        required_command = read_command()
        try:
            required_command_num = int(required_command)
        except ValueError:
            required_command_num = -1
        for command_num, command in enumerate(COMMANDS_LIST):
            if command_num == required_command_num or command[0] == required_command:
                command[1](*args)
```

```python
COMMANDS_LIST = [('parse kritika24', parse),
                 ('parse mogu-pisat', parse2),
                 ('analyze criterion', criterion_analyze),
                 ('analyze all criteria', all_criteria_analyze),
                 ('debug', test),
                 ('exit', exit_func),
                 ]
```

main.py

```
This is final project v.1.0
0 - parse kritika24
1 - parse mogu-pisat
2 - analyze criterion
3 - analyze all criteria
4 - debug
5 - exit
Enter required command: 3
```

Парсеры: parser, parser2

Датасет ≈ 2000 – 4000 (в зависимости от критерия)

```python
import numpy as np
import requests as rq
from bs4 import BeautifulSoup
import re
import time
import pandas as pd
from parser import ending
```

```python
print(f"Read {len(ege_list_with_essays)} essays")
print(f"Encode error in {encode_error_counter} essays")
column_names = ['Оригинальный текст', 'Текст сочинения']
for num in range(1, NUMBER_OF_CRITERIA + 1):
    column_names.append(f'Критерий К{num}')
df = pd.DataFrame(ege_list_with_essays, columns=column_names)
# df.to_excel('table with K6,K9.xlsx')
df.to_csv(args[1])
```

Mogu-pisat

Анализ критериев (criterion_analyzer.py)

```
import pandas as pd
import numpy as np
from sklearn.linear_model import LinearRegression, Ridge, Lasso
from sklearn.model_selection import train_test_split
from sklearn.metrics import mean_absolute_error, mean_squared_error, accuracy_score
from sklearn.neighbors import KNeighborsRegressor
from sklearn.tree import DecisionTreeClassifier, DecisionTreeRegressor
from sklearn.linear_model import LogisticRegression
from sklearn.svm import SVC
from gensim.models.doc2vec import Doc2Vec, TaggedDocument
from rus_vectors import rus_vectorize
```

Анализ критериев (criterion_analyzer.py)

```
ALGORITMS = [
    (Lasso, 'Лассо', [0.01, 0.1, 1.0, 5.0, 10.0, 50.0, 100.0, 500., 1000.]),
    (KNeighborsRegressor, 'регрессия ближайшими соседями', [3, 4, 5, 6, 7, 8, 9, 10]),
    (DecisionTreeClassifier, 'классификатор деревом решений', []),
    (DecisionTreeRegressor, 'регрессия деревом решений', []),
    (LogisticRegression, 'логистическая регрессия', []),
    (SVC, 'опорных векторов ', []),
        ]
```

Анализ критериев (criterion_analyzer.py)

```
def trivial_model(df, criterion_num):                #средняя температура по больнице
    y_test = df[f'Критерий К{criterion_num}']
    y_preds = list(map(round, [sum(y_test) / len(y_test)] * len(y_test)))
    #y_preds = regression.predict(X_test)
    accuracy_percents = accuracy_score(y_test, y_preds)
    print(f'Ошибка тривиальной модели для критерия К{criterion_num} с баллом {y_preds[0]}: {mean_absolute_error(y_te
```

Анализ критериев (criterion_analyzer.py)

```
def criterion_analyze(*args, criterion_num = None):
    set_dataframe_print_options(DESIRED_WIDTH, MAX_COLUMNS)
    if criterion_num is None:
        criterion_num = int(input('Введите номер критерия: '))
    original_text_flag = True if (criterion_num in [1, 2, 3, 4]) and ORIGINAL_TEXT_ANALYZE else False
    df = construct_dataframe(*args, criterion_num=criterion_num, original_text_flag=original_text_flag)
    trivial_model(df, criterion_num)
    print('-' * 75)
    print('Word2Vec')
    print('-' * 75)
    vectorized_df = vectorizer(df, 0, original_text_flag, 5, 3, 1)
    train_model(vectorized_df, df[f'Критерий К{criterion_num}'], criterion_num)
    print('-' * 75)
    print('RusVectors')
    print('-' * 75)
    vectorized_df = vectorizer(df, 1, original_text_flag)
    train_model(vectorized_df, df[f'Критерий К{criterion_num}'], criterion_num)
    print('~' * 75)
```

Анализ критериев (criterion_analyzer.py)

```python
def all_criteria_analyze(*args):
    for criterion_num_ in range(1, NUMBER_OF_CRITERIA + 2):
        criterion_analyze(*args, criterion_num = criterion_num_)

2 usages
def punctuation_shift(text):
    for punctuation_sign in [',', '.', '!', '?', ':', '–', ';', '"', '«', '»']:
        #print(f"|{punctuation_sign}|", end=' ')
        text = text.replace(punctuation_sign, f" {punctuation_sign} ")
    return text

def punctuation_delete(text):
    for punctuation_sign in [',', '.', '!', '?', ':', '–', ';', '"', '«', '»']:
        #print(f"|{punctuation_sign}|", end=' ')
        text = text.replace(punctuation_sign, f" ")
    return text
```

Rus_vectors

- Берем предобработанную модель, приклеиваем тег, получился список слов с тегами.
