ДИСЦИПЛИНА **Технологии и инструментарий анализа больших данных**

(полное наименование дисциплины без сокращений)

ИНСТИТУТ **информационных технологий**

КАФЕДРА **прикладной математики**

(полное наименование кафедры)

ВИД УЧЕБНОГО МАТЕРИАЛА **Материалы для практических занятий**

(в соответствии с пп.1-11)

ПРЕПОДАВАТЕЛЬ **Новикова Ольга Александровна**

(фамилия, имя, отчество)

СЕМЕСТР **7, 2023-2024**

(указать семестр обучения, учебный год)

План практических занятий

1. Язык программирования Python
2. Циклы, функции Python и работа с библиотеками
3. Визуализация в языке программирования Python
4. Статистика на Python
5. Визуализация многомерных данных с помощью t-SNE
6. Предобработка данных и статистические тесты
7. Корреляция и линейная регрессия
8. Дисперсионный анализ
9. Классификация
10. Задача кластеризации
11. Ансамблевое обучение
12. Ассоциативные правила
13. Генетические алгоритмы

Практическая работа №1

1. Установить Python.

2. Написать программу, которая вычисляет площадь фигуры, параметры которой подаются на вход. Фигуры, которые подаются на вход: треугольник, прямоугольник, круг. Результатом работы является словарь, где ключ – это название фигуры, а значение – это площадь.

3. Написать программу, которая на вход получает два числа и операцию, которую к ним нужно применить. Должны быть реализованы следующие операции: +, -, /, //, abs – модуль, pow или ** – возведение в степень. Результатом работы программы является одно число.

4. Написать программу, вычисляющую площадь треугольника по переданным длинам трёх его сторон по формуле Герона:

$$\sqrt{p(p-a)(p-b)(p-c)},$$

где $p = \frac{a+b+c}{2}$.

На вход программе подаются целые числа, выводом программы должно являться вещественное число, соответствующее площади треугольника.

5. Оформить отчет о проделанной работе. В отчете должен быть приведен код и результат его выполнения.

Практическая работа №2

1. Напишите программу, которая считывает с консоли числа (по одному в строке) до тех пор, пока сумма введённых чисел не будет равна 0 и после этого выводит сумму квадратов всех считанных чисел.
2. Напишите программу, которая выводит последовательность чисел, длинною N, где каждое число повторяется столько раз, чему оно равно. На вход программе передаётся неотрицательное целое число N. Например, если N = 7, то программа должна вывести 1 2 2 3 3 3 4. Вывод элементов списка через пробел – print(*list).
3. Матрицу произвольного размера вытянуть в один вектор, не применяя встроенные методы Python. Пример:

```
Исходная матрица
 [[1 1 1]
 [1 1 1]
 [1 1 1]]
Результат  [1, 1, 1, 1, 1, 1, 1, 1, 1]
```

   Для создания матрицы можно использовать np.random.rand(кол-во строк, кол-во столбцов)
4. Скачать и загрузить данные **insurance.csv**, используя библиотеку pandas.
5. Использовать метод info().
6. Узнать количество пропущенных значений с помощью метода isna() и sum().
7. Провести интерполяцию данных, если это необходимо (метод interpolate()).
8. Используя метод apply(), добавить новый столбец в датайфрейм, в котором хранятся значения признака, является ли человек многодетным (если больше или равно 3 – True, иначе False).
9. Составить отчет о проделанной работе. В отчете должен быть представлен код и результаты его выполнения с выводами.

ПРАКТИЧЕСКАЯ РАБОТА №3

1. Найти и выгрузить многомерные данные с использованием библиотеки pandas. В отчёте описать найденные данные.
2. Вывести информацию о данных при помощи методов **.info()**, .**head()**. Проверить данные на наличие пустых значений. В случае их наличия удалить данные строки или интерполировать пропущенные значения. При необходимости дополнительно предобработать данные для дальнейшей работы с ними.
3. Построить столбчатую диаграмму (**.bar**) с использованием модуля **graph_objs** из библиотеки **Plotly** со следующими параметрами:

3.1. По оси X указать дату или название, по оси У указать количественный показатель.

3.2. Сделать так, чтобы столбец принимал цвет в зависимости от значения показателя (marker=dict(color=признак, coloraxis="coloraxis")).

3.3. Сделать так, чтобы границы каждого столбца были выделены чёрной линией с толщиной равной 2.

3.4. Отобразить заголовок диаграммы, разместив его по центру сверху, с 20 размером текста.

3.5. Добавить подписи для осей X и Y с размером текста, равным 16. Для оси абсцисс развернуть метки так, чтобы они читались под углом, равным 315.

3.6. Размер текста меток осей сделать равным 14.

3.7. Расположить график во всю ширину рабочей области и присвоить высоту, равную 700 пикселей.

3.8. Убрать лишние отступы по краям.

4. Построить круговую диаграмму (**go.Pie**), использовав данные и стиль оформления из предыдущего графика. Сделать так, чтобы границы каждой доли были выделены чёрной линией с толщиной, равной 2.
5. Построить линейный график накопленных значений количественного показателя.

