Министерство науки и высшего образования Российской Федерации Федеральное государственное бюджетное образовательное учреждение высшего образования

ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ СИСТЕМ УПРАВЛЕНИЯ И РАДИОЭЛЕКТРОНИКИ (ТУСУР)

Кафедра автоматизации обработки информации (АОИ)

Вариант 2

Отчёт к Лабораторной работе №3

по дисциплине «Нейронные сети и их применение»

Выполнил:

Студент группы 422–M1

\_\_\_\_Белоус Г.В.

Принял:

К.т.н., Доцент кафедры АОИ

\_\_\_\_Аксёнов С.В.

Оглавление

[1 Введение 3](#_Toc131164226)

[1.1 Цель лабораторной работы 3](#_Toc131164227)

[1.2 Задание на лабораторную работу 3](#_Toc131164228)

[2 Ход выполнения работы 5](#_Toc131164229)

[2.1 Обучение четырех регрессионных модели, построенных на наборе данных с помощью четырёх алгоритмов 5](#_Toc131164230)

[2.2 Исследовать работы алгоритмов с разными значениями гиперпараметров 8](#_Toc131164231)

[2.3 Реализация приложения осуществляющее загрузку модели 9](#_Toc131164232)

[3 Вывод 10](#_Toc131164233)

# Введение

## 1.1 Цель лабораторной работы

Построения регрессоров, подбора гиперпараметров моделей и анализа качества работы регрессионных моделей.

## **1.2 Задание на лабораторную работу**

Написать программу на Python, которая обучает четыре регрессионных модели, построенных на наборе с помощью четырёх алгоритмов: линейный регрессор, полиномиальный регрессор, регрессор, основанный на случайном лесе и один из следующих инструментов: Gaussian Process Regression, Support Vector Regression, Gradient Boosting Regressor или AdaBoost Regressor.

Выбрать признаки, использующиеся при обучении, и, если необходимо, выполнить их предобработку. Разделить выборку на обучающую и тестовую.

В работе необходимо исследовать работу алгоритмов с разными значениями гиперпараметров.

Для моделей на основе деревьев вывести значения важности признаков.

Выбрать наилучшую модель из полученных регрессоров.

Сохранить лучшую модель (pickle). Спроектировать и реализовать приложение (настольное, Web- или чат-бот) на Python, осуществляющее загрузку модели, проверку корректности ввода данных, требуемых для регрессионной модели. Для реализации программы можно использовать любой инструмент (Flask, Django, Tkinter и т.д.)

Написать короткий отчет по работе, включив в него программы с комментариями, значения качества моделей (коэффициент детерминации, среднюю квадратичную и среднюю абсолютную ошибки).

Содержание отчета.

1. Титульный лист
2. Задача по полученному варианту (выбор варианта осуществляется по последнему номеру в зачетной книжке, если он равен 0 или 5 – то берем вариант 1, если 2 или 6 – вариант 2, если 3 или 7 то вариант 3, если 4 или 8 - вариант 4, если 5 – вариант 5, если 9 – вариант 6)
3. Подобранные гиперпараметры для каждого алгоритма
4. Величина метрик для каждого алгоритма
5. Демонстрация работы приложения с загруженной моделью
6. Вывод

# Ход выполнения работы

Датасет, данные из которого будут обрабатываться в данной работе, представляет из себя набор атрибутов физико-химических характеристик белого вина с конечной оценкой (quality). В наборе данных чуть меньше 5 тыс. записей, что позволяет использовать датасет полностью.

Перед созданием моделей набор данных был проверен на отсутствие пропущенных записей и использован метод train\_test\_split, разделяющий всю выборку на матрицу признаков для тренировки модели и вектор целевого значения quality (train), а так же матрицу признаков и вектор предсказываемых значений для проверки моделей (test).

## Обучение четырех регрессионных модели, построенных на наборе данных с помощью четырёх алгоритмов

В работе будут сравниваться четыре регрессионные модели: LinearRegression, PolynomialFeatures, RandomForestRegressor и AdaBoostRegressor.

Библиотека sklearn позволяет использовать все эти алгоритмы и строить модель по одному принципу.

Для подбора оптимальных гиперпараметров был использован инструмент GridSearchCV, производящего автоматизированный подбор оптимальных гиперпараметров. В линейной регрессии нет гиперпараметров, которые можно менять, в полиномиальной регрессии можно менять степень, в случайном лесе уже множество гиперпараметров, которые можно подобрать, а AdaBoost будет использовать в качестве базовой модели наилучшую из приведённых выше.

На картинках 2.1, 2.2, 2.3 и 2.4 показаны скриншоты кода, создающего модели:

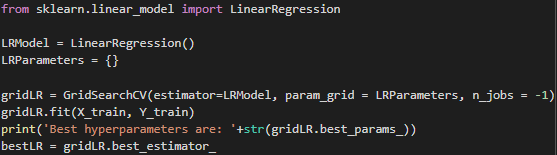


Рисунок 2.1 – Программный код для создания модели линейной регрессии

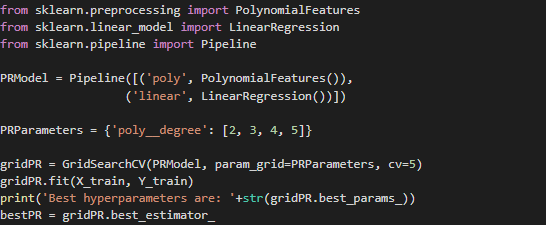


Рисунок 2.2 – Программный код для создания модели полиномиальной регрессии

В результате вычислений для модели случайного леса, оптимальным гиперпараметром, а именно степень полинома, стал {'poly\_\_degree': 2}.

