Машинное обучение

Проектная работа
«What's Cooking?»

Чернов Клим

Количество объектов:

Train: 39774

Test: 9944

Количество классов: 20

Метрика: Accuracy

Задача:
мультиклассовая классификация по текстовому признаку


```
{
  "id": 18009,
  "ingredients": [
    "baking powder",
    "eggs",
    "all-purpose flour",
    "raisins",
    "milk",
    "white sugar"
  ]
},
{
  "id": 28583,
  "ingredients": [
    "sugar",
    "egg yolks",
    "corn starch",
    "cream of tartar",
    "bananas",
    "vanilla wafers",
    "milk",
    "vanilla extract",
    "toasted pecans",
    "egg whites",
    "light rum"
  ]
},
{
  "id": 41580,
  "ingredients": [
    "sausage links",
    "fennel bulb",
    "fronds",
    "olive oil",
    "cuban peppers",
    "onions"
  ]
},
```


```
['greek',
 'southern_us'
 'filipino',
 'indian',
 'jamaican',
 'spanish',
 'italian',
 'mexican',
 'chinese',
 'british',
 'thai',
 'vietnamese',
 'cajun_creole
 'brazilian',
 'french',
 'japanese',
 'irish',
 'korean',
 'moroccan',
 'russian']
```

Leaderboard

#	Score
1	0.83216
2	0.82853
3	0.82461
4	0.82441
5	0.82300
6	0.82260
7	0.82210
8	0.82089
9	0.82079
10	0.82069

217	0.79997
218	0.79987
219	0.79967
220	0.79957
221	0.79947
222	0.79937
223	0.79907
224	0.79907
225	0.79897
226	0.79897

Обработка признака

```
df.loc[39772, 'ingredients']
```

```
"['boneless chicken skinless thigh', 'minced garlic', 'steamed white rice', 'baking powder', 'corn starch', 'dark soy sauce',
'kosher salt', 'peanuts', 'flour', 'scallions', 'Chinese rice vinegar', 'vodka', 'fresh ginger', 'egg whites', 'broccoli', 'toa
sted sesame seeds', 'sugar', 'store bought low sodium chicken stock', 'baking soda', 'Shaoxing wine', 'oil']"
```

↓ Объединение в текст

```
df.loc[39772, 'text']
```

```
'boneless chicken skinless thigh minced garlic steamed white rice baking powder corn starch dark soy sauce kosher salt peanuts
flour scallions Chinese rice vinegar vodka fresh ginger egg whites broccoli toasted sesame seeds sugar store bought low sodium
chicken stock baking soda Shaoxing wine oil'
```

↓ Лемматизация существительных, удаление чисел и стоп-слов, приведение к нижнему регистру

```
df.loc[39772, 'tokens']
```

```
'boneless chicken skinless thigh minced garlic steamed white rice baking powder corn starch dark soy sauce kosher salt peanut f
lour scallion chinese rice vinegar vodka fresh ginger egg white broccoli toasted sesame seed sugar store bought low sodium chic
ken stock baking soda shaoxing wine oil'
```

Pipeline['nlp']: TF-IDF

Первая часть пайплайна: создание матрицы TF-IDF

```
TfidfVectorizer(
                min_df=2,
                max_df=0.5,
                stop_words='english'
                  )
```

	abalone	abura	acai	accent	achiote	acid	acinus	ackee	acorn	active	...	yum	yuzu	yuzukosho	zero	zest	zesty	zinfandel	ziti	zucchini	épi
0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.000000	0.0	0.0	0.0	
1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.000000	0.0	0.0	0.0	
2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.000000	0.0	0.0	0.0	
3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.000000	0.0	0.0	0.0	
4	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.000000	0.0	0.0	0.0	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	
39769	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.000000	0.0	0.0	0.0	
39770	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.347271	0.0	0.0	0.0	
39771	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.000000	0.0	0.0	0.0	
39772	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.000000	0.0	0.0	0.0	
39773	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.000000	0.0	0.0	0.0	

