Home Page - Select or create a | lesson_8 - Jupyter Notebook | Обучение | GeekBrains - образ | ПЗ. Видеоурок ""Обучение без | Почта Mail.ru

localhost:8890/notebooks/lesson_8.ipynb#

Приостановлена

jupyter lesson_8 Last Checkpoint: 32 минуты назад (autosaved)

Logout

File Edit View Insert Cell Kernel Widgets Help

Trusted | Python 3

Run Code

Задание 1 Импортируйте библиотеки pandas, numpy и matplotlib. Загрузите "Boston House Prices dataset" из встроенных наборов данных библиотеки sklearn. Создайте датафреймы X и y из этих данных. Разбейте эти датафреймы на тренировочные (X_train, y_train) и тестовые (X_test, y_test) с помощью функции train_test_split так, чтобы размер тестовой выборки составлял 20% от всех данных, при этом аргумент random_state должен быть равен 42. Масштабируйте данные с помощью StandardScaler. Постройте модель TSNE на тренировочный данных с параметрами: n_components=2, learning_rate=250, random_state=42. Постройте диаграмму рассеяния на этих данных.

In [1]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
```

Загрузим "Boston House Prices dataset" из встроенных наборов данных библиотеки sklearn. Создадим датафреймы X и y из этих данных.

In [2]:
```
from sklearn.datasets import load_boston

boston = load_boston()
X = pd.DataFrame(boston['data'], columns=boston['feature_names'])
y = pd.DataFrame(boston['target'], columns=["price"])
```

Разобьем эти датафреймы на тренировочные (X_train, y_train) и тестовые (X_test, y_test) с помощью функции train_test_split, с аргументами test_size=0.2, random_state=42.

In [3]:
```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

Масштабируем данные с помощью StandardScaler.

In [5]:
```
from sklearn.preprocessing import StandardScaler
scaler = StandardScaler()

X_train_scaled = pd.DataFrame(
    scaler.fit_transform(X_train),
    columns=X_train.keys())
X_test_scaled = pd.DataFrame(
    scaler.fit_transform(X_test),
    columns=X_test.keys())
```

In [7]:
```
X_train_scaled.head()
```

Out[7]:

	CRIM	ZN	INDUS	CHAS	NOX	RM	AGE	DIS	RAD	TAX	PTRATIO	B	LSTAT

lesson_8.ipynb

Показать все

EN 14:54 15.02.2020

Home Page - Select or create a | lesson_8 - Jupyter Notebook | Обучение | GeekBrains - обра | ПЗ. Видеоурок ""Обучение бе | Почта Mail.ru

localhost:8890/notebooks/lesson_8.ipynb#

Приостановлена

jupyter lesson_8 Last Checkpoint: 33 минуты назад (autosaved)

Logout

File Edit View Insert Cell Kernel Widgets Help

Trusted | Python 3

Run Code

In [7]:
```
X_train_scaled.head()
```

Out[7]:

	CRIM	ZN	INDUS	CHAS	NOX	RM	AGE	DIS	RAD	TAX	PTRATIO	B	LSTAT
0	1.287702	-0.500320	1.033237	-0.278089	0.489252	-1.428069	1.028015	-0.802173	1.706891	1.578434	0.845343	-0.074337	1.753505
1	-0.336384	-0.500320	-0.413160	-0.278089	-0.157233	-0.660087	-0.431199	0.324349	-0.624360	-0.584648	1.204741	0.430184	-0.561474
2	-0.403253	1.013271	-0.715218	-0.278089	-1.008723	-0.402063	-1.618599	1.330697	-0.974048	-0.602724	-0.637176	0.065297	-0.651595
3	0.388230	-0.500320	1.033237	-0.278089	0.489252	-0.300450	0.591681	-0.839240	1.706891	1.578434	0.845343	-3.868193	1.525387
4	-0.325282	-0.500320	-0.413160	-0.278089	-0.157233	-0.831094	0.033747	-0.005494	-0.624360	-0.584648	1.204741	0.379119	-0.165787

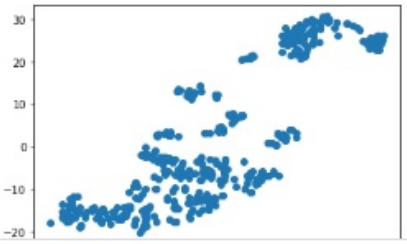
Построим модель TSNE на тренировочный данных с параметрами: n_components=2, learning_rate=250, random_state=42.

In [8]:
```
from sklearn.manifold import TSNE
tsne = TSNE(n_components=2, learning_rate=250, random_state=42)

X_train_tsne = tsne.fit_transform(X_train_scaled)

print('Количество признаков до:\t{}'.format(X_train_scaled.shape[1]))
print('Количество признаков после:\t{}'.format(X_train_tsne.shape[1]))
```

```
Количество признаков до:        13
Количество признаков после:     2
```

Количество признаков до: 13 Количество признаков после: 2 Построим диаграмму рассеяния на этих данных.

