1. Министерство науки и высшего образования Российской Федерации
2. Санкт-Петербургский Политехнический Университет Петра Великого
3. —
4. Институт компьютерных наук и кибербезопасности
5. Высшая школа технологий искусственного интеллекта

**Курсовая работа**

по дисциплине «Машинное обучение. Часть 1»

1. Выполнил: студент группы
2. 5140201/30301 С.П. Хомец

*<подпись>*

1. Проверил: Л.В. Уткин
2. д.т.н., профессор
3. *<подпись>*

Санкт-Петербург

2024 г.

1. **Введение**
   1. **Использованный датасет**

В качестве датасет был выбран Parkinsons Disease Data Set. Он содержит описание набора данных о биомедицинских измерениях голоса для диагностики болезни Паркинсона. Данные включают биомедицинские измерения голоса 31 человека, из которых 23 имеют болезнь Паркинсона. Цель данных – различать здоровых людей от тех, у которых есть болезнь Паркинсона, согласно колонке “status”, где 0 – здоров, 1 – болезнь Паркинсона.

Информация об атрибутах:

* ‘name’ – имя субъекта и номер записи;
* ‘MDVP:Fo(Hz)’, ‘MDVP:Fhi(Hz)’, ‘MDVP:Flo(Hz)’ – различные характеристики частоты голоса;
* ‘MDVP:Jitter(%)’, ‘MDVP:Jitter(Abs)’, ‘MDVP:RAP’, ‘MDVP:PPQ’, ‘Jitter:DDP’ – различные меры вариации в фундаментальной частоте;
* ‘MDVP:Shimmer’, ‘MDVP:Shimmer(dB)’, ‘Shimmer:APQ3’, ‘Shimmer:APQ5’, ‘MDVP:APQ’, ‘Shimmer:DDA’ – различные меры вариации в амплитуде;
* ‘NHR’, ‘HNR’ – две меры соотношения шума к тональным компонентам в голосе;
* ‘status’ – здоровье субъекта;
* ‘RPDE’, ‘D2’ – две нелинейные меры динамической сложности;
* ‘DFA’ – экспонента фрактального масштабирования сигнала;
* ‘spread1’, ‘spread2’, ‘PPE’ – три нелинейные меры вариации фундаментальной частоты.

Сам набор данных небольшой, содержит 195 примеров, 22 признака.

* 1. **Использованные модели**
     1. **SVM**

Имеется m-мерное пространство . Каждый пример представлен точкой в нашем случае требуется бинарная классификация, поэтому каждому соответствует метка .

Требуется построить разделяющую гиперплоскость:

Такую, чтобы точки, лежащие по разные стороны от нее, имели разные метки:

При этом, оптимальной гиперплоскостью является та, которая максимизирует ширину полосы (зазор) между классами, при этом сама находится по середине этой полосы.

Ставится следующая задача оптимизации:

При условии:

Функция Лагранжа:

*.*

Необходимые условия седловой точки:

Используем условия седловой точки и переходим к двойственной задаче:

При ограничениях:

Тогда функция нашей разделяющей гиперплоскости представляется в виде:

С учётом перехода в пространство большей размерности (для решения задачи при линейно не разделимых данных) функция разделяющей гиперплоскости в новом пространстве имеет вид:

Также требуется ввести значение штрафа, для неверно предсказанных примеров:

, C – штрафной параметр.

**Таким образом, требуется настроить параметр штрафа С, выбрать оптимальное ядро и его параметры.**

* + 1. **Наивный байесовский классификатор**

В основе метода лежит теорема Байеса, которая связывает апостериорные и априорные вероятности.

Пусть C – множество классов, x – пример, y – метка класса:

Наивность заключается в том, что данный метод предполагает независимость признаков при условии класса.

Класс, к которому относится пример, будем тем, вероятность принадлежности к которому для примера максимальная.

* + 1. **Бэггинг**

Идея метода заключается в использовании множества слабых классификаторов, которые чуть лучше, чем случайное угадывание, для предсказывания класса для примера, а затем объединение этих предсказаний, путём использования голосования.

