Московский государственный технический университет им. Н. Э. Баумана Факультет «Информатика и системы управления»

Кафедра «Системы обработки информации и управления» Курс «Технологии машинного обучения»

Отчет по лабораторной работе N24

Подготовка обучающей и тестовой выборки, кросс-валидация и подбор гиперпараметров на примере метода ближайших соседей.

Группа: РТ5-61

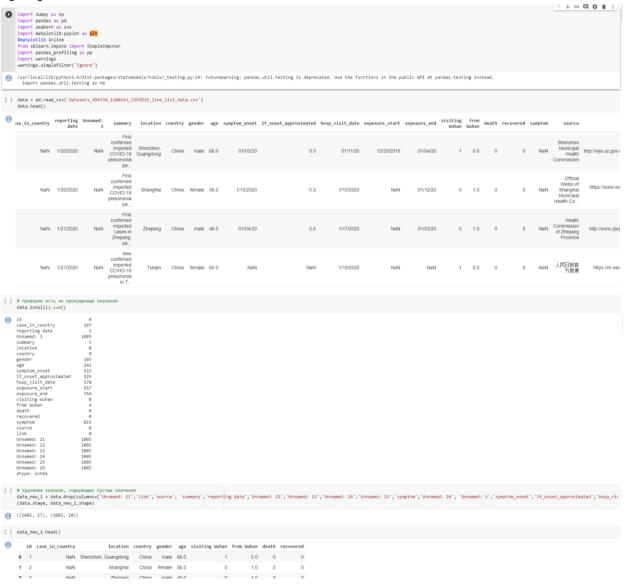
Студент: Коржов С.Ю.

Преподаватель: Гапанюк Ю.Е.

Цель лабораторной работы: изучение сложных способов подготовки выборки и подбора гиперпараметров на примере метода ближайших соседей. **Задание:**

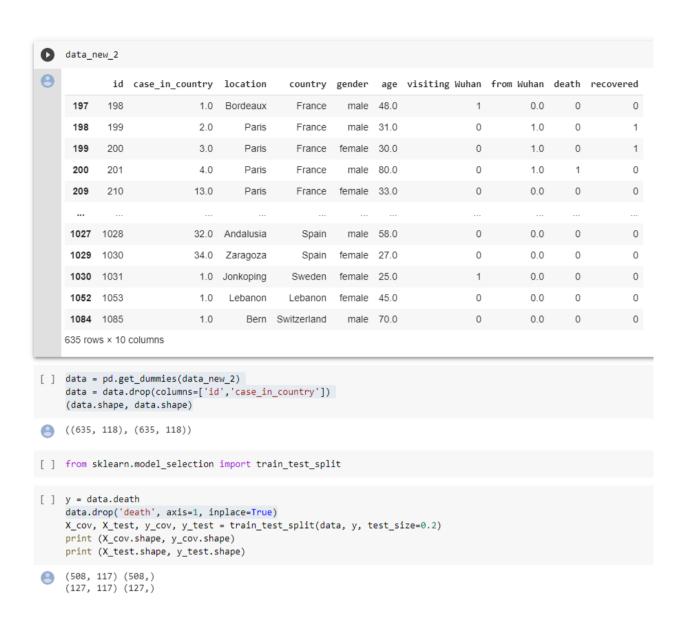
- 1. Выберите набор данных (датасет) для решения задачи классификации или регрессии.
- 2. С использованием метода train_test_split разделите выборку на обучающую и тестовую.
- 3. Обучите модель ближайших соседей для произвольно заданного гиперпараметра К. Оцените качество модели с помощью подходящих для задачи метрик.
- 4. Постройте модель и оцените качество модели с использованием кроссвалилации.
- 5. Произведите подбор гиперпараметра К с использованием GridSearchCV и кросс-валидации.

