Московский Государственный Технический Университет им. Н. Э. Баумана

Факультет «Информатика и Системы управления»

Кафедра «Автоматизированные системы обработки информации и управления»

Дисциплина «Технологии машинного обучения»

Отчёт по рубежному контролю №1

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

Выполнил:

Студент группы ИУ5ц-83Б

Костников И.А.

Преподаватель:

Гапанюк Ю.Е.

Москва, 2020 г.

1 Цель работы

Изучение способов подготовки выборки и подбора гиперпараметров на примере метода ближайших соседей.

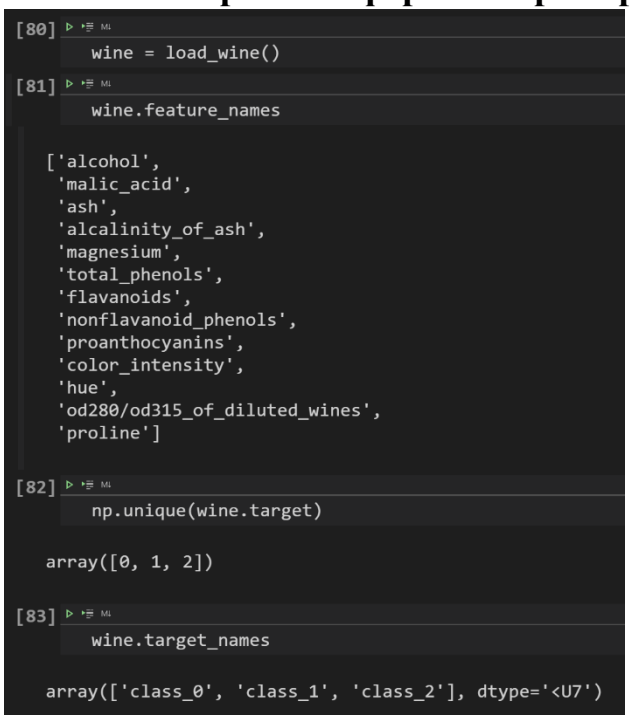
2 Краткое описание

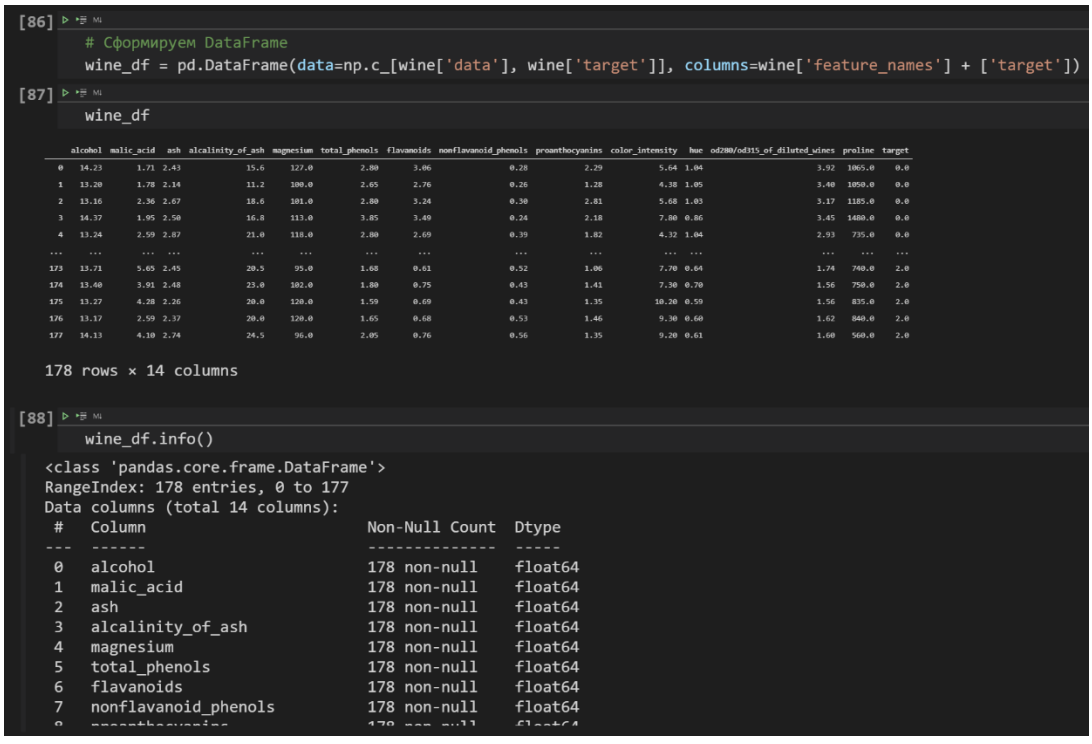
Подготовка данных

3 Текст программы

Текст программы представлена во втором файле (Lab3wine)

4 Экранные формы с примерами выполнения программы.