Формально:

Обучающая выборка:

Случайно выбираем t элементов из D с возвращением s раз:

Обучаемся на каждом и получаем последовательность s выходов:

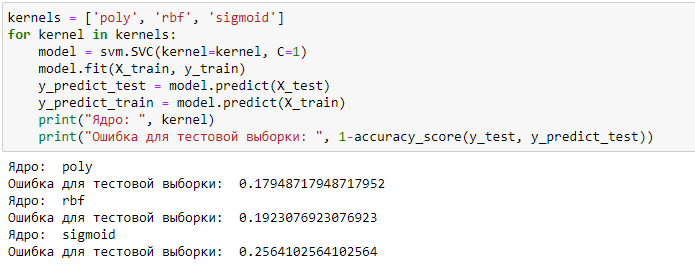
Получаем итоговый классификатор, который уже будет являться сильным:

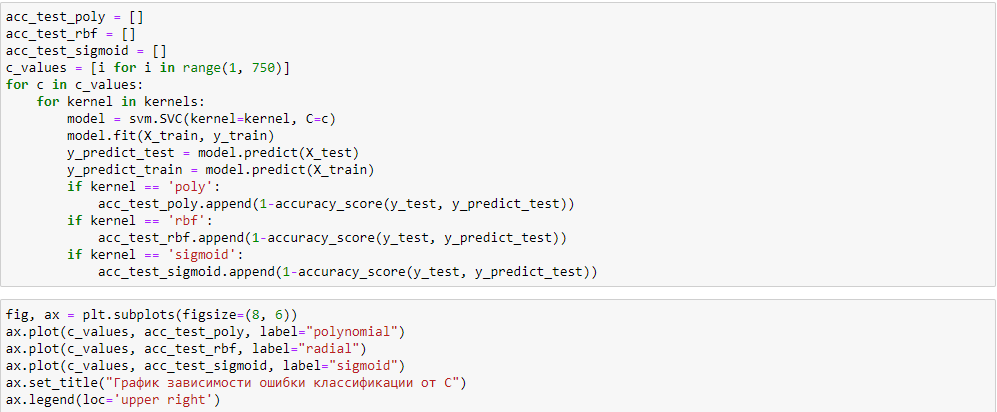
Математическое ожидание ошибки предсказания:

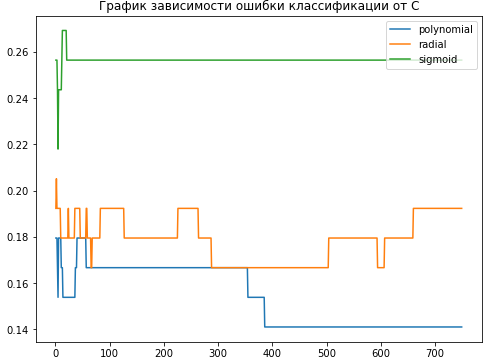
Таким образом, если ошибки не коррелированы, то мы получим ошибку в s раз меньше, тем средняя ошибка всех моделей, но на практике ошибки сильно коррелированы, так как данные в моделях могу совпадать. Однако ошибка сильной модели всегда меньше или равна ошибкам всех слабых моделей.

1. **Задание 1**
   1. **SVM**

Подбор оптимальных параметров для SVM:







Таким образом, оптимальными параметрами для SVM оказались:

Ядро: полиномиальное со степенью 2.

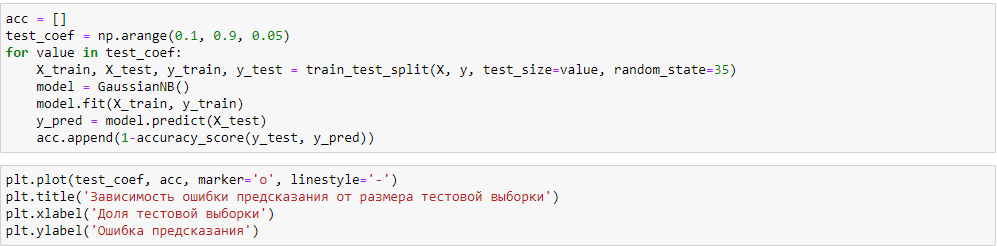
Значение штрафа: 400 (дальше точность выходит на плато).