39774 rows × 2178 columns

```
"['KRAFT Zesty Italian Dressing', 'purple onion', 'broccoli florets', 'rotini', 'pitted black olives', 'Kraft Grated Parmesan C
heese', 'red pepper']"
```

Pipeline['model']: варианты модели

Вторая часть пайплайна: модель

Логистическая регрессия:

Номер	Описание	CV Train	CV Test	Kaggle Score
1 (baseline)	LogisticRegression()	0.81522	**0.77654**	0.78258
2	LogisticRegression(C=2.7056825281253234)	0.84183	**0.784**	0.786

Решающие деревья:

3	DecisionTreeClassifier()	0.99947	0.61761	0.63093
4	DecisionTreeClassifier(min_samples_split=65, min_samples_leaf=7, max_leaf_nodes=8018)	0.70794	0.63182	0.63837

Случайный лес:

5	RandomForestClassifier()	0.99947	0.74536	0.75764
6	RandomForestClassifier(max_leaf_nodes=19367, n_estimators=194)	0.99947	0.74767	0.75834
7	RandomForestClassifier(min_samples_split=65, min_samples_leaf=7, max_leaf_nodes=8018, n_estimators=194)	0.70912	0.67258	0.68332

Pipeline['model']: варианты модели

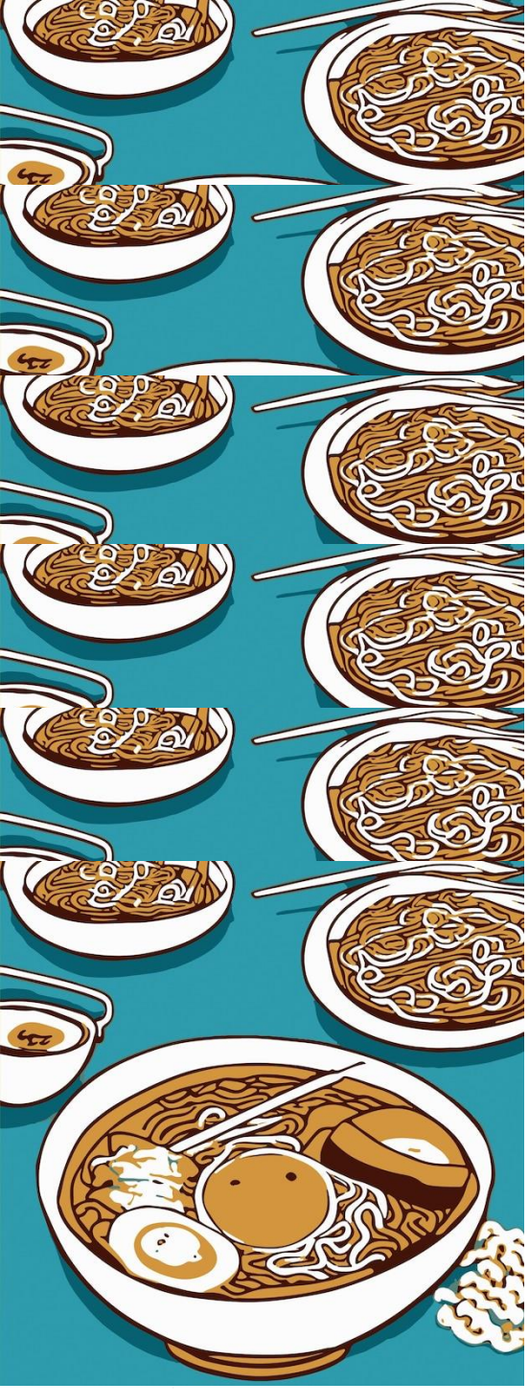
Вторая часть пайплайна: модель

Бустинги:

Номер	Описание	CV Train	CV Test	Kaggle Score
8	GradientBoostingClassifier()	0.85338	0.73171	0.74155
9	LightGBMClassifier()	0.98466	0.77719	0.7858
10	LGBMClassifier(learning_rate=0.038, max_depth=22, n_estimators=279)	0.98607	0.78365	0.78861

Гибриды:

