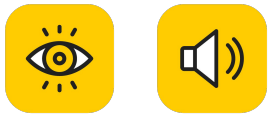
OTUS

Machine Learning. Professional
Защита проекта

otus.ru

Меня хорошо видно && слышно?



Защита проекта

Тема: Анализ результатов IELTS методами ML

Золотарева Наталья

Заместитель начальника контрольно-аналитической лаборатории
ООО ВТФ

Цель проекта

Создание модели способной прогнозировать оценку письменной части языкового экзамена IELTS на основе текста эссе

План работы

1. Исследование данных
2. Генерирование дополнительных признаков
3. Построение моделей
4. Исследовать эффективности сгенерированных признаков

Используемые технологии

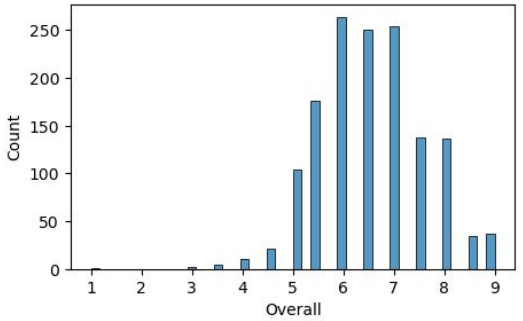
1. spacy для лемматизации и токенизации
2. TF-IDF для векторизации (при построении моделей)
3. предобученный Word2Vec из Gensim для векторизации (для генерации признаков)
4. readability (для генерации признаков)

Данные

Начальные данные:

- 1435 ненулевых строк
- колонки

  Task_Type - тип задания (описание диаграммы или эссе)
  Question - текст вопроса
  Essay - текст эссе
  Overall - итоговая оценка



Overall	
1.0	1
3.0	2
3.5	4
4.0	8
4.5	21
5.0	92
5.5	161
6.0	238
6.5	214
7.0	222
7.5	118
8.0	126
8.5	32
9.0	36

Новые признаки

Word_counter

Количество слов в эссе из списков слов разных уровней A1, A2, B1, B2, C1, C2, AC

https://dictionary.cambridge.org/dictionary/english/

Academic Word List (AWL)

https://simple.wiktionary.org/wiki/Wiktionary:Academic_word_list

Similarity

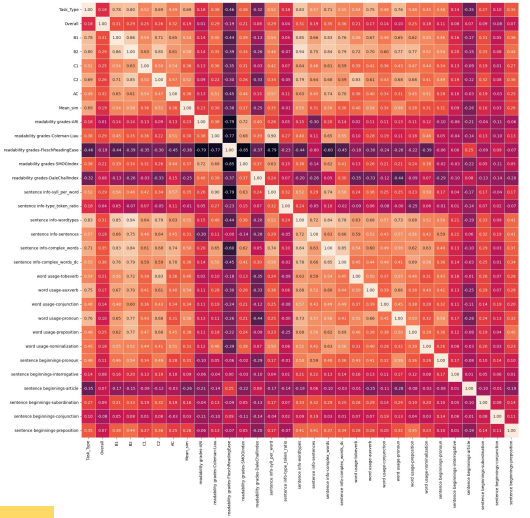
cosine_similarity между эмбеддингом вопроса и суммарным эмбеддингом кадого абзаца, полученного суммой эмбеддингов их предложений

Для получения эмбеддингов использован предобученный word2vec ‘GoogleNews-vectors-negative300.bin.gz’

Данные

readability https://pypi.org/project/readability/

Реализация метрик удобочитаемости, основанных на поверхностных характеристиках (линейные регрессии, основанные на количестве слов, слогов и предложений).



- **automated readability index (ARI)**

$$4.71\left(\frac{\text{characters}}{\text{words}}\right)+0.5\left(\frac{\text{words}}{\text{sentences}}\right)-21.43$$

- **characters per word**

- **LIX** (швед. Läsbarhetsindex) — **индекс удобочитаемости**

$$\text{LIX}=\frac{A}{B}+\frac{C\cdot 100}{A}$$

где

- A — количество слов в тексте,
- B — количество предложений в тексте,
- C — количество слов длиннее 6 букв.