**Полученная ошибка: 0.141**

* 1. **Наивный байесовский классификатор**

Среди Bernoulli Naïve Bayes, Multinomial Naive Bayes, Gaussian Naïve Bayes был выбран Gaussian, так как данные представляют собой непрерывные случайные величины.

В качестве настраиваемого параметра: размер выборки.





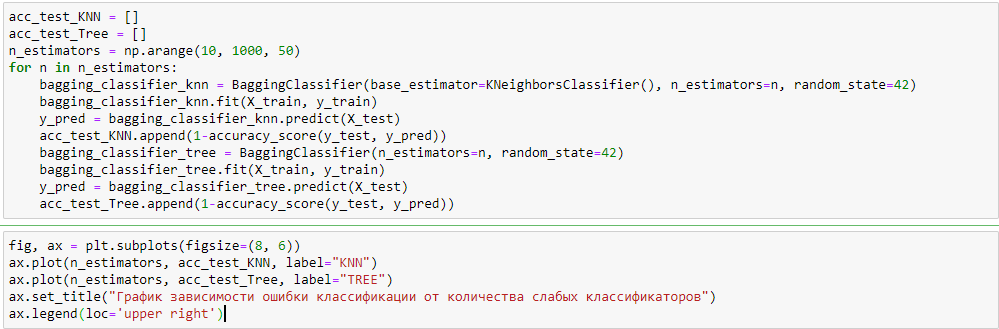
Оптимальная доля тестовой выборки оказалось равной: 0.1

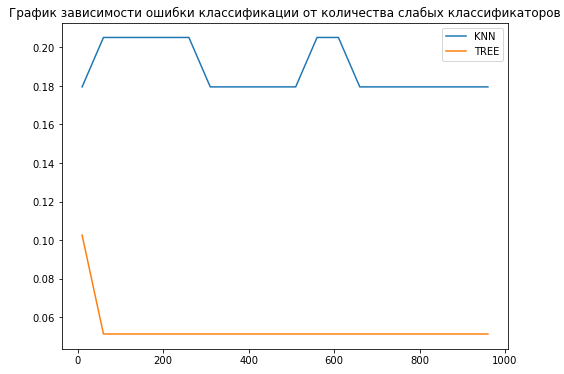
**Полученная минимальная ошибка оказалась равна: 0.2**

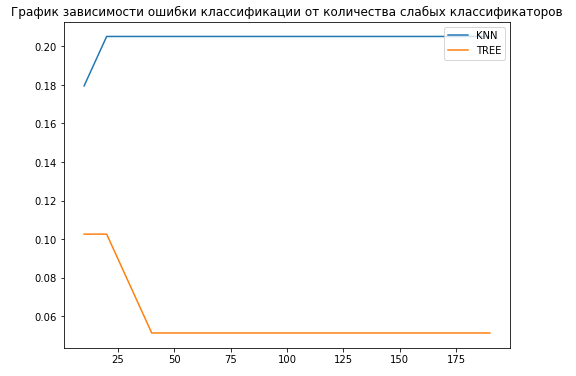
* 1. **Бэггинг**

Для бэггинга можно подбирать слабый классификатор и количество слабых классификаторов.

В качестве модели слабого классификатора будем сравнивать Decision Tree и KNN.