- Составляем вектор, который дает модель. По этому слову с тегом ставит вектор размерности 300. Добавляем к вектору текста. Получаем среднее значение (средний вектор по тексту).
- В итоговом списке лежат 300-мерные вектора текстов.

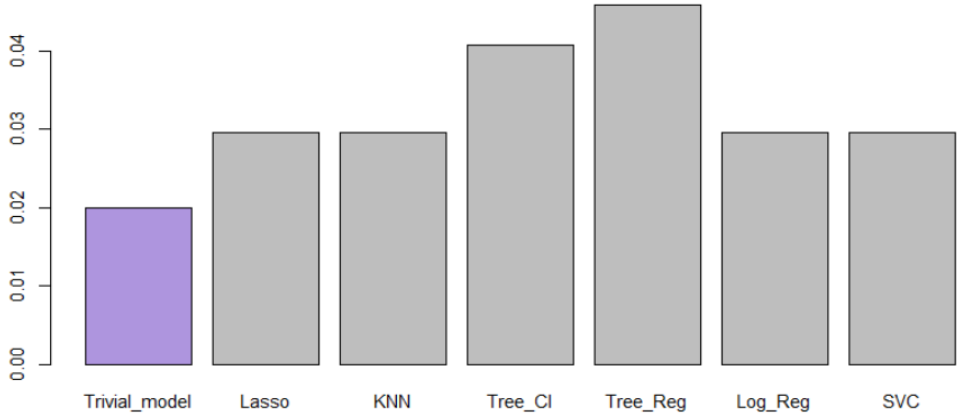
```
def rus_vectorize(list_of_texts):
    model_0 = gensim.models.KeyedVectors.load_word2vec_format('180/model.bin', binary=True)
    udpipe_model_url = 'https://rusvectores.org/static/models/udpipe_syntagrus.model'
    udpipe_filename = udpipe_model_url.split('/')[-1]
    model = Model.load(MODEL_FILE)
    process_pipeline = Pipeline(model, 'tokenize', Pipeline.DEFAULT, Pipeline.DEFAULT, 'conllu')
    def tag_ud(word):
        tagged = []
        output = process(process_pipeline, text=word)
        tagged_line = ' '.join(output)
        tagged.append(tagged_line)
        return '\n'.join(tagged)
    result_list = []
    for text in list_of_texts:
        vec = [0] * 300
        counter = 0
        for word in text:
            try:
                add_vec = model_0[tag_ud(word)]
                vec = list(map(sum, zip(vec, add_vec)))
                counter += 1
            except KeyError:
                pass
        vec = list(map(lambda x: x/counter, vec))
        result_list.append(vec)
