Практика по теме 1.4. Методы понижения размерности

Результат выполнения каждого пункта должен быть выведен на экран. *В коде должны присутствовать комментарии и выводы*.

Пункты задания:

1. Примените метод главных компонент для выделения новых признаков для данных:
– Ознакомьтесь с примером применения метода главных компонент для выделения новых признаков для данных Abalone.data (файл *PCA_пример.ipynb*).
– Примените метод главных компонент для выделения новых признаков для данных diamonds.csv:

- преобразуйте категориальные признаки с помощью кодировщика *OneHotEncoder* (пример использования см. ниже), оставить в таблице исходные числовые признаки и новые бинаризованные:
- стандартизуйте числовые признаки;
- постройте график для 2-х компонент, найденных с помощью метода PCA (для меток используйте переменную *cut*, пример см. ниже);
- задайте число компонент, равное числу признаков в преобразованной таблице, и определите вклад каждой компоненты, выведите таблицу результатов на экран;
- определите оптимальное число компонент.

2. Примените метод UMAP для выделения новых признаков для данных:
– Ознакомьтесь с примером применения метода UMAP для выделения новых признаков для данных Abalone.data (файл *UMAP_пример.ipynb*).
– Примените метод UMAP для выделения новых признаков для данных *diamonds.csv*:

- преобразуйте категориальные признаки с помощью кодировщика *OneHotEncoder* (или другого), стандартизировать исходные числовые признаки, оставить в таблице числовые признаки и новые бинаризованные (нечисловые, преобразованные в числовые);
- примените метод UMAP для 2-х компонент;
- подоберите значения параметров метода UMAP (*n_neighbors*, *min_dist*, *metric*), при которых классы визуально наиболее различимы.

Пример кода бинаризации (кодировщик OneHotEncoder):

```
from sklearn.preprocessing import OneHotEncoder


enc = OneHotEncoder(handle_unknown='ignore')
enc.fit(diamonds_data[['cut', 'color', 'clarity']])
enc.categories_
data_cat_tr = enc.transform(diamonds_data[['cut', 'color',
'clarity']]).toarray()


X_number = diamonds_data.drop(['cut', 'color', 'clarity'], axis=1)
X_bin = data_cat_tr


Z = []
Z = np.hstack((X_number, X_bin))
```

Пример категоризации нечисловой переменную cut для дальнейшего использования в качестве меток при визуализации:

```
#Выделяем категориальный признак cut, который будет отображаться на графике в качестве меток
#Перед этим переводим нечисловые идентификаторы признака cut в числовые
dct = {'Fair': 0,'Good': 1,'Very Good':2, 'Premium':3, 'Ideal':4}
diamonds['cut'] = diamonds['cut'].map(dct)
y = diamonds['cut']
#Задаем числовые метки категориального признака cut
target_names = ['Fair','Good','Very Good', 'Premium', 'Ideal']
```