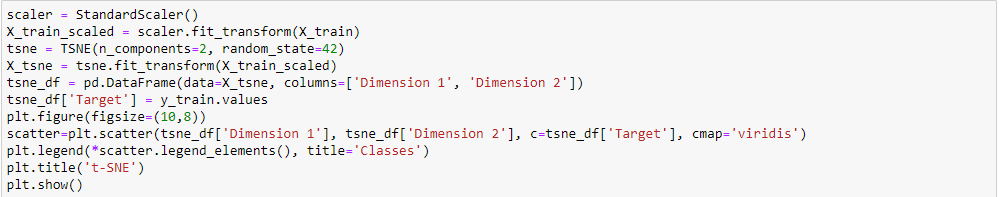


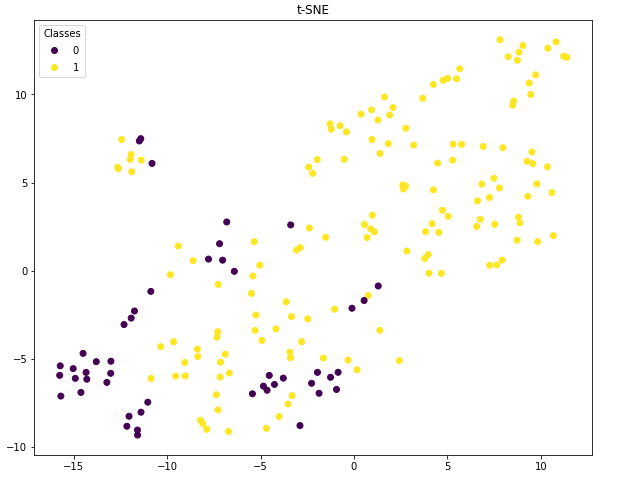




Таким образом, лучше оказалась модель со слабым классификатором - Decision Tree, которая после количества слабых классификаторов равного 40 выходит на плато, достигая **минимальной ошибки классификации = 0.051.**

* 1. **t-SNE**

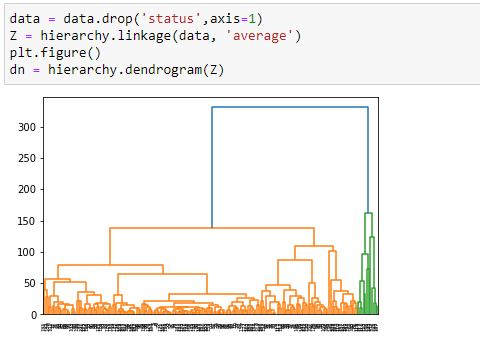
****

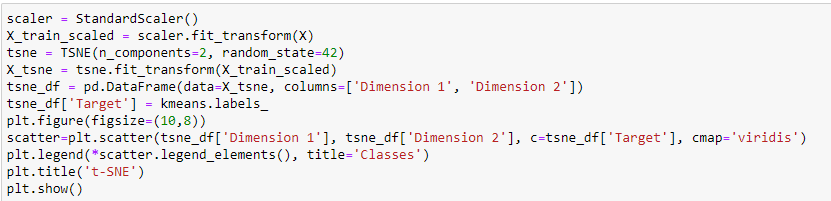
****

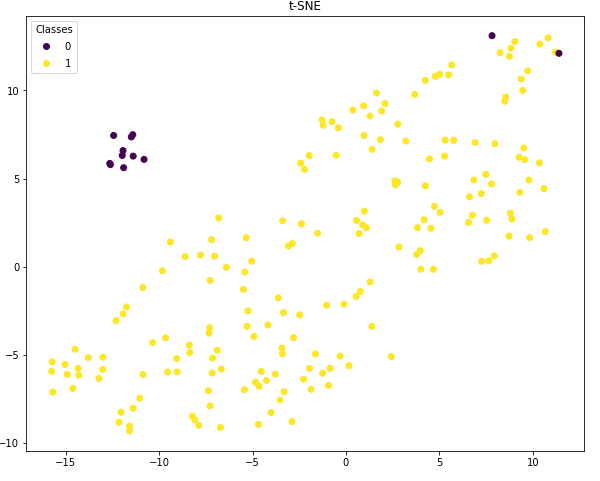
1. **Задание 2**

Исходя из результатов, полученных в предыдущем задание, наилучшая модель по вероятности ошибочной классификации на тестовых данных, ожидаемо, - Бэггинг, построенный на 40 деревьях решений.