    return result_list
```

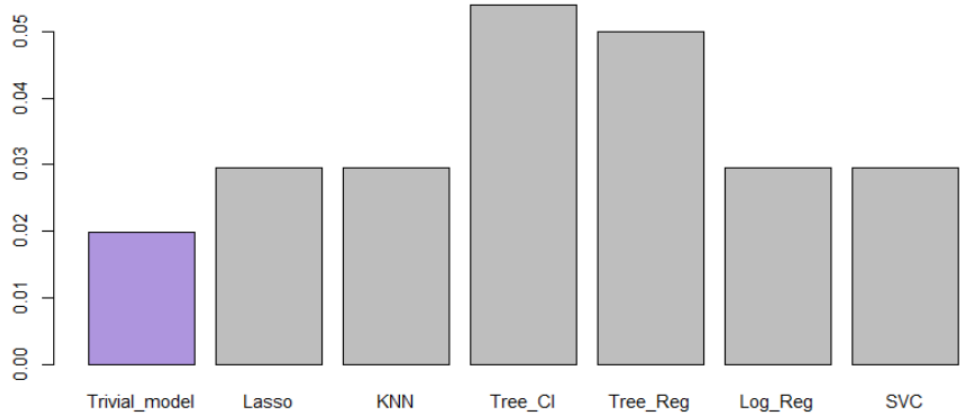
Результаты (не подходят, изначально было ясно)

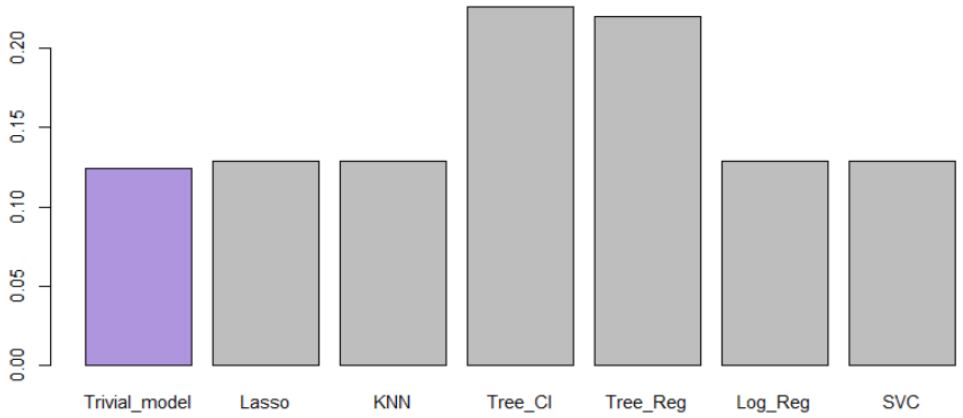
- К11 и К12. Этика и фактическая точность

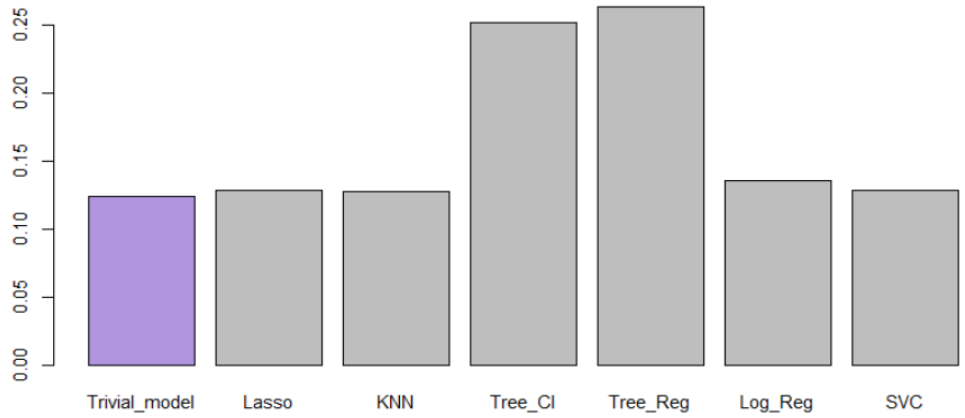
```
Ошибка тривиальной модели для критерия К11 с баллом 1: 0.019862490450725745 (98.0 %)
--------------------------------------------------------------------------------
Word2Vec
--------------------------------------------------------------------------------
Ошибка метода Лассо для критерия К11: 0.029531568228105907 (97.0 %)
Ошибка метода регрессия ближайшими соседями для критерия К11: 0.029531568228105907 (97.0 %)
Ошибка метода классификатор деревом решений для критерия К11: 0.04073319755600815 (95.9 %)
Ошибка метода регрессия деревом решений для критерия К11: 0.045824847250509164 (95.4 %)
Ошибка метода логистическая регрессия для критерия К11: 0.029531568228105907 (97.0 %)
Ошибка метода опорных векторов  для критерия К11: 0.029531568228105907 (97.0 %)
--------------------------------------------------------------------------------
RusVectors
--------------------------------------------------------------------------------
Ошибка метода Лассо для критерия К11: 0.029531568228105907 (97.0 %)