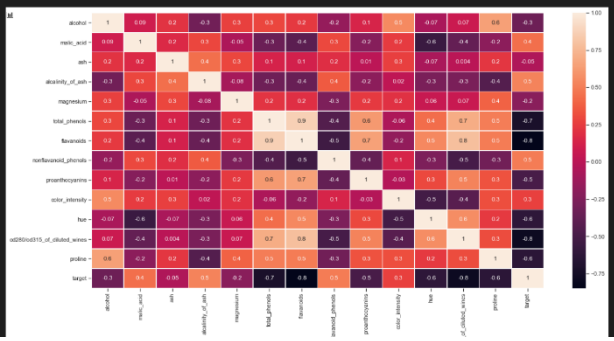


[89]
```
wine_df.describe()
```

	alcohol	malic_acid	ash	alcalinity_of_ash	magnesium	total_phenols	flavanoids	nonflavanoid_phenols	proanthocyanins	color_intensity	hue	od280/od315_of_diluted_wines	proline	target
count	178.000000	178.000000	178.000000	178.000000	178.000000	178.000000	178.000000	178.000000	178.000000	178.000000	178.000000	178.000000	178.000000	178.000000
mean	13.000618	2.336348	2.366517	19.494944	99.741573	2.295112	2.029270	0.361854	1.590899	5.058090	0.957449	2.611685	746.893258	0.938202
std	0.811827	1.117146	0.274344	3.339564	14.282484	0.625851	0.998859	0.124453	0.572359	2.318286	0.228572	0.709913	314.907474	0.775035
min	11.030000	0.740000	1.360000	10.600000	70.000000	0.980000	0.340000	0.130000	0.410000	1.280000	0.480000	1.270000	278.000000	0.000000
25%	12.362500	1.602500	2.210000	17.200000	88.000000	1.742500	1.205000	0.270000	1.250000	3.220000	0.782500	1.937500	500.500000	0.000000
50%	13.050000	1.865000	2.360000	19.500000	98.000000	2.355000	2.135000	0.340000	1.555000	4.690000	0.965000	2.780000	673.500000	1.000000
75%	13.677500	3.082500	2.557500	21.500000	107.000000	2.800000	2.875000	0.437500	1.950000	6.200000	1.120000	3.170000	985.000000	2.000000
max	14.830000	5.800000	3.230000	30.000000	162.000000	3.880000	5.080000	0.660000	3.580000	13.000000	1.710000	4.000000	1680.000000	2.000000

[90]
```
fig, ax = plt.subplots(figsize=(20,10))
sns.heatmap(wine_df.corr(), annot = True, fmt='.1g', linewidths=.5, ax=ax)
```

```
<AxesSubplot:>
```



Построение базовой модели на основе ближайших соседей

[97]
```
# 2 ближайших соседа
cl1_1 = KNeighborsClassifier(n_neighbors=2)
cl1_1.fit(wine_X_train, wine_y_train)
target1_1 = cl1_1.predict(wine_X_test)
len(target1_1), target1_1
```

```
(89,
 array([0, 1, 2, 1, 0, 1, 2, 0, 1, 1, 0, 1, 1, 0, 2, 1, 1, 1, 1, 0, 0, 1,
        1, 0, 0, 1, 0, 0, 0, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 0, 0, 1, 1, 0,
        0, 1, 0, 0, 0, 0, 0, 1, 1, 1, 0, 1, 0, 1, 1, 1, 1, 0, 1, 1, 2, 0,
        1, 0, 1, 1, 1, 0, 1, 0, 1, 1, 2, 2, 0, 1, 0, 0, 1, 0, 1, 1, 1, 0,
        1]))
```

