Лабораторная работа 4 по ТМО

Водка Игорь, ИУ5-61

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

Выберите набор данных (датасет) для решения задачи классификации или регресии.

Возьмём из прошлой лабы. `../lab3/winemag-data-130k-v2.csv`

В случае необходимости проведите удаление или заполнение пропусков и кодирование категориальных признаков.

In [355]:
```
# read the data
import pandas as pd
reviews = pd.read_csv("../lab3/winemag-data-130k-v2.csv", index_col=0)
pd.set_option('max_rows', 5)

for column in ["country", "region_1", "region_2"]:
    reviews[column] = reviews[column].fillna("Unknown")

reviews['price'] = reviews.groupby('country').transform(lambda x: x.fillna(x.mean()))
```

In [356]:
```
from sklearn.preprocessing import LabelEncoder, OneHotEncoder

# country ok, winery ok
for feature in ['country', 'province', 'region_1', 'region_2', 'variety', 'winery']:
    le = LabelEncoder()
    reviews[feature] = reviews[feature].dropna()
    processed = pd.DataFrame({'result': reviews[feature]})
    reviews[feature] = le.fit_transform(processed['result'].astype(str))
```

In [357]:
```
reviews
```

Out[357]:

	country	description	designation	points	price	province	region_1	region_2	taster_name	taster_twitter_handle
0	22	Aromas include tropical fruit, broom, brimston...	Vulkà Bianco	87	87.0	331	424	15	Kerin O'Keefe	@kerinokeefe
1	31	This is ripe and fruity, a wine that is smooth...	Avidagos	87	87.0	108	1094	15	Roger Voss	@vossroger
...	...	...	...	...	...	...	...	...	...	...
129969	15	A dry style of Pinot Gris, this is crisp with ...	NaN	90	90.0	11	21	15	Roger Voss	@vossroger
129970	15	Big, rich and off-dry, this is powered by inte...	Lieu-dit Harth Cuvée Caroline	90	90.0	11	21	15	Roger Voss	@vossroger

129971 rows × 13 columns

С использованием метода train_test_split разделите выборку на обучающую и тестовую.

In [358]:
```
from sklearn.model_selection import train_test_split

X = reviews[['country', 'price', 'province', 'region_1', 'variety', 'winery']]
y = reviews['points']
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.1, random_state=42
)
```

Обучите модель ближайших соседей для произвольно заданного гиперпараметра K. Оцените качество модели с помощью трех подходящих для задачи метрик.

In [359]:
```
from sklearn.neighbors import KNeighborsRegressor
neigh = KNeighborsRegressor(n_neighbors=5)
neigh.fit(X_train, y_train)
```

Out[359]:
```
KNeighborsRegressor(algorithm='auto', leaf_size=30, metric='minkowski',
          metric_params=None, n_jobs=None, n_neighbors=5, p=2,
          weights='uniform')
```

In [360]:
```
y_pred = neigh.predict(X_test)
```

In [361]:
```
y_pred
```

Out[361]:
```
array([88. , 87. , 86.6, ..., 89.4, 87.2, 89.6])
```

In [362]:
```
y_test.values
```

Out[362]:
```
array([83, 85, 86, ..., 89, 91, 91])
```

In [363]:
```
# пусть первая метрика - максимальный модуль разности
metric1 = max(abs(y_test.values[k] - v) for k, v in enumerate(y_pred))
print("Метрика №1:", metric1)

# пусть вторая метрика - самопальная mean_squared_error
metric2 = sum(pow(y_test.values[k] - v, 2) for k, v in enumerate(y_pred)) / len(y_pred)
print("Метрика №2:", metric2)
```

```
Метрика №1: 10.0
Метрика №2: 4.792521926450363
```

Постройте модель и оцените качество модели с использованием кросс-валидации. Проведите эксперименты с тремя различными стратегиями кросс-валидации.

In [ ]:
```
```

In [364]:
```
from sklearn.model_selection import cross_val_score
estimator = neigh
scores = cross_val_score(estimator, X_train, y_train, cv=10) # 10 folds by Stratified)KFold
scores
```

Out[364]:
```
array([0.46893334, 0.46132663, 0.4546627 , 0.45731714, 0.45293478,
       0.47408317, 0.4616699 , 0.43496419, 0.443433  , 0.45459557])
```

In [365]:
```
print("Точность в первом случае: %0.2f (+/- %0.2f)" % (scores.mean(), scores.std() * 2))
```

```
Точность в первом случае: 0.46 (+/- 0.02)
```

In [366]:
```
from sklearn.model_selection import ShuffleSplit
cv = ShuffleSplit(n_splits=10, test_size=0.15, random_state=3)
scores2 = cross_val_score(estimator, X_train, y_train, cv=cv)
scores2
```

Out[366]:
```
array([0.45391062, 0.44961495, 0.45048036, 0.44908869, 0.44108992,
       0.42965588, 0.44428762, 0.44557639, 0.43150089, 0.45315958])
```

In [367]:
```
print("Точность во втором случае: %0.2f (+/- %0.2f)" % (scores2.mean(), scores2.std() * 2))
```

```
Точность во втором случае: 0.44 (+/- 0.02)
```

Произведите подбор гиперпараметра K с использованием GridSearchCV и кросс-валидации.

In [368]:
```
from sklearn.model_selection import GridSearchCV
```

In [369]:
```
grid_search = GridSearchCV(estimator, cv=5, param_grid={'n_neighbors': [1, 2, 3]})
grid_search.fit(X_train, y_train)
y_pred2 = grid_search.predict(X_test)
```

In [370]:
```
# пусть первая метрика - максимальный модуль разности
metric1 = max(abs(y_test.values[k] - v) for k, v in enumerate(y_pred2))
print("Метрика №1 для найденного параметра:", metric1)

# пусть вторая метрика - самопальная mean_squared_error
metric2 = sum(pow(y_test.values[k] - v, 2) for k, v in enumerate(y_pred2)) / len(y_pred2)
print("Метрика №2 для найденного параметра:", metric2)
```

```
Метрика №1 для найденного параметра: 12.0
Метрика №2 для найденного параметра: 4.034966917987383
```

In [ ]:
```
```