Курс «Базовая обработка данных на языке Python»

автор: Киреев В.С., к.т.н., доцент

Лабораторная работа № 8

Тема: «Использование объектов и методов библиотек matplotlib, sklearn и pyod для проведения разведочного анализа данных в Python»

Цель работы: изучить основы разведочного анализа данных (exploratory data analysis, EDA) с помощью библиотек matplotlib, sklearn и pyod.

Теоретическая справка

Разведочный анализ данных (Exploratory Data Analysis, EDA) - начальный этап в проектах по изучению данных. Он включает в себя анализ и визуализацию данных для понимания их ключевых характеристик, выявления закономерностей и определения взаимосвязей между переменными. Это метод изучения и исследования наборов данных для понимания их преобладающих характеристик, выявления закономерностей, обнаружения выбросов и определения взаимосвязей между переменными. EDA обычно проводится в качестве предварительного шага перед проведением дополнительных формальных статистических анализов или моделирования.

Ключевые аспекты EDA включают в том числе:

- Распределение данных: Изучение распределения точек данных для понимания их диапазона, центральных тенденций (среднее значение, медиана) и дисперсии (дисперсия, стандартное отклонение).
- Графические представления: Использование таких графиков, как гистограммы, квадратные диаграммы, диаграммы рассеяния и гистограммы, для визуализации взаимосвязей в данных и распределений переменных.
- Обработка пропущенных значений: Обнаружение и принятие решения о том, как поступить с пропущенными точками данных - путем вменения или удаления, в зависимости от их влияния и объема пропущенных данных.
- Сводная статистика: Вычисление ключевых статистических данных, которые позволяют понять тенденции и нюансы данных.
- Проверка предположений: Многие статистические тесты и модели предполагают, что данные отвечают определенным условиям (например, нормальность или гомоскедастичность). EDA помогает проверить эти предположения.
- Обнаружение аномалий, также называемое обнаружением выбросов: Процесс поиска закономерностей в любом наборе данных, которые значительно отклоняются от ожидаемого или «нормального поведения». Выбросы могут повлиять на статистический анализ и могут указывать на ошибки ввода данных или уникальные случаи.

Scikit-learn - это библиотека Python с открытым исходным кодом, которая реализует ряд алгоритмов машинного обучения, предварительной обработки, кросс-валидации и визуализации с помощью единого интерфейса. Это библиотека машинного обучения с открытым исходным кодом, которая предоставляет множество инструментов для решения различных задач машинного обучения, таких как классификация, регрессия, кластеризация и многие другие.

```
from sklearn.ensemble import IsolationForest
model2=IsolationForest()
model2.fit_predict(data.iloc[:,1:-1])
```

Matplotlib —библиотека визуализации на Python. Она построена на массивах NumPy и предназначена для работы с более широким стеком SciPy и состоит из нескольких графиков, таких как линейный, столбчатый, точечный, гистограмма и т. д. Основная цель Matplotlib — предоставить пользователям инструменты и функциональные возможности для графического представления данных, что упрощает их анализ и понимание. Первоначально она была разработана Джоном Д. Хантером в 2003 году.

Пример построения стеблевого графика. График стеблей, также известный как график стеблей и листьев, - это тип графика, используемый для отображения данных вдоль числовой линии. Стеблевые графики особенно полезны для визуализации дискретных наборов данных, где значения представлены в виде «стеблей», простирающихся от базовой линии, а точки данных обозначены как «листья» вдоль стеблей.

```
import matplotlib.pyplot as plt
import numpy as np
x = np.linspace(0.1, 2 * np.pi, 41)
y = np.exp(np.sin(x))
plt.stem(x, y, use_line_collection = True)
plt.show()
```

PyOD – это Python-библиотека с более чем 30 современными алгоритмами обнаружения редких и подозрительных данных или событий. PyOD включает широкий класс алгоритмов обнаружения аномалий, начиная с классических алгоритмов, таких как Isolation Forest, и заканчивая новейшими методами глубокого обучения и новыми алгоритмами (например, COPOD)

```
from pyod.models.abod import ABOD
clf = ABOD(contamination=outliers_fraction)
clf.fit(X)
clf.predict(X)
```

Анализ главных компонент (Principal Components Analysis) - это неконтролируемая задача предварительной обработки, которая выполняется перед применением любого алгоритма ML. PCA основан на «ортогональном линейном преобразовании», которое представляет собой математическую технику проецирования атрибутов набора данных на новую систему координат. Атрибут, который описывает наибольшую дисперсию, называется первым главным компонентом и помещается в первую координату. PCA, преобразует данные из пространства высокой размерности в пространство низкой размерности, выбирая наиболее важные атрибуты, которые отражают максимум информации о наборе данных.

```
from sklearn.decomposition import PCA
principal=PCA(n_components=3)
principal.fit(Scaled_data)
x=principal.transform(Scaled_data)
```

Самостоятельное задание

1. Провести разведочный анализ на загруженных данных. Здесь и далее задания выполняются на массиве транзакций по кредитным картам.
   1.1. Определить число строк и столбцов и их типы данных
   1.2. Визуализировать распределения данных по каждому столбцу (матрица распределений. Для каждого столбца выбрать свою визуализацию)
   1.3. Визуализировать данные в двухмерном пространстве, проведя предварительно анализ главных компонент
2. Провести поиск выбросов (аномалий) на загруженных данных
   2.1. Использовать Isolation Forest
   2.2. Использовать DBSCAN
   2.3. Использовать LOF
   2.4. Сравнить результаты
3. Использовать для поиска аномалий предварительную факторизацию с помощью PCA и методами из п.2.
4. Интерпретировать аномалии выделенные из п.2-п.3. Определить какие аномалии совпадают со фродовыми транзакциями.
5. Выделить решающие правила, позволяющие выделить фрод на датасете