Ошибка метода регрессия ближайшими соседями для критерия К11: 0.029531568228105907 (97.0 %)
Ошибка метода классификатор деревом решений для критерия К11: 0.0539714867617108 (94.6 %)
Ошибка метода регрессия деревом решений для критерия К11: 0.04989816700610998 (95.0 %)
```

Визуализация. R
K11, K12









К1 (проблема текста)

```
Ошибка тривиальной модели для критерия К1 с баллом 1: 0.09856675886346493 (90.1 %)
--------------------------------------------------------------------------
Word2Vec
--------------------------------------------------------------------------
Ошибка метода Лассо для критерия К1: 0.09346733668341708 (90.7 %)
Ошибка метода регрессия ближайшими соседями для критерия К1: 0.09346733668341708 (90.7 %)
Ошибка метода классификатор деревом решений для критерия К1: 0.18592964824120603 (81.4 %)
Ошибка метода регрессия деревом решений для критерия К1: 0.17788944723618091 (82.2 %)
Ошибка метода логистическая регрессия для критерия К1: 0.09346733668341708 (90.7 %)
Ошибка метода опорных векторов  для критерия К1: 0.09346733668341708 (90.7 %)
--------------------------------------------------------------------------
RusVectors
--------------------------------------------------------------------------
Ошибка метода Лассо для критерия К1: 0.09346733668341708 (90.7 %)