[98]
```
# 10 ближайших соседа
cl1_2 = KNeighborsClassifier(n_neighbors=10)
cl1_2.fit(wine_X_train, wine_y_train)
target1_2 = cl1_2.predict(wine_X_test)
len(target1_2), target1_2
```

```
(89,
 array([1, 1, 2, 1, 0, 1, 1, 0, 1, 1, 0, 1, 1, 0, 1, 1, 1, 0, 1, 0, 0, 1,
        2, 0, 2, 1, 0, 0, 0, 2, 1, 2, 1, 0, 2, 1, 1, 1, 1, 0, 0, 1, 1, 2,
        0, 2, 2, 0, 0, 0, 1, 1, 1, 0, 1, 0, 0, 1, 1, 1, 1, 0, 2, 2, 1, 0,
        1, 0, 1, 2, 1, 1, 1, 2, 1, 1, 2, 1, 0, 2, 1, 0, 1, 0, 1, 1, 1, 0,
        1]))
```

Метрики качества классификации

[99]
```
accuracy_score(wine_y_test, target1_1)
```

```
0.6404494382022472
```

[100]
```
accuracy_score(wine_y_test, target1_2)
```

```
0.6629213483146067
```

Произведите подбор гиперпараметра K с использованием GridSearchCV

```
[101]
print(classification_report(wine_y_test, target1_2))
```

```
              precision    recall  f1-score   support

           0       0.93      0.79      0.85        33
           1       0.61      0.82      0.70        34
           2       0.33      0.23      0.27        22

    accuracy                           0.66        89
   macro avg       0.62      0.61      0.61        89
weighted avg       0.66      0.66      0.65        89
```

```
[102]
def print_gridResults(grid):
    print(f'Подобранный параметр: {grid.best_params_}')
    print(f'Оценка при подобранном параметре: {grid.best_score_}')
    return [grid.best_params_, grid.best_score_]
```

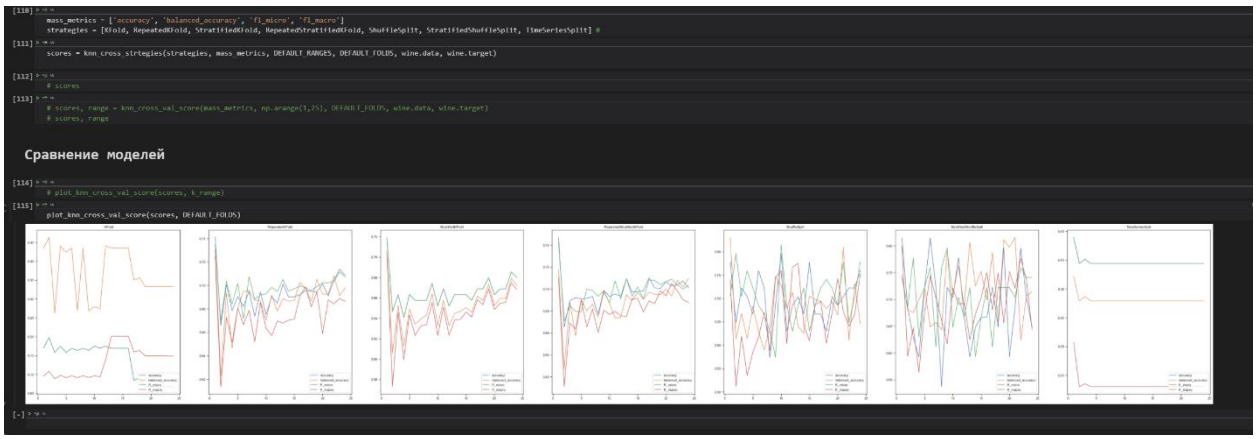
```
[103]
param_grid = {'n_neighbors' : np.arange(1, 25)}
grid = GridSearchCV(KNeighborsClassifier(), param_grid, cv=5)
grid.fit(wine_X_train, wine_y_train)
print_gridResults(grid)
```

```
Подобранный параметр: {'n_neighbors': 17}
Оценка при подобранном параметре: 0.7405228758169934

[{'n_neighbors': 17}, 0.7405228758169934]
```

Произведите подбор гиперпараметра K с использованием RandomizedSearchCV

```
[104]
param_grid = {'n_neighbors' : np.arange(1, 25)}
grid2 = RandomizedSearchCV(KNeighborsClassifier(), param_grid, cv=5, n_iter=2)
grid2.fit(wine_X_train, wine_y_train)
print_gridResults(grid2)
```

```
Подобранный параметр: {'n_neighbors': 15}
Оценка при подобранном параметре: 0.7189542483660132

[{'n_neighbors': 15}, 0.7189542483660132]
```



5 Вывод

В данной лабораторной работе я научился разделять данные на обучающую и тестовую. После разделения построил модель на основе алгоритма ближайших соседей. Познакомился с кросс-валидацией и подбором гиперпараметров.