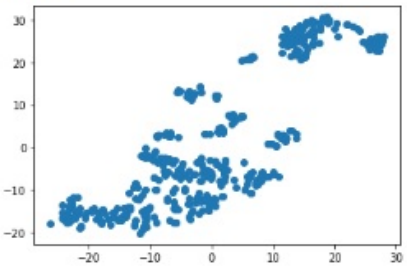
In [9]:
```
plt.scatter(X_train_tsne[:, 0], X_train_tsne[:, 1])

plt.show()
```



lesson_8.ipynb

Показать все

EN 14:54 15.02.2020



Задание 2 С помощью KMeans разбейте данные из тренировочного набора на 3 кластера, используйте все признаки из датафрейма X_train. Параметр max_iter должен быть равен 100, random_state сделайте равным 42. Постройте еще раз диаграмму рассеяния на данных, полученных с помощью TSNE, и раскрасьте точки из разных кластеров разными цветами. Вычислите средние значения price и CRIM в разных кластерах.

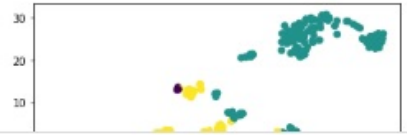
Решение:

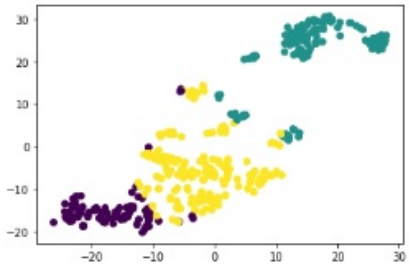
С помощью KMeans разоббем данные из тренировочного набора на 3 кластера, используя все признаки из датафрейма X_train. В качестве параметров возьмем n_clusters=3, random_state=42, max_iter=100

In [10]:
```
from sklearn.cluster import KMeans

model = KMeans(n_clusters=3, random_state=42, max_iter=100)
```

Построим диаграмму рассеяния на данных, полученных с помощью TSNE, и раскрасим точки из разных кластеров разными цветами.

In [11]:
```
labels_train = model.fit_predict(X_train_scaled)

plt.scatter(X_train_tsne[:, 0], X_train_tsne[:, 1], c=labels_train)

plt.show()
```





Вычислим средние значения price и CRIM в разных кластерах.

In [12]:
```
for i in range(3):
    price_mean = y_train.loc[labels_train == i, 'price'].mean()
    CRIM_mean = X_train.loc[labels_train == i, 'CRIM'].mean()
    print(f'Кластер {i}: \n\
    Среднее значение price: {price_mean:.3f}\n\
    Среднее значение CRIM: {CRIM_mean:.3f}\n')
```

```
Кластер 0:
    Среднее значение price: 27.788
    Среднее значение CRIM: 0.074

Кластер 1:
    Среднее значение price: 16.165
    Среднее значение CRIM: 10.797

Кластер 2:
    Среднее значение price: 24.958
    Среднее значение CRIM: 0.422
```

Поделив данные на три кластера, можно заметить что чем дороже цены на квартиры из данного кластера, тем меньше криминала в районах квартир данного кластера

Задание 3 Применить модель KMeans, построенную в предыдущем задании, к данным из тестового набора. Вычислить средние значения price и CRIM в разных кластерах на тестовых данных.

Home Page - Select or create a | lesson_8 - Jupyter Notebook | Обучение | GeekBrains - образ | ПЗ. Видеоурок ""Обучение без | Почта Mail.ru

localhost:8890/notebooks/lesson_8.ipynb#

Приостановлена

jupyter lesson_8 Last Checkpoint: 34 минуты назад (autosaved)

Logout

File Edit View Insert Cell Kernel Widgets Help

Trusted | Python 3

Run Code

```
Кластер 0:
    Среднее значение price: 27.788
    Среднее значение CRIM: 0.074

Кластер 1:
    Среднее значение price: 16.165
    Среднее значение CRIM: 10.797

Кластер 2:
    Среднее значение price: 24.958
    Среднее значение CRIM: 0.422
```

Поделив данные на три кластера, можно заметить что чем дороже цены на квартиры из данного кластера, тем меньше криминала в районах квартир данного кластера

Задание 3 Применить модель KMeans, построенную в предыдущем задании, к данным из тестового набора. Вычислить средние значения price и CRIM в разных кластерах на тестовых данных.

In [14]:
```
labels_test = model.predict(X_test_scaled)

for i in range(3):
    price_mean = y_test.loc[labels_test == i, 'price'].mean()
    CRIM_mean = X_test.loc[labels_test == i, 'CRIM'].mean()
    print(f'Кластер {i}: \n\
    Среднее значение price: {price_mean:.3f}\n\
    Среднее значение CRIM: {CRIM_mean:.3f}\n')
```

```
Кластер 0:
    Среднее значение price: 28.414
    Среднее значение CRIM: 0.080

Кластер 1:
    Среднее значение price: 16.437
    Среднее значение CRIM: 10.166

Кластер 2:
    Среднее значение price: 22.031
    Среднее значение CRIM: 0.285
```

In [ ]:

lesson_8.ipynb

Показать все

EN 14:56 15.02.2020