

ЛАБОРАТОРНАЯ РАБОТА №1

МЕТОД К-БЛИЖАЙШИХ СОСЕДЕЙ. ЛОГИСТИЧЕСКАЯ РЕГРЕССИЯ

(Продолжительность лабораторного занятия – 4 часа)

А. НАЗНАЧЕНИЕ И КРАТКАЯ ХАРАКТЕРИСТИКА РАБОТЫ

В процессе выполнения настоящей работы закрепляются знания студентов по разделам «Метрические методы классификации и регрессии» и «Линейные модели классификации и регрессии» курса «Применение методов искусственного интеллекта в электроэнергетике». Работа имеет экспериментальный характер и включает анализ данных и работы алгоритмов машинного обучения.

Целью работы является получение практических навыков работы с метрическими и линейными моделями классификации и регрессии.

Б. СОДЕРЖАНИЕ РАБОТЫ

Работа содержит:

1. Анализ и предварительную предобработку данных.
2. Поиск оптимального значения k для метрического метода и C для линейного метода, используя кросс-валидацию.
3. Повторение пункта 2 после масштабирования признаков, анализ и визуализация получившихся результатов.

Работа включает:

1. Экспериментальную работу в лаборатории.
2. Составление исполнительного отчета.

Работа выполняется на компьютерах в среде разработки JupyterLab.

В. ЗАДАНИЕ НА РАБОТУ В ЛАБОРАТОРИИ

1. Загрузить анализируемые данные, выданные преподавателем. Подготовить данные для обучения.
2. Разделить выборку данных на обучающую и тестовую. Обучить модель ближайших соседей на обучающей выборке и проверить качество модели на обучающей и на тестовой выборках с изменением числа соседей от 1 до 50.
3. Создать генератор разбиений для кросс-валидации по пяти блокам.
4. Найти оптимальное значение k в диапазоне от 1 до 50 с шагом 1 для метода k -ближайших соседей.
5. Найти оптимальное значение C в диапазоне от 0,01 до 1 с шагом 0,01 для метода логистической регрессии.
6. Произвести масштабирование признаков и повторить пункты 4,5.

Г. МЕТОДИЧЕСКИЕ УКАЗАНИЯ К РАБОТЕ В ЛАБОРАТОРИИ

К пункту 1.

Для того, чтобы загрузить данные в формате «.csv» используйте метод `read_csv` библиотеки `Pandas`, аргументом которого является путь к файлу. Метод возвращает объект класса `DataFrame`. Конвертировать категориальный признак в числовой можно, воспользовавшись методом `get_dummies` библиотеки `Pandas`.

К пункту 2.

Используйте метод `train_test_split` библиотеки `Scikit-Learn` для разделения выборки на обучающую и тестовую:

```
from sklearn.model_selection import train_test_split
```

Импортируйте модель k-ближайших соседей из библиотеки `Scikit-Learn`:

```
from sklearn.neighbors import KNeighborsClassifier
```

Установить количество ближайших соседей можно, используя аргумент `n_neighbors` конструктора класса `KNeighborsClassifier`.

Используйте метод `fit` класса `KNeighborsClassifier` для обучения модели. И класс `accuracy_score` для оценки доли верных ответов модели:

```
from sklearn.metrics import accuracy_score
```

Для визуализации получившихся результатов используйте метод `plot` класса `pyplot`:

```
import matplotlib.pyplot as plt
```

К пункту 3.

Импортируйте генератор разбиения:

```
from sklearn.model_selection import KFold
```

Выставьте количество блоков в генераторе разбиения, используя аргумент `n_splits=5` конструктора. Присвойте булево значение «True» аргументу конструктора `shuffle` класса `KFold`.

К пункту 4.

Используйте кросс-валидацию для оценки качества алгоритма:

```
from sklearn.model_selection import cross_val_score
```

```
array = cross_val_score(model, X, y, cv=kf, scoring='accuracy')
```

Где «X», «y» объекты и ответы соответственно, «kf» – генератор разбиений, «model» – модель классификации.

К пункту 5.

Импортируйте модель логистической регрессии:

```
from sklearn.linear_model import LogisticRegression
```

Диапазон (вектор) вещественных чисел с определенным шагом удобно получить, используя метод `arange` библиотеки NumPy:

```
import numpy as np  
np.arange(0.01,1,0.01)
```

Значение `C` для логистической регрессии можно установить через атрибут конструктора:

```
LogisticRegression(C=C)
```

К пункту 6.

Масштабировать числовые признаки можно, используя `StandardScaler`:

```
from sklearn.preprocessing import StandardScaler  
scaler = StandardScaler()  
X = scaler.fit_transform(X)
```

Метод `fit_transform` рассчитывает среднее и дисперсию по всем столбцам матрицы `X`, после чего вычитает среднее из каждого значения признака объекта и нормирует на дисперсию.

Д. МЕТОДИЧЕСКИЕ УКАЗАНИЯ К ОФОРМЛЕНИЮ ИСПОЛНИТЕЛЬНОГО ОТЧЕТА

Исполнительный отчет должен включать в себя:

- титульный лист с названием лабораторной работы и фамилией студента;
- цель лабораторной работы;
- листинг кода;
- результаты работы каждого пункта задания в виде графиков Matplotlib с подписанными осями;
- выводы о проделанной работе.

Вопросы к лабораторной работе №1

1. С чем связано название "метрические" у методов классификации и регрессии?
2. На какие виды подразделяются алгоритмы классификации и регрессии по принципу прогнозирования?
3. На каких гипотезах основаны метрические методы?

4. Что означает "k" в методе k-ближайших соседей?
5. Объяснить различие качеств алгоритмов из пункта В.2.
6. Как работает кросс-валидация и для чего она нужна?
7. Как влияет значение C на результаты работы логистической регрессии?
8. Какого типа задачи решает логистическая регрессия: классификации, регрессии, кластеризации?
9. Как и почему масштабирование признаков повлияло на результаты обучения и работы алгоритмов?