11	LogisticRegression(C=2.7056825281253234) * 0.5 + LGBMClassifier(learning_rate=0.038, max_depth=22, n_estimators=279) * 0.5	0.95702	**0.79715**	0.79977



Выводы

Лучшая модель:

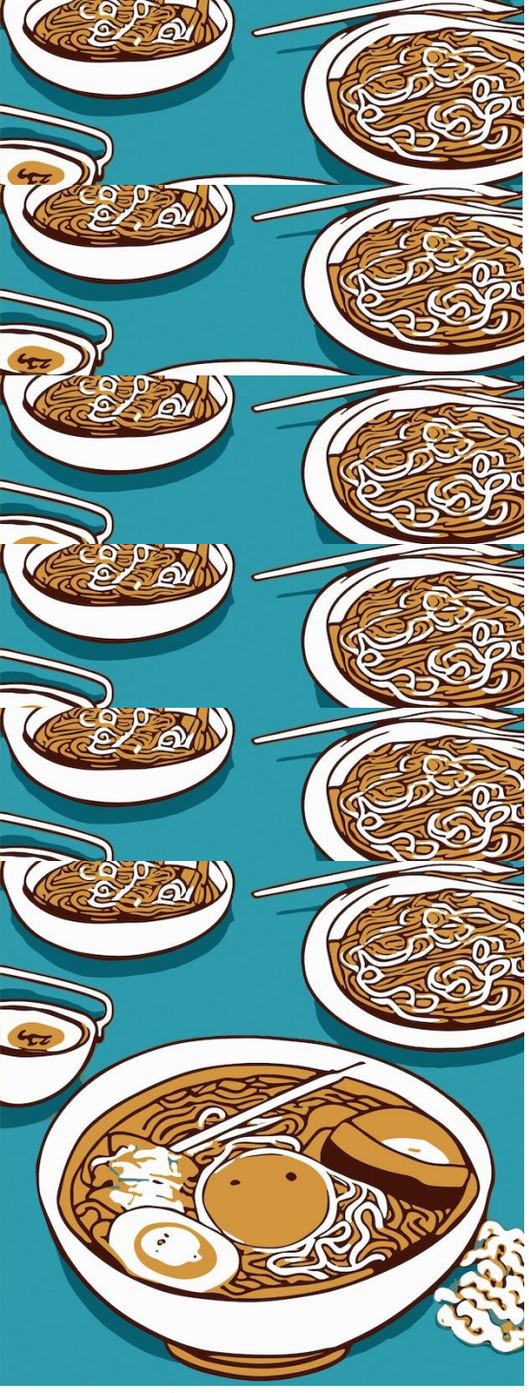
```
p_logit = Pipeline([('nlp', TfidfVectorizer(min_df=2, max_df=0.5, stop_words='english')),
                    ('model', LogisticRegression(n_jobs=-1, C=2.7056825281253234))])

p_LGBM = Pipeline([('nlp', TfidfVectorizer(min_df=2, max_df=0.5, stop_words='english')),
                   ('model', lgb.LGBMClassifier( n_estimators=279,max_depth=22, learning_rate=0.03803766047009802))])

p_logit.fit(Xtrain, ytrain)
p_LGBM.fit(Xtrain, ytrain)
proba_logit = p_logit.predict_proba(Xtest)
proba_LGBM = p_LGBM.predict_proba(Xtest)
proba_hybrid = proba_logit * 0.5 + proba_LGBM * 0.5
predict_hybrid = np.array([p_logit.classes_[proba_hybrid[i].argmax()] for i in range(proba_hybrid.shape[0])])
```

Попытки:

Описание	Причина отказа
Глубокая обработка текста: взятие n-грамм, noun phrases, ключевых слов	Слишком сложно
Сокращение размерности: SelectKBest(), SelectPercentile()	Ухудшение скора
Подбор гиперпараметров GradientBoostingClassifier()	Слишком долго
Обучение CatBoostClassifier()	Слишком долго

Ограничения:

Не были рассмотрены другие метрики (precision, recall, др.) и осталось неизвестно качество предикта каждого класса.



СПАСИБО ЗА ВНИМАНИЕ!