Практическое задание

Источник данных: https://www.kaggle.com/c/house-prices-advanced-regression-techniques

Набор данных house_price.csv содержит характеристики домов и цену продажи ($).

Задача: построить модель, которая по числовым характеристикам жилья предскажет его цену.

Для построения модели используйте LassoCV (https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LassoCV.html).

В LassoCV параметр alpha (λ) подбирается автоматически с помощью перекрестной проверки (кросс-валидации). Список значений для alpha можно задать явно используя параметр **alphas**. С помощью параметра **cv** устанавливают количество блоков для *k*-блочной перекрестной проверки. Атрибут **alpha_** возвращает значение alpha, выбранное с помощью перекрестной проверки, **coef_** возвращает коэффициенты модели, а **intercept_** возвращает сдвиг. Метод **score** возвращает коэффициент детерминации. Другие параметры, атрибуты и методы см. в документации).

Замечание. Параметры в классах Sklearn – это гиперпараметры, т.е. параметры, которые мы задаем при создании экземпляра класса!

1. Загрузите данные в DataFrame, используя функцию `read_csv`.

2. Сколько строк и столбцов в данных? Есть ли пропуски? Для ответа на вопросы используйте метод `info()`.

3. Для выполнения задания в наборе данных необходимо оставить только числовые признаки. Для этого можно использовать следующий программный код (а можно придумать свой):

```
numeric_dtypes = ['int64', 'float64']
numerics = []
for i in df.columns:
    if df[i].dtype in numeric_dtypes:
        numerics.append(i)
df = df[numerics]
```

```
numeric_dtypes = ['int64', 'float64']
numerics = []
for i in df.columns:
    if df[i].dtype in numeric_dtypes:
        numerics.append(i)
df = df[numerics]
```

4. Удалите столбец Id и пропущенные значения.

5. Разделите набор данных на входные данные **X** (все столбцы кроме SalePrice) и ответы **y** (столбец SalePrice).

6. Разделите данные на обучающую и тестовую выборки.

7. Обучите модель LassoCV (установите значение гиперпараметра **cv** самостоятельно). Оцените качество полученной модели. Посмотрите на коэффициенты модели. Есть ли коэффициенты равные 0? Что это означает? Попробуйте их убрать и построить модель заново. Как изменилось качество полученной модели?
from sklearn.linear_model import LassoCV
lasso = LassoCV(cv=?)

Сделайте выводы.

8. Попробуйте использовать L2-регуляризацию, т.е. обучите модель RidgeCV. Сравните полученный результат LassoCV и RidgeCV.

1. Загрузите данные в DataFrame, используя функцию `read_csv`.

```
df = pd.read_csv('house_price.csv')
```

2. Сколько строк и столбцов в данных? Есть ли пропуски? Используйте метод `info()`.

```
df.info()
```

3. Для выполнения задания в наборе данных необходимо оставить только числовые признаки. Для этого можно использовать следующий программный код (а можно придумать свой):

```
numeric_dtypes = ['int64', 'float64']
numerics = []
for i in df.columns:
    if df[i].dtype in numeric_dtypes:
        numerics.append(i)
df = df[numerics]
```

```
numeric_dtypes = ['int64', 'float64']
numerics = []
for i in df.columns:
    if df[i].dtype in numeric_dtypes:
        numerics.append(i)
df = df[numerics]
```

4. Удалите столбец Id и пропущенные значения.

```
del df['Id']
```

```
df = df.dropna()
```

5. Разделите набор данных на входные данные **X** (все столбцы кроме SalePrice) и ответы **y** (столбец SalePrice).

```
X = df.drop('SalePrice', axis = 1)
y = df['SalePrice']
```

6. Разделите данные на обучающую и тестовую выборки.

```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=42)
```

```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=42)
```

7. Обучите модель LassoCV (установите значение гиперпараметра **cv** самостоятельно). Оцените качество полученной модели. Посмотрите на

коэффициенты модели. Есть ли коэффициенты равные 0? Что это означает? Сделайте выводы.

```
from sklearn.linear_model import LassoCV
lasso = LassoCV(cv=5)  # Для примера возьмем cv = 5
```

Это означает, что для подбора alpha будет использоваться 5-блочная перекрестная проверка.

```
lasso.fit(X_train, y_train)  # Обучаем
lasso.score(X_test, y_test)  # Оцениваем качество

lasso.coef_ # Значения коэффициентов
```