Прогнозирование статуса студента

Кочетков Олег Александрович

Предварительная обработка данных

После первичного знакомство с предоставленными данными, обнаружилась необходимость заполнения пропусков, преобразование текстовых значений в числовые, странная корреляция с ID

0	ID	13584 non-null	int64
1	Код_группы	13584 non-null	int64
2	Год_Поступления	13584 non-null	int64
3	Пол	13577 non-null	object
4	Основания	13584 non-null	object
5	Изучаемый_Язык	12758 non-null	object
6	Дата_Рождения	13584 non-null	object

```
X['ID'] %= X.Код_группы
    for c in ['Общежитие', 'Село', 'Иностранец']:
        X[c] = X[c].map({0:-1, 1:1}).fillna(0).astype(np.int8)
            X['Пол'] = X.Пол.map({'Муж':-1,
'Жен':1}).fillna(0).astype(np.int8)
X['Дата_Рождения'] = pd.to_datetime(X.Дата_Рождения
).dt.year.astype(np.uint16)
```

```
1 df.corr()
```

	ID	Код_группы	Год_Поступления	Год_Окончания_УЗ	Пособие
ID	1.000000	0.764826	0.695542	0.403148	NaN
Код_группы	0.764826	1.000000	0.579809	0.393579	NaN
Год_Поступления	0.695542	0.579809	1.000000	0.444285	NaN
Год_Окончания_УЗ	0.403148	0.393579	0.444285	1.000000	NaN

Предварительная обработка данных

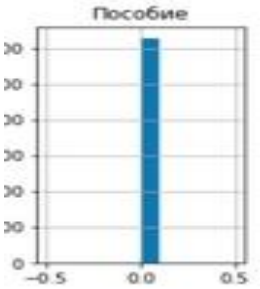
Далее проводилась попытка свести средний балл аттестата (явно различающейся по шкалам) к одной относительной шкале и бинаризация некоторых признаков

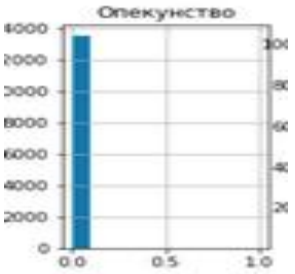
```
for cntri in [((X.СрБаллАттестата>0) & (X.СрБаллАттестата<6)),
              ((X.СрБаллАттестата>=8) & (X.СрБаллАттестата<21)),
              ((X.СрБаллАттестата>20) & (X.СрБаллАттестата<30)),
              ((X.СрБаллАттестата>=30) & (X.СрБаллАттестата<=100)),
              ((X.СрБаллАттестата>1000))]:
    mi = X.loc[cntri,'СрБаллАттестата'].min()
    ma = X.loc[cntri,'СрБаллАттестата'].max()
    X.loc[cntri, 'СрБаллАттестата'] = X.loc[cntri,
                                'СрБаллАттестата'].map(lambda x: (x - mi) / ma)
X = pd.get_dummies(X, prefix='Осн', columns=['Основания'])
X = pd.get_dummies(X, prefix='CF', columns=['КодФакультета'])
```

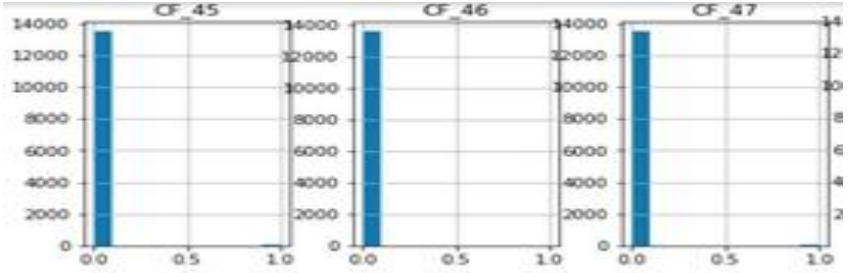
Предварительная обработка данных

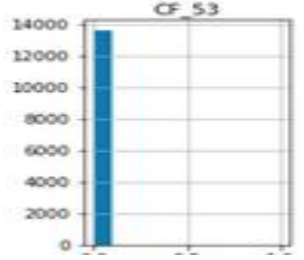
Избавимся от несущественных признаков

```
    X.drop(columns=['Пособие',
'Опекунство',
                    'CF_37', 'CF_46',
'CF_47', 'CF_49', 'CF_53',
                    ],
          inplace=True)
```









Предварительная обработка данных

Для получения признаков из текстовых данных, остальные колонки объединялись в один текст и проводилась векторизация текста, а затем уменьшение размерности этих векторов

```
X['text'] = df[['Страна_ПП', 'Где_Находится_УЗ', 'Уч_Заведение', 'Изучаемый_Язык',
                'Страна_Родители']].fillna('X').apply( lambda x: ' '.join(map(str, x)), axis=1)
model = SentenceTransformer('all-MiniLM-L6-v2')
textv = model.encode(X['text'])
```

```
pca = PCA(n_components=11, svd_solver='full', random_state=42)
pca.fit(pd.DataFrame(textv.to_list()))
pca_train = pca.transform(pd.DataFrame(textv.to_list()))
X = pd.concat([X, pd.DataFrame(pca_train)], axis=1)
X.drop(columns=['text'], inplace=True)
```

Обучение модели

Прежде чем приступить к обучению модели, проведем:

- разбиение данных на обучающую и тренировочную выборки
- расширение данных для выравнивания классов

В качестве модели для получения прогнозов, использовался CatBoostClassifier

```
X_train, X_test, y_train, y_test = train_test_split(
                                X, y, test_size=0.15, random_state=0)
ada = ADASYN( sampling_strategy='all', random_state=0,
                        n_neighbors=7, n_jobs=-1)
X_b, y_b = ada.fit_resample(X_train, y_train)
cbc = CatBoostClassifier(random_state=42, iterations=2000, task_type='GPU',
                        learning_rate = 0.033, depth=9, l2_leaf_reg=4,
                         early_stopping_rounds = 50)
cbc.fit(X_b, y_b,  eval_set = (X_test, y_test),verbose=100)
```

```
1 y_train.value_counts()
```

```
 4     6990
 3     4027
-1      529
```

```
6 y_b.value_counts()
```

```
/usr/local/lib/python3.
  FutureWarning,
-1     7045
 4     6990
 3     6774
```

Спасибо за внимание!