5.1. Сделать график с линиями и маркерами, цвет линии 'crimson', цвет точек 'white', цвет границ точек 'black', толщина границ точек равна 2.

5.2. Добавить сетку на график, сделать её цвет 'ivory' и толщину равную 2. (Можно сделать это при настройке осей с помощью gridwidth=2, gridcolor='ivory').

6. Постараться создать аналогичные графики с использованием библиотеки matplotlib.

7. На основе проделанной работы составить отчёт с описанием и скриншотами полученных результатов, сделать выводы о выбранном организации (процессе) на основе полученных графиков, сравнить библиотеки.

Практическая работа №4

1. Загрузить данные из файла “insurance.csv”.
2. С помощью метода describe() посмотреть статистику по данным. Сделать выводы.
3. Построить гистограммы для числовых показателей. Сделать выводы.
4. Найти меры центральной тенденции и меры разброса для индекса массы тела (bmi) и расходов (charges). Отобразить результаты в виде текста и на гистограммах (3 вертикальные линии). Добавить легенду на графики. Сделать выводы.
5. Построить box-plot для числовых показателей. Названия графиков должны соответствовать названиям признаков. Сделать выводы.
6. Используя признак charges или imb, проверить, выполняется ли центральная предельная теорема. Использовать различные длины выборок n. Количество выборок = 300. Вывести результат в виде гистограмм. Найти стандартное отклонение и среднее для полученных распределений. Сделать выводы.
7. Построить 95% и 99% доверительный интервал для среднего значения расходов и среднего значения индекса массы тела.
8. Проверить распределения следующих признаков на нормальность: индекс массы тела, расходы. Сформулировать нулевую и альтернативную гипотезы. Для каждого признака использовать KS-тест и q-q plot. Сделать выводы на основе полученных p-значений.
9. Оформить отчет на основе проделанной работы. Написать выводы.

Практическая работа №5

Выполнить визуализацию многомерных данных, используя t-SNE. Необходимо использовать набор данных MNIST или fashion MNIST (можно использовать и другие готовые наборы данных, где можно наблюдать разделение объектов по кластерам). Рассмотреть результаты визуализации для разных значений перплексии.

Выполнить визуализацию многомерных данных, используя UMAP с различными параметрами n_neighbors и min_dist. Рассчитать время работы алгоритма с помощью библиотеки time и сравнить его с временем работы t-SNE.

Оформить отчет о проделанной работе. Отчет должен содержать результаты визуализации для разных значений параметров и выводы.

Практическая работа №6

1. Загрузить данные из файла “ECDCCases.csv”.
2. Проверить в данных наличие пропущенных значений. Вывести количество пропущенных значений в процентах. Удалить два признака, в которых больше всех пропущенных значений. Для оставшихся признаков обработать пропуски: для категориального признака использовать заполнение значением по умолчанию (например, «other»), для числового признака использовать заполнение медианным значением. Показать, что пропусков больше в данных нет.
3. Посмотреть статистику по данным, используя describe(). Сделать выводы о том, какие признаки содержат выбросы. Посмотреть, для каких стран количество смертей в день превысило 3000 и сколько таких дней было.
4. Найти дублирование данных. Удалить дубликаты.
5. Загрузить данные из файла “ bmi.csv”. Взять оттуда две выборки. Одна выборка – это индекс массы тела людей с региона northwest, вторая выборка – это индекс массы тела людей с региона southwest. Сравнить средние значения этих выборок, используя t-критерий Стьюдента. Предварительно проверить выборки на нормальность (критерий Шопиро-Уилка) и на гомогенность дисперсии (критерий Бартлетта).
6. Кубик бросили 600 раз, получили следующие результаты:

N	**Количество выпадений**
1	97
2	98
3	109
4	95
5	97
6	104

С помощью критерия Хи-квадрат проверить, является ли полученное распределение равномерным. Использовать функцию scipy.stats.chisquare().

7. С помощью критерия Хи-квадрат проверить, являются ли переменные зависимыми.

Создать датафрейм, используя следующий код:

```
data = pd.DataFrame({'Женат': [89,17,11,43,22,1],
                     'Гражданский брак': [80,22,20,35,6,4],
                     'Не состоит в отношениях': [35,44,35,6,8,22]})
data.index = ['Полный рабочий день','Частичная занятость','Временно не работает','На домохозяйстве','На пенсии','Учёба']
```

Использовать функцию scipy.stats.chi2_contingency().

Влияет ли семейное положение на занятость?

8. Оформить отчет о проделанной работе, написать выводы.

Практическая работа № 7-8

1. Определить два вектора, представляющие собой число автомобилей, припаркованных в течении 5 рабочих дней у бизнес-центра на уличной стоянке и в подземном гараже.

День	Улица	Гараж
Понедельник	80	100
Вторник	98	82
Среда	75	105
Четверг	91	89
Пятница	78	102

Найти и интерпретировать корреляцию между переменными «Улица» и «Гараж» (подсчитать корреляцию по Пирсону).

