Лабораторные работы по предмету BigData

Адаптация для Python

Лаб01

**Знакомство с языком Python**

1. Установите и запустите программу Python.
2. Настройте рабочую директорию с помощью функции os.chdir().
3. Создайте вектор X с значениями от 0 до 1 и постройте гистограмму с использованием библиотеки matplotlib.
4. Сохраните результаты моделирования в текстовый файл (\*.txt, \*.csv), гистограмму в графический файл (\*.jpg, \*.svg) с использованием функции записи в файл.
5. Познакомьтесь с основными структурами данных: numeric() (числовые данные), list() (списки), dict() (словари), set() (множества).
6. Познакомьтесь со строковой переменной.
7. Установите и загрузите дополнительные пакеты с помощью функции install.packages() и import.

**Лабораторная работа 5**

**Обработка данных. Выбор экземпляров (Instance Selection)**

1. **Выполните классификацию к-ближайших соседей с использованием функции** KNeighborsClassifier **из пакета** scikit-learn **на наборе данных** iris **[1]. Проведите нормализацию данных, разделите выборку на обучающую и тестовую. Оцените построенную модель с использованием функции** cross\_val\_score **из пакета** scikit-learn**. Постройте матрицу ошибок [2] и диагональную оценку качества прогноза (diagonal mark quality prediction).**
2. **Рассмотрите пример реализации метода опорных векторов с использованием функции** SVC **из пакета** scikit-learn**. Постройте линейный классификатор для прогнозирования. Для подбора параметров модели выполните перекрестную проверку с делением исходной выборки на 10 равных частей (cross=10) [3, с.172].**
3. **Выполните расчет главных компонент с использованием пакета** numpy **и его функции** linalg.svd**. Постройте ординационную диаграмму методом PCA [3, с. 49] и сделайте выводы.**

Лабараторная 6

1. **С использованием функции discretize из пакета** pandas **и** numpy **выполните преобразование непрерывной переменной в категориальную [1] различными методами:** interval **(равная ширина интервала),** frequency **(равная частота),** cluster **(кластеризация) и** fixed **(категории задают границы интервалов). Используйте набор данных iris. Сделайте выводы.**

В Python для дискретизации можно использовать библиотеки pandas и numpy. Например, для метода interval можно использовать pd.cut, для frequency — pd.qcut, а для cluster — методы кластеризации из sklearn.

1. **С использованием пакета discretization выполните дискретизацию с использованием алгоритмов Chi2 и CAIM [2]. Используйте набор данных iris. Сравните результаты и сделайте выводы.**

Лабораторная 7

1. Установите пакет pyspark, установите Java Virtual Machine (JVM). Подключитесь к локальному Spark-кластеру. Загрузите таблицу flights из пакета nycflights13 в Spark-кластер. Выполните запросы (задание 3, Лабораторная работа 2). Сравните результаты, сделайте выводы.
   * Вместо sparklyr используйте pyspark для работы с Apache Spark в Python.
   * Для загрузки данных используйте pyspark.sql.SparkSession.
2. Настройте для использования Hadoop [2-5], подключите количество слов в файле \*.txt с использованием HDFS [3]. Файл сгенерировать самостоятельно.
   * Вместо Nadoor используйте hadoop и hdfs для работы с распределённой файловой системой.
   * Используйте библиотеку hdfs для взаимодействия с HDFS в Python.
3. Установите pymongo [6, 7]. Подключите библиотеку pymongo. Выполните пример для набора its с использованием функции pymongo.MongoClient(). Сохраните код и сделайте выводы.
   * Вместо mongolite используйте pymongo для работы с MongoDB в Python.
   * Используйте pymongo.MongoClient для подключения к MongoDB и выполнения операций.