Ошибка метода регрессия ближайшими соседями для критерия К1: 0.0964824120603015 (90.4 %)
Ошибка метода классификатор деревом решений для критерия К1: 0.2100502512562814 (79.0 %)
Ошибка метода регрессия деревом решений для критерия К1: 0.17889447236180905 (82.1 %)
```

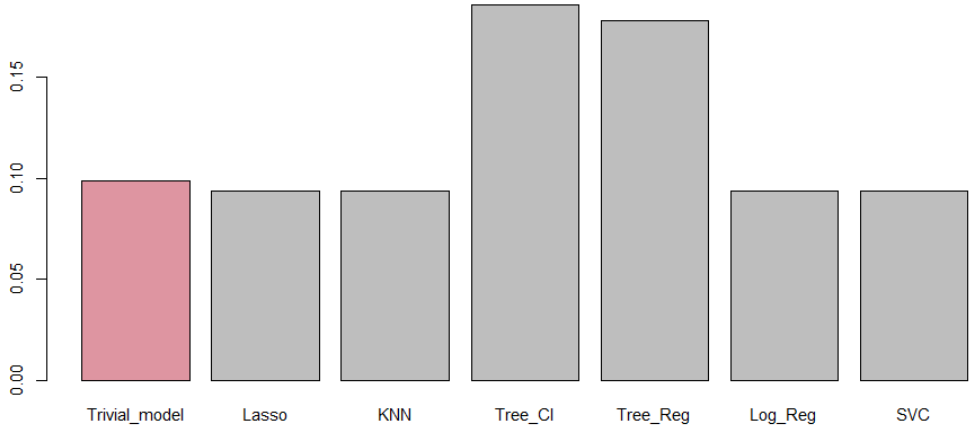
```
Ошибка метода логистическая регрессия для критерия К1: 0.09748743718592964 (90.3 %)
Ошибка метода опорных векторов  для критерия К1: 0.09346733668341708 (90.7 %)
```

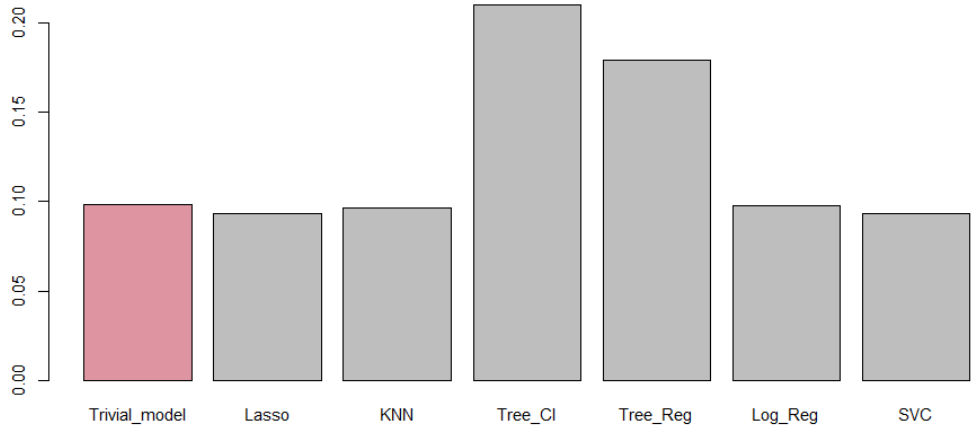
К2 (комментарий к проблеме)

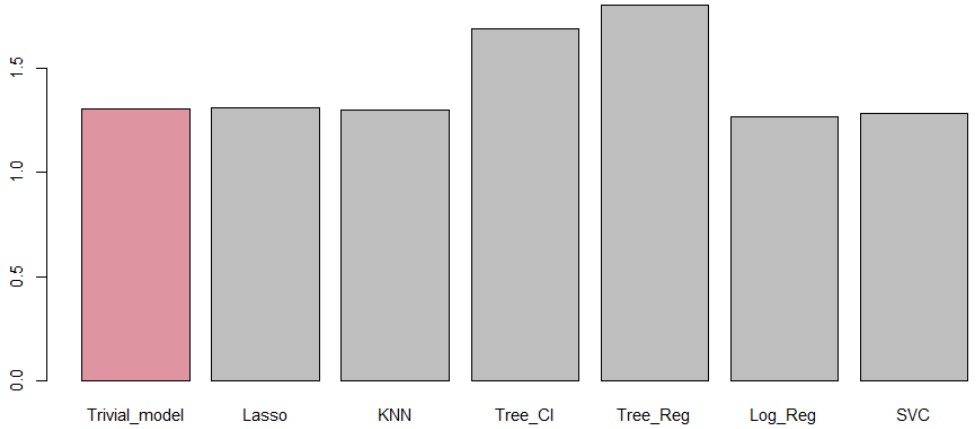
```
Ошибка тривиальной модели для критерия К2 с баллом 4: 1.302766393442623 (20.6 %)
---------------------------------------------------------------------
Word2Vec
---------------------------------------------------------------------
Ошибка метода Лассо для критерия К2: 1.3084016393442623 (20.5 %)
Ошибка метода регрессия ближайшими соседями для критерия К2: 1.298155737704918 (23.8 %)
Ошибка метода классификатор деревом решений для критерия К2: 1.6905737704918034 (23.0 %)
Ошибка метода регрессия деревом решений для критерия К2: 1.8043032786885247 (20.1 %)
```

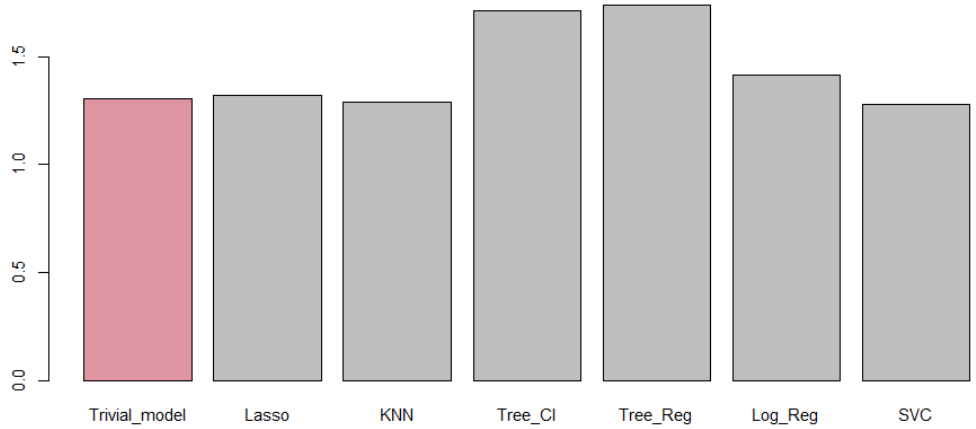
```
Ошибка метода логистическая регрессия для критерия К2: 1.2674180327868851 (33.4 %)