2. Построить диаграмму рассеяния.
3. Загрузить bitcoin.csv.
4. Скрыть последние 14 дней.

```
projection = 14
data['predict'] = data['close'].shift(-projection)
```

5. Предсказать стоимость криптовалюты за последние 14 дней с помощью линейной регрессии.
6. Вывести угол наклона и y-перехват.
7. Построить диаграмму.
8. Загрузить housePrice.csv
9. Произвести предобработку.
10. Реализовать линейную регрессию вручную, без использования библиотеки.
11. Вывести угол наклона и y-перехват.
12. Построить диаграмму.
13. Оформить отчет о проделанной работе, написать выводы.

Практическая работа №9

1. Загрузить данные: 'insurance.csv'. Вывести и провести предобработку. Вывести список уникальных регионов.

2. Выполнить однофакторный ANOVA тест, чтобы проверить влияние региона на индекс массы тела (BMI), используя первый способ, через библиотеку Scipy.

3. Выполнить однофакторный ANOVA тест, чтобы проверить влияние региона на индекс массы тела (BMI), используя второй способ, с помощью функции anova_lm() из библиотеки statsmodels.

4. С помощью t критерия Стьюдента перебрать все пары. Определить поправку Бонферрони. Сделать выводы.

5. Выполнить пост-хок тесты Тьюки и построить график.

6. Выполнить двухфакторный ANOVA тест, чтобы проверить влияние региона и пола на индекс массы тела (BMI), используя функцию anova_lm() из библиотеки statsmodels.

7. Выполнить пост-хок тесты Тьюки и построить график.

8. Оформить отчет о проделанной работе, написать выводы.

Практическая работа №10

1. Найти данные для классификации. Данные в группе повторяться не должны! Предобработать данные, если это необходимо.
2. Изобразить гистограмму, которая показывает баланс классов. Сделать выводы.
3. Разбить выборку на тренировочную и тестовую. Тренировочная для обучения модели, тестовая для проверки ее качества.
4. Применить алгоритмы классификации: логистическая регрессия, SVM, KNN. Построить матрицу ошибок по результатам работы моделей (использовать confusion_matrix из sklearn.metrics).
5. Сравнить результаты классификации, используя accuracy, precision, recall и f1-меру (можно использовать classification_report из sklearn.metrics). Также сравнить время работы алгоритмов. Сделать выводы.
6. Оформить отчет о проделанной работе.

Практическая работа №11

1. Найти данные для кластеризации. Данные в группе не должны повторяться! Внимание, если признаки в данных имеют очень сильно разные масштабы, то необходимо данные предварительно нормализовать.
2. Провести кластеризацию данных с помощью алгоритма k-means. Использовать «правило локтя» и коэффициент силуэта для поиска оптимального количества кластеров.
3. Провести кластеризацию данных с помощью алгоритма иерархической кластеризации.
4. Провести кластеризацию данных с помощью алгоритма DBSCAN.
5. Сравнить скорость работы алгоритмов. Результаты изобразить в виде таблицы.
6. Визуализировать кластеризованные данные с помощью t-SNE или UMAP если данные многомерные. Если данные трехмерные, то можно использовать трехмерный точечный график.
7. Оформить отчет о проделанной работе. Сделать выводы.

Практическая работа №12

1) Найти данные для задачи классификации или для задачи регрессии.
2) Реализовать баггинг.
3) Реализовать бустинг на тех же данных, что использовались для баггинга.
4) Сравнить результаты работы алгоритмов (время работы и качество моделей). Сделать выводы.
5) Оформить отчет о проделанной работе.

Практическая работа №13-14

1. Загрузить данные Market_Basket_Optimisation.csv.
2. Визуализировать данные (отразить на гистограммах относительную и фактическую частоту встречаемости для 20 наиболее популярных товаров).
3. Применить алгоритм Apriori, используя 3 разные библиотеки (apriori_python, apyori, efficient_apriori).
4. Применить алгоритм FP-Growth из библиотеки fpgrowth_py.
5. Сравнить время выполнения всех алгоритмов и построить гистограмму.
6. Загрузить данные data.csv.
7. Визуализировать данные (отразить на гистограммах относительную и фактическую частоту встречаемости для 20 наиболее популярных товаров).
8. Применить алгоритм Apriori, используя 3 разные библиотеки (apriori_python, apyori, efficient_apriori).
9. Применить алгоритм FP-Growth из библиотеки fpgrowth_py.
10. Сравнить время выполнения всех алгоритмов и построить гистограмму.
11. Сформулировать выводы и сделать отчет.

Практическая работа №15-16

1. Загрузить датасет breast-cancer-wisconsin1.data;
2. Реализовать выделение признаков для классификации набора данных breast-cancer-wisconsin1.data по примеру приведенным выше с zoo.data;
3. Вывести лучшее решение и на сколько увеличилось значение верности;
4. Сформировать отчёт.