Домашнее задание 1
Машинное обучение в прикладных задачах

1 Задание 1: Байесовская классификация

- Открыть в Pandas файл **names.csv** (см. вложения). Ответить на вопросы ниже, используя средства языка Python и необходимых библиотек;
- Разделить данные в выборке на обучающий набор и тестирование (выбор принципа разделения за вами – например, 70% данных в обучении и 30% в тестировании);
- Обучить наивную байесовскую классификацию из файла **Sem2.ipynb** (см. вложения) на тренировочном наборе данных. Затем с помощью метода **classify()** разметить имена по полу в тестировочном наборе данных;
- Посчитайте среднюю долю правильных "ответов"классификатора. Какие еще метрики можно построить, чтобы оценить, насколько хорошо справился с задачей данный классификатор?
- Модифицируйте функцию **get_features()** таким образом, чтобы в качестве целевого признака бралась другая структура (не последняя буква имени). Возможно, это будет набор из первой и последней буквы. Или, например, имя целиком...
- Модифицируйте метод **classify()** так, чтобы вместо логарифмов брались исходные значения вероятностей, а вместо $argmin(...)$ считался функционал $argmax(...)$. Также можете использовать другой метод классификации (из лекций или учебников, или модифицировать его самому методом проб и ошибок).
- Улучшилась ли доля правильных ответов алгоритма после модификации целевого признака и метода **classify()**? Какие выводы можно сделать о выборе целевых признаков и о влиянии классифицирующей функции на результат алгоритма?
- Запустите гауссовский и мультиномиальный классификатор методами из **sklearn.naive_bayes**. Насколько точна классификация в данном

случае? Какой из трех методов оказался точнее (наивный, гауссовский или мультиномиальный)?

2 Задание 2: Классификация ирисов

- Теперь возьмем датасет, содержащий описание цветков ириса и их классификацию по сортам (Setosa, Versicolour, Virginica). Этот набор данных содержится в **sklearn.datasets.load_iris()**.
- Разделите данные на обучение и тестировку (аналогично заданию 1);
- С помощью метода LDA (линейный дискриминантный анализ) реализуйте классификацию сортов ириса на основании признаков датасета;
- По метрикам из задания 1 оцените эффективность классификатора;
- Сравните метод LDA из **sklearn.discriminant_analysis** и реализацию из **Sem3.ipynb** (см. вложения).
- Рассмотрите документацию метода LDA и измените параметры классификатора таким образом, чтобы алгоритм работал эффективнее (например, поменять параметр **solver**). Какие параметры классификатора сильнее сказываются на конечном результате? Чем это может быть объяснимо?

3 Задание 3: kNN

- Открыть датасет **sklearn.datasets.load_wine**, содержащий информацию о трех различных сортах вина (class0, class1, class2). Ответить на вопросы ниже, используя средства языка Python и необходимых библиотек;
- Использовать три подхода к делению выборки на тренировочную и тестовую: KFold, LOO, Stratified KFold. Для воспроизводимости зафиксируйте параметр random_state=42;
- Для каждого из методов кросс-валидации, а также для каждого $k \in [1, 50]$ (число "соседей") прогнать алгоритм ближайших соседей (sklearn.neighbors.KNeighborsClassifier) и посчитать долю правильных ответов. Какая кросс-валидация и при каком значении k дает лучший результат?
- Произведите масштабирование признаков с помощью функции sklearn.preprocessing.scale. Снова найдите оптимальное k на трех разных кросс-валидациях. Чему оно равно? Изменилось ли оно? Изменился ли оптимальный метод валидации?