Ошибка метода опорных векторов  для критерия К2: 1.2848360655737705 (34.7 %)
---------------------------------------------------------------------
RusVectors
---------------------------------------------------------------------
Ошибка метода Лассо для критерия К2: 1.3176229508196722 (19.8 %)
Ошибка метода регрессия ближайшими соседями для критерия К2: 1.2889344262295082 (27.4 %)
Ошибка метода классификатор деревом решений для критерия К2: 1.709016393442623 (24.3 %)
Ошибка метода регрессия деревом решений для критерия К2: 1.7377049180327868 (20.3 %)
```

Визуализация. R
K1, K2



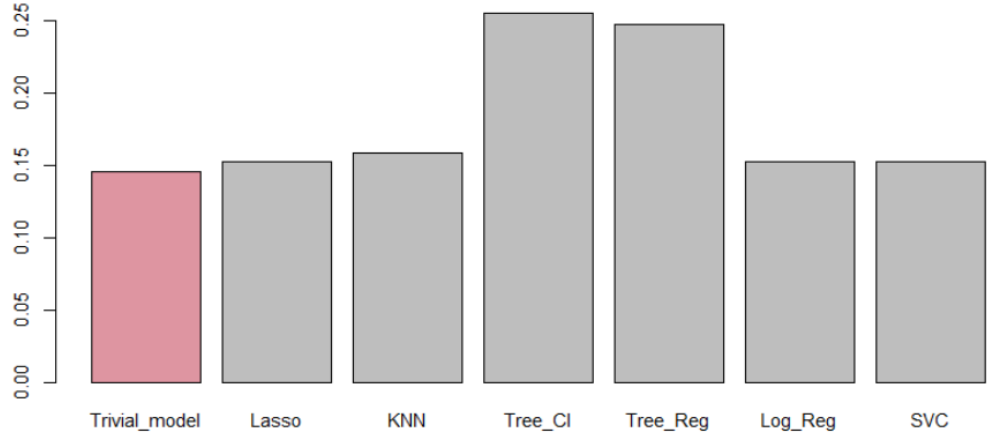


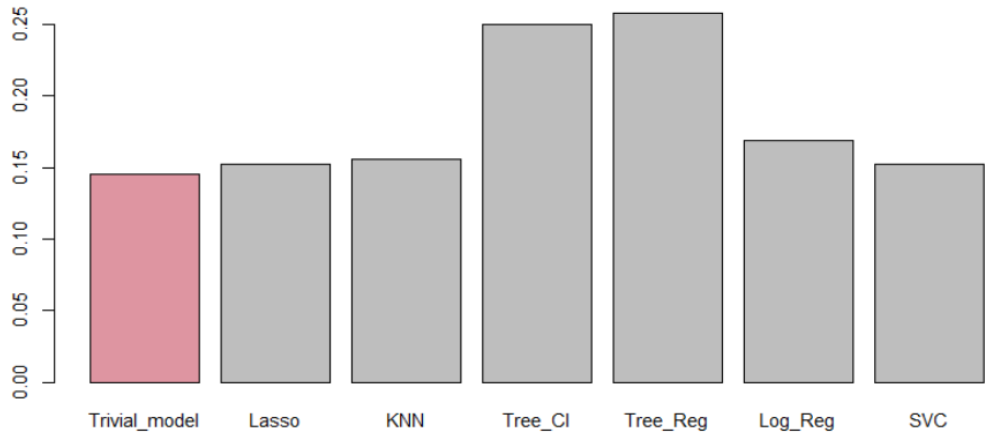


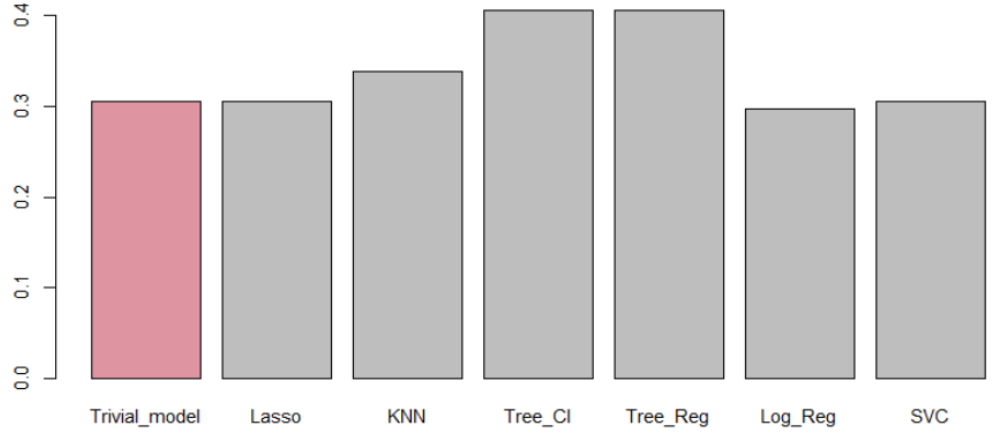


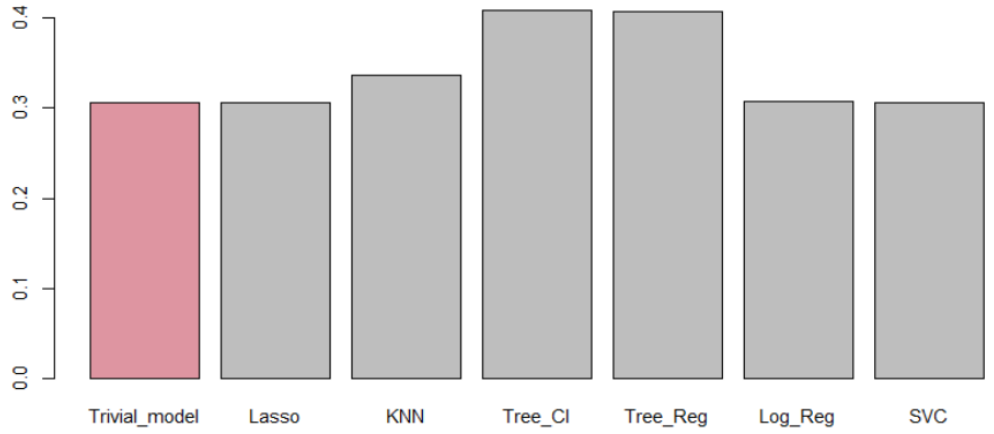
```
Word2Vec
-----------------------------------------------------------------------
Ошибка метода Лассо для критерия К3: 0.15261044176706828 (84.7 %)
Ошибка метода регрессия ближайшими соседями для критерия К3: 0.15863453815261044 (84.1 %)
Ошибка метода классификатор деревом решений для критерия К3: 0.25502008032128515 (74.6 %)
Ошибка метода регрессия деревом решений для критерия К3: 0.2469879518072289 (75.3 %)
Ошибка метода логистическая регрессия для критерия К3: 0.15261044176706828 (84.7 %)
Ошибка метода опорных векторов  для критерия К3: 0.15261044176706828 (84.7 %)