Результат:

readability
23 признаков

слова
5 признаков

“похожесть”
1 признак

Baseline

train 0.7
test 0.15
val 0.15

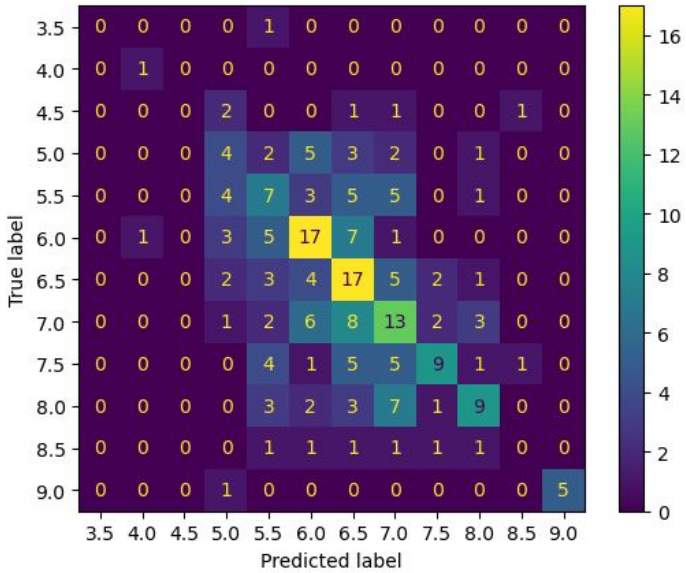
stratify=data['Overall']

	test	val
accuracy_score	0.395	0.381

данные: ['Essay', 'Question', 'Task_Type']

модель: LogisticRegression

	precision	recall	f1-score	support
3.5	0.00	0.00	0.00	1
4.0	0.50	1.00	0.67	1
4.5	0.00	0.00	0.00	5
5.0	0.24	0.24	0.24	17
5.5	0.25	0.28	0.26	25
6.0	0.44	0.50	0.47	34
6.5	0.34	0.50	0.40	34
7.0	0.33	0.37	0.35	35
7.5	0.60	0.35	0.44	26
8.0	0.53	0.36	0.43	25
8.5	0.00	0.00	0.00	6
9.0	1.00	0.83	0.91	6
accuracy			0.38	215
macro avg	0.35	0.37	0.35	215
weighted avg	0.39	0.38	0.37	215



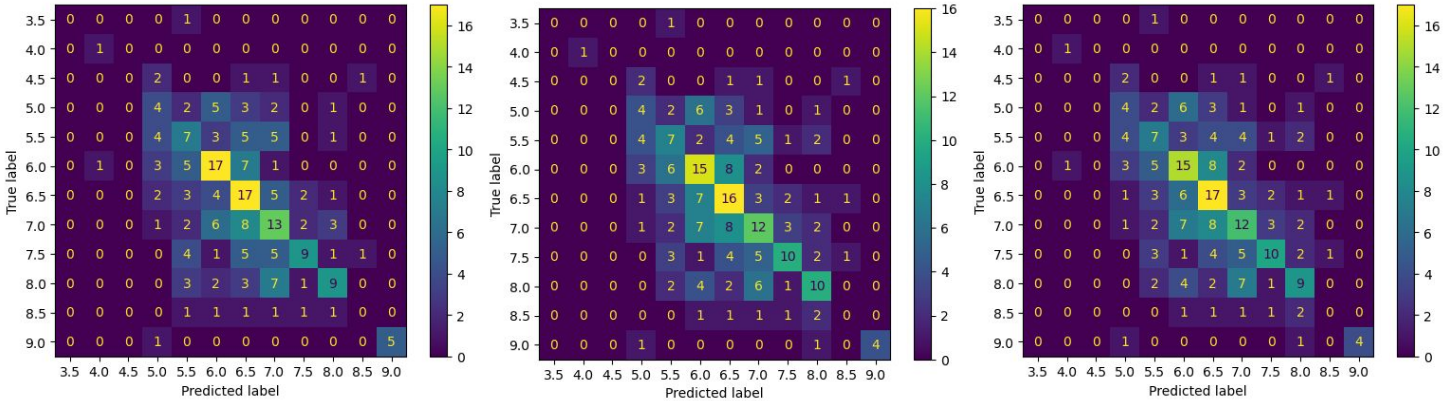
validation set

Влияние новых признаков

данные: все признаки vs. нескоррелированные признаки
модель: LogisticRegression

accuracy_score

	baseline	все данные	без скоррелированных данных
test	0.395	0.395	0.391
val	0.381	0.367	0.367