1. **Задание 3**
   1. **Описание метода k-средних**
2. Случайно разбиваем объекты на кластеров.
3. Вычисляем центры тяжести кластеров:
4. Вычисляем расстояния от точки до всех . Записываем в тот класс, расстояние до центра тяжести которого минимальное.
5. Повторяем шаг 3 для всех
6. Если хотя бы один кластер изменился, то переходим на шаг 2, иначе завершение.
   1. **Реализация**







**Ошибка кластеризации: 0.277**

1. **Задание 4**
   1. **Описание метода Лассо**

Математическое ожидание отклонения имеет вид:

Логистическая регрессия получается в результате использования логарифмического функционала риска:

Это эквивалентно замене логистической или сигмоидной функцией:

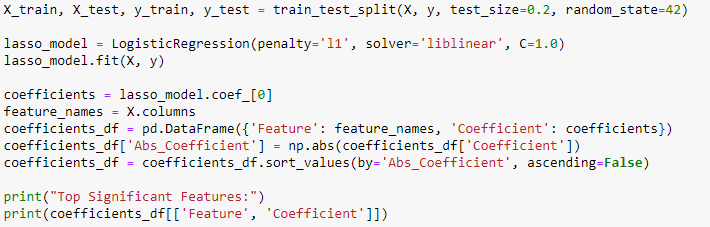
Получим вероятности для классификации:

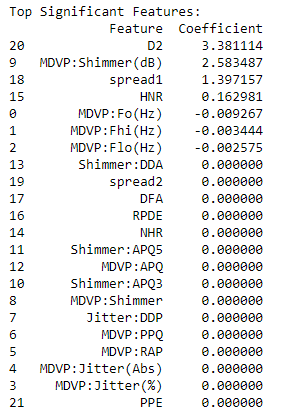
Ищем коэффициенты , как ОМП оценки:

Берём отрицательный логарифм функции правдоподобия, накладываем ограничения на коэффициенты . Получаем требуемую модель:

Ограничения на :

Модель:

* 1. **Реализация**

****

Параметры, у которых значения коэффициента равны 0, являются не влияющими на результат. Чем выше коэффициент, тем выше влияние.

1. **Задание 5**

Используем автокодер для понижения размерности до 7. Достаём закодированные данные из скрытого слоя и применяем бэггинг из первого задания.

**Изображение выглядит как текст, Шрифт, снимок экрана

Автоматически созданное описание**

**Изображение выглядит как текст, Шрифт, снимок экрана, линия

Автоматически созданное описание**

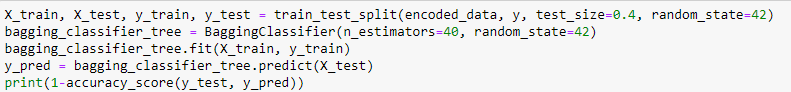
Получаем, что **точность ухудшилась и стала равной 0.19**.

Теперь используем регуляризацию L1, чтобы занулить выходы некоторых нейронов внутреннего слоя, чтобы сократить размерность.

Изображение выглядит как текст, снимок экрана, Шрифт, линия

Автоматически созданное описание

Используем также бэггинг из первого задания.



**Точность ухудшилась и стала равной 0.23.**

В итоге, предпочтительнее оказалось уменьшение скрытых слоёв, чем разряженность скрытого слоя.

Изображение выглядит как снимок экрана, диаграмма

Автоматически созданное описание

Зашумленный автокодер используется для того, чтобы помогать избавляться от шумов во входных данных. Для этого во входные данные вносится гауссовский шум.

Изображение выглядит как текст, Шрифт, снимок экрана

Автоматически созданное описание

Для зашумленного автокодера точность оказалось равной 0.25, что является худшим результатом среди рассмотренных автокодеров.