-----------------------------------------------------------------------
RusVectors
-----------------------------------------------------------------------
Ошибка метода Лассо для критерия К3: 0.15261044176706828 (84.7 %)
Ошибка метода регрессия ближайшими соседями для критерия К3: 0.15562248995983935 (84.4 %)
Ошибка метода классификатор деревом решений для критерия К3: 0.25 (75.0 %)
Ошибка метода регрессия деревом решений для критерия К3: 0.2580321285140562 (74.2 %)
```

Визуализация. R
К3, К4

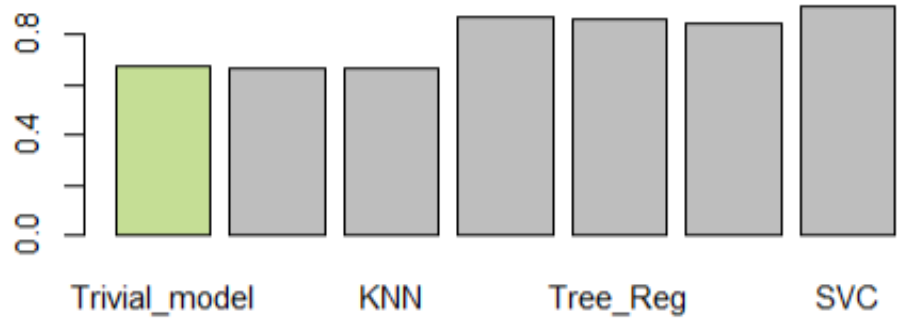


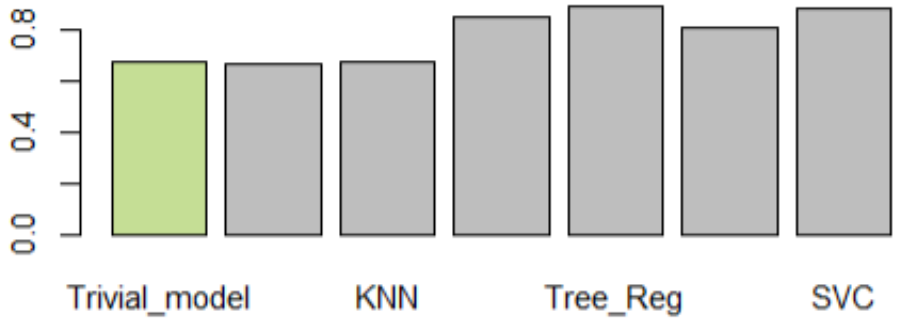


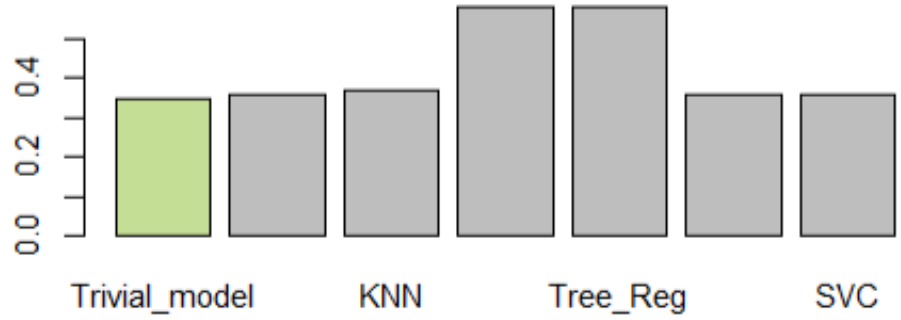


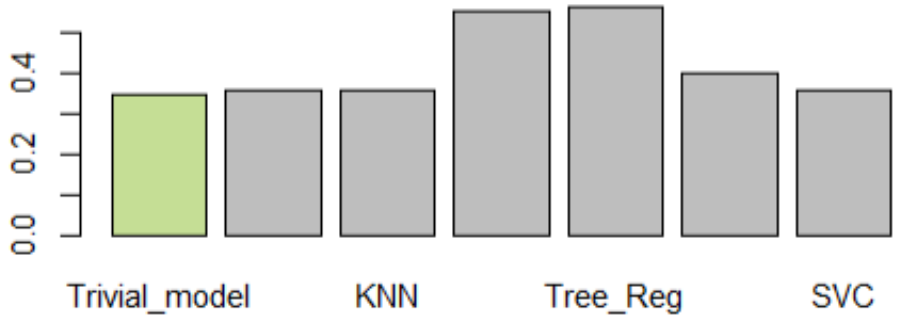


Визуализация. R
K5, K6

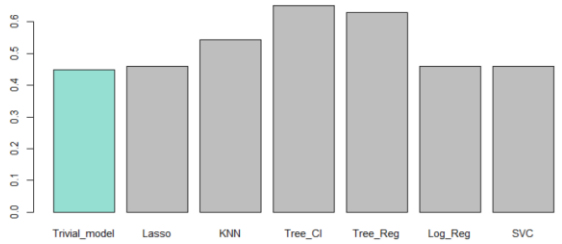


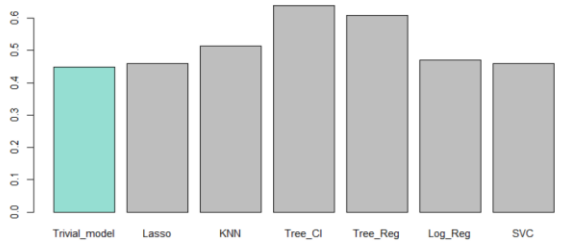


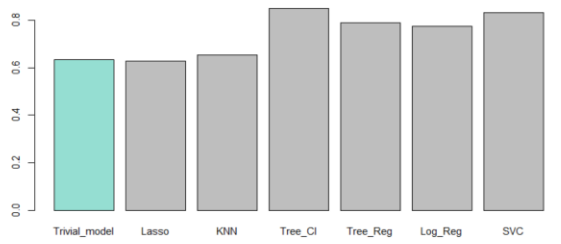


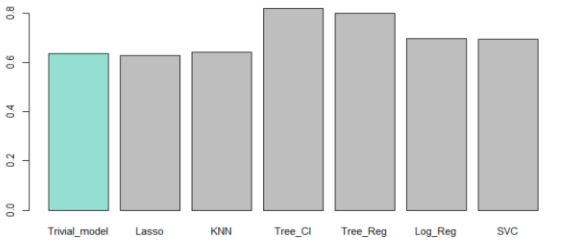


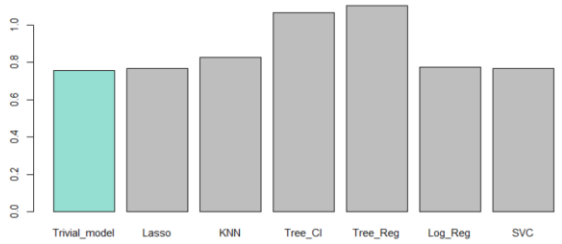
Визуализация. R
K7, K8, K9, 10

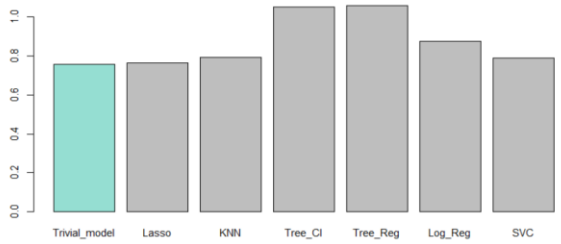


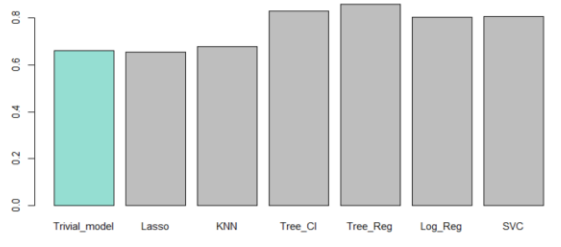


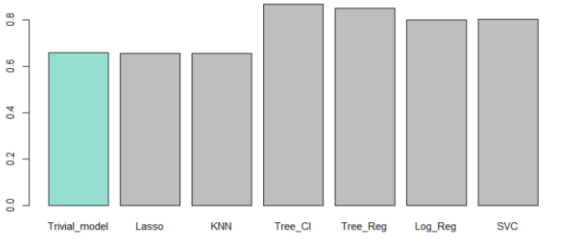










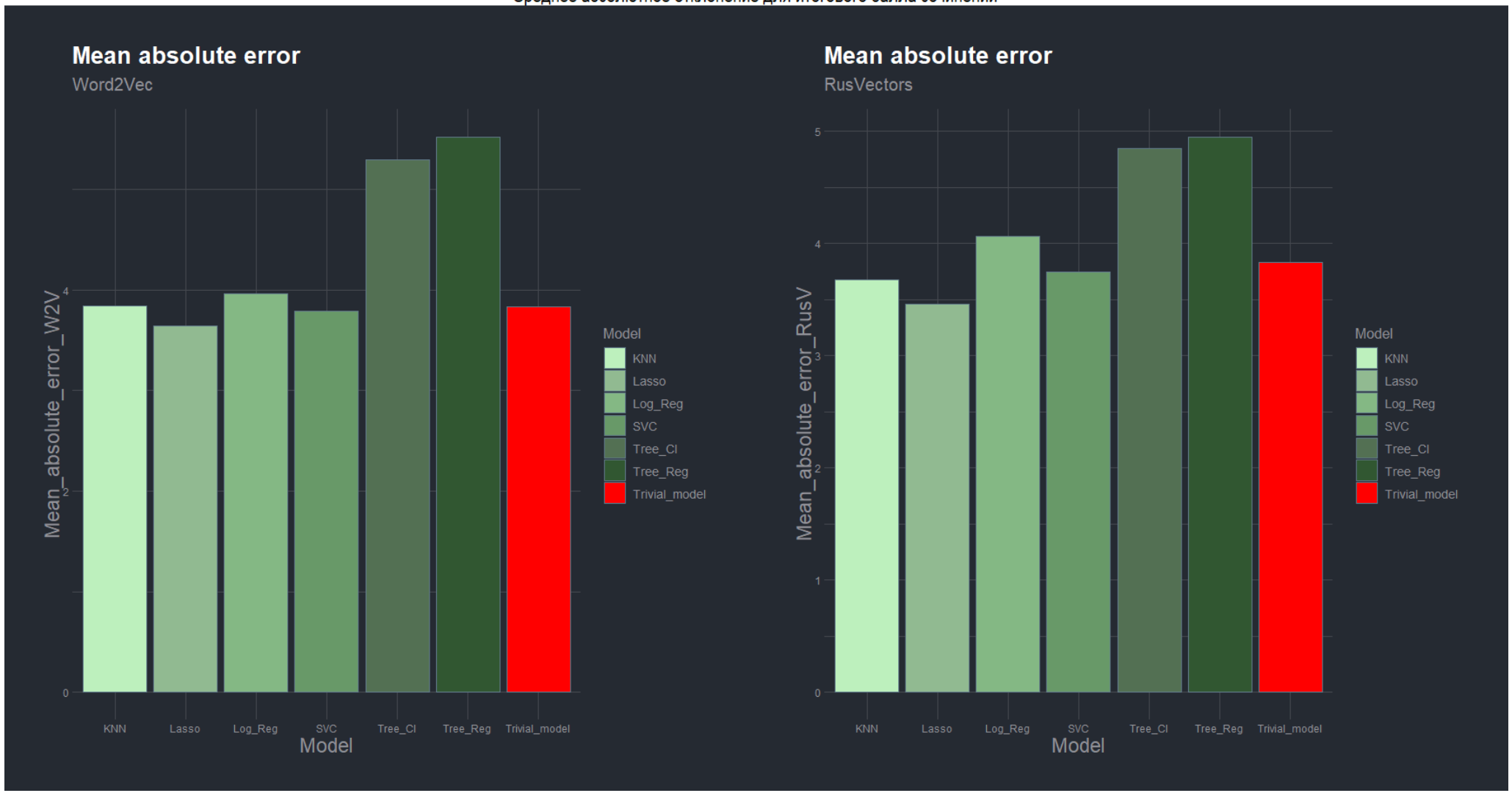


Итоговый балл (сумма)

```
Word2Vec
------------------------------------------------------------------
Ошибка метода Лассо для критерия K13: 3.641237113402062 (8.9 %)