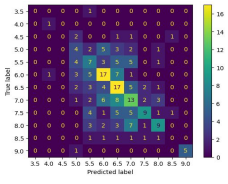


Влияние выбора модели

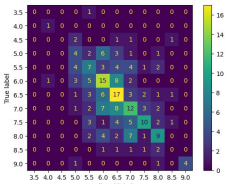
данные: нескоррелированные признаки

accuracy_score

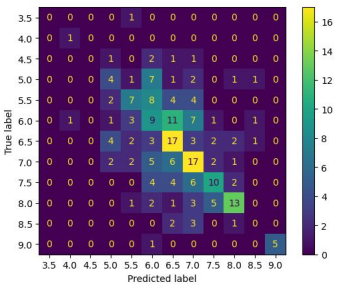
	baseline	LogisticRegression	Gradient Boosting	Random Forest	SVC	CatBoost	Voting Classifier
test	0.395	0.391	0.423	0.433	0.395	0.405	0.405
val	0.381	0.367	0.386	0.451	0.386	0.479	0.414
время обучения	5.6 s	4.23 s	2min 15s	2.71 s	2.25 s	36.7 s	2min 37s



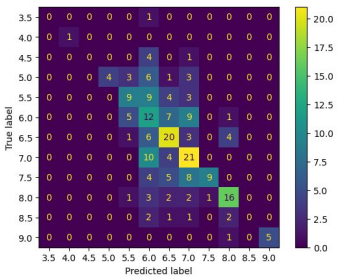
baseline



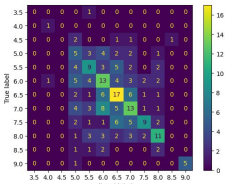
LogisticRegression



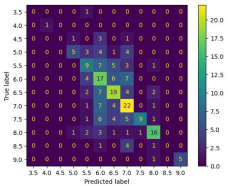
GradientBoosting



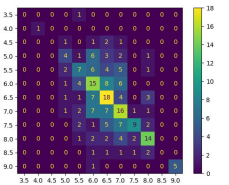
RandomForest



SVC



CatBoost



Voting Classifier

Влияние выбора данных

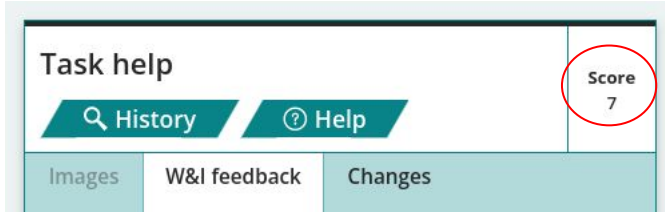
модель: VotingClassifier

accuracy_score	baseline	чистые признаки	все признаки	начальные признаки	простые признаки	сложные признаки
test	0.395	0.405	0.428	0.409	0.414	0.391
val	0.381	0.414	0.423	0.405	0.409	0.437

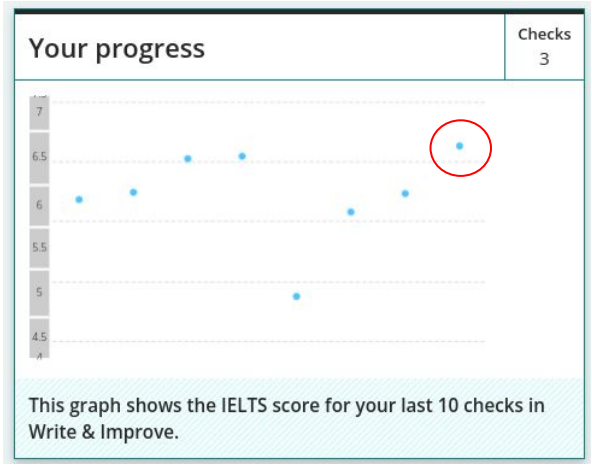
модель: RandomForestClassifier

accuracy_score	baseline	чистые признаки	все признаки	начальные признаки	простые признаки	сложные признаки
test	0.395	0.433	0.386	0.381	0.386	0.391
val	0.381	0.451	0.465	0.367	0.437	0.442

Проверка модели



https://writeandimprove.com



Результаты модели RandomForestClassifier:

```
['6.0', '7.0']
```

Результаты модели VotingClassifier:

```
['7.0', '6.5']
```

Выводы

1. Даже сравнительно простые модели могут дать хорошие результаты при достаточном количестве данных
2. Корреляция в данных иногда не вредна
3. Ансамбли из нескольких моделей позволяют получать более стабильный результат

Спасибо за внимание!