Текст программы и экранные формы с примерами выполнения программы:



```
[ ] data_new_1.isnull().sum()
e id
    case_in_country
                       197
    location
                       0
    country
                        0
    gender
                       183
    age
                       242
    visiting Wuhan
                        Θ
    from Wuhan
                        4
    death
                         0
    recovered
                         0
    dtype: int64
[ ] # Удаление строки, содержащих пустые значения
    data_new_2 = data_new_1.dropna(axis=0,how='any',subset=['case_in_country','gender', 'age'])
    (data_new_1.shape, data_new_2.shape)
((1085, 10), (635, 10))
[ ] data_new_2.head()
8
           id case_in_country location country gender age visiting Wuhan from Wuhan
                                                                                             death recovered
     197 198
                           1.0 Bordeaux
                                                    male 48.0
                                                                                     0.0
                                          France
     198
         199
                           2.0
                                   Paris
                                          France
                                                    male 31.0
                                                                           0
                                                                                     1.0
                                                                                                 0
                                                                                                     02/12/20
                           3.0
                                   Paris
                                          France
                                                  female 30.0
                                                                                                     02/12/20
     199 200
                                                                           0
                                                                                     1.0
                                                                                                0
     200 201
                           4.0
                                   Paris
                                          France
                                                    male 80.0
                                                                           0
                                                                                     1.0
                                                                                         2/14/2020
                                                                                                           0
     209 210
                          13.0
                                   Paris
                                          France
                                                  female 33.0
                                                                                     0.0
                                                                                                           0
[ ] data_new_2.shape[0]
635
[ ] data_new_2['death'] = data_new_2['death'].apply(lambda x: 0 if x=='0' else 1)
```

[] data_new_2['recovered'] = data_new_2['recovered'].apply(lambda x: 0 if x=='0' else 1)



```
Codyveние модели

[ ] from sklearn.neighbors import KNeighborsClassifier

[ ] KNeighborsClassifierObj = KNeighborsClassifier(n_neighbors=10)

[ ] KNeighborsClassifierObj.fit(X_cov, y_cov)

② KNeighborsClassifier(algorithm='auto', leaf_size=30, metric='minkowski', metric_params=None, n_jobs=None, n_neighbors=10, p=2, weights='uniform')

[ ] y_predicted = KNeighborsClassifierObj.predict(X_test)

Метрика качества

[ ] from sklearn.metrics import accuracy_score, balanced_accuracy_score, precision_score, f1_score, classification_report

[ ] accuracy_score(y_test, y_predicted)

② 0.9606299212598425

[ ] precision_score(y_test, y_predicted)

③ 0.0
```

Смертность не зависит от других параметров.

Кросс-валидация

```
[ ] from sklearn.model_selection import GridSearchCV
[ ] n_range = np.array(range(5,55,5))
    tuned_parameters = [{'n_neighbors': n_range}]
[ ] clf gs = GridSearchCV(KNeighborsClassifier(), tuned parameters, cv=5, scoring='f1 weighted')
[ ] clf_gs.fit(X_cov, y_cov)
GridSearchCV(cv=5, error score=nan,
                 estimator=KNeighborsClassifier(algorithm='auto', leaf_size=30,
                                               metric='minkowski',
                                               metric_params=None, n_jobs=None,
                                               n_neighbors=5, p=2,
                                               weights='uniform'),
                 iid='deprecated', n_jobs=None,
                 param_grid=[{'n_neighbors': array([ 5, 10, 15, 20, 25, 30, 35, 40, 45, 50])}],
                 pre_dispatch='2*n_jobs', refit=True, return_train_score=False,
                 scoring='f1 weighted', verbose=0)
[ ] clf_gs.best_params_
   {'n_neighbors': 5}
[ ] clf_gs.best_score_
0.9588742014124028
                plt.plot(n_range, clf_gs.cv_results_['mean_test_score'])
                [<matplotlib.lines.Line2D at 0x7f1752b97278>]
                 1.00
                 0.98
                 0.96
                 0.94
                 0.92
                                       20
                            10
                                                  30
                                                                        50
```