Ошибка метода регрессия ближайшими соседями для критерия K13: 3.841237113402062 (8.2 %)
Ошибка метода классификатор деревом решений для критерия K13: 5.292783505154639 (6.6 %)
Ошибка метода регрессия деревом решений для критерия K13: 5.515463917525773 (6.4 %)
Ошибка метода логистическая регрессия для критерия K13: 3.9608247422680414 (7.0 %)
Ошибка метода опорных векторов  для критерия K13: 3.7876288659793813 (8.5 %)
------------------------------------------------------------------
RusVectors
------------------------------------------------------------------
Ошибка метода Лассо для критерия K13: 3.461855670103093 (11.1 %)
Ошибка метода регрессия ближайшими соседями для критерия K13: 3.672164948453608 (7.6 %)
Ошибка метода классификатор деревом решений для критерия K13: 4.847422680412371 (8.9 %)
Ошибка метода регрессия деревом решений для критерия K13: 4.944329896907217 (6.8 %)
```

Визуализация R

Среднее абсолютное отклонение для итогового балла сочинений



Выводы

- Данные недостаточно разнообразны.
- Векторизация (не все так очевидно, RusVectors не хватило).
- Возможно, нужны дополнительные фичи. Текст автора в первых 4-х на маленьком датасете не очень сильно влияют, абзацы на 0,001 улучшают результат К6. Как гипотеза – проверка орфографии и пунктуации.