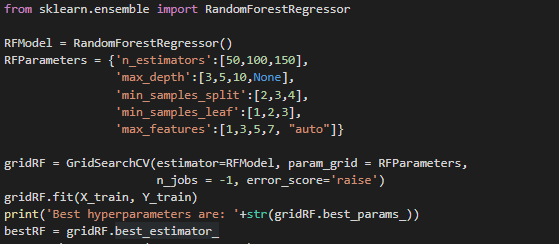


Рисунок 2.3 – Программный код для создания модели случайного леса

В результате вычислений для модели случайного леса, оптимальными гиперпараметрами стали {'max\_depth': None, 'max\_features': 3, 'min\_samples\_leaf': 1, 'min\_samples\_split': 2, 'n\_estimators': 150}.

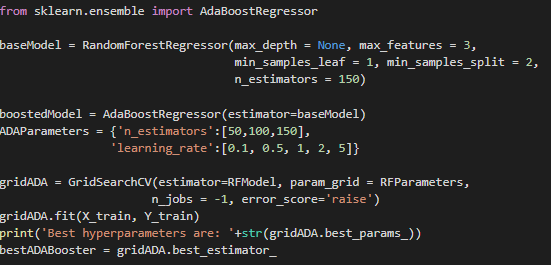


Рисунок 2.4 – Программный код для создания модели AdaBoost с базовой моделью случайного леса

Модель случайного леса была взята за базовую в следствии наилучших показателей по метрикам, что продемонстрировано в таблице 2.1.

## Исследовать работы алгоритмов с разными значениями гиперпараметров

Для итоговой оценки были взяты четыре модели с оптимальными гиперпараметрами, подобранные с помощью инструмента GridSearchCV. В таблице 2.1 показаны результаты метрик для каждого алгоритма:

Таблица 2.1 – Показатели метрик алгоритмов регрессоров

|  |  |  |  |
| --- | --- | --- | --- |
| Алгоритм \ Метрика |  |  |  |
| LinearRegression | 0.317 | 0.534 | 0.573 |
| PolynomialFeatures | 0.373 | 0.49 | 0.551 |
| RandomForestRegressor | 0.546 | 0.355 | 0.429 |
| AdaBoostRegressor | 0.559 | 0.345 | 0.432 |

На рисунке 2.5 показано распределение важности атрибутов для модели использующей лес решений:

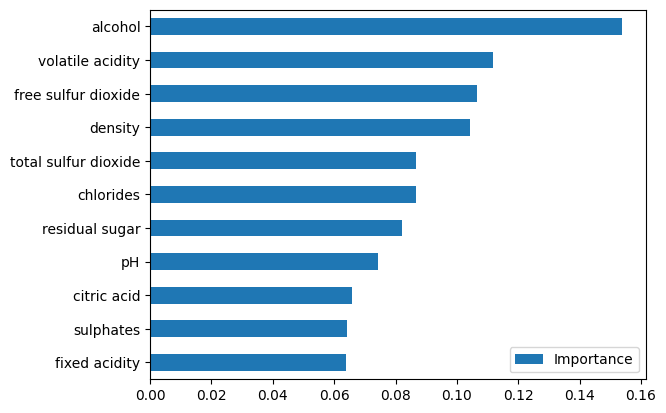


Рисунок 2.5 – Важность атрибутов в модели Random Forest

## Реализация приложения осуществляющее загрузку модели

Сохранение модели производится с помощью библиотеки pickle, размер модели в случае с сохранением AdaBoostRegressor составляет 1.3 GB.

Модель загружается в оконное приложение, реализованное с помощью библиотеки tkinter. В приложении есть четыре активных элемента: текстбокс для ввода данных на основе которых будет строится предсказание, кнопка “predict”, которая будет запускать методы предсказания загруженной модели, лейбл на котором будет появляться результат предсказания или сообщение об ошибки и кнопка “browse”, которая открывает файловый диалог для выбора необходимого файла с разрешением “.pkl”.

Результат работы программы и её интерфейс показан на рисунке 2.6:

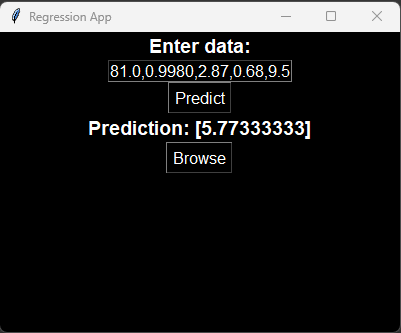


Рисунок 2.6 – Интерфейс программы реализующего загрузку модели и предсказания значения

# Вывод

В ходе выполнения лабораторной работы были построены четыре модели регрессора основанные на четырех алгоритмах: LinearRegression, PolynomialFeatures, RandomForestRegressor и AdaBoostRegressor.

Наилучшим, исходя из метрик, оказался алгоритм AdaBoostRegressor, так как базовой моделью для него был выбран RandomForestRegressor, который алгоритм AdaBoost улучшил за счёт своего ансамблевого подхода и взвешивания предшествующих результатов каждой рассчитанной модели, по сути, представляя собой ансамбль ансамбля деревьев решений. Однако взамен наилучших показателей метрик, минусом данного алгоритма является его долгие вычисления.

Исходя из рисунка с важностью признаков, можно сделать вывод, что содержания алкоголя в вине является наиболее важным фактором, формирующим итоговую оценку.

Так же из результатов в таблице 2.1 видно, что для модели AdaBoostRegressor MAE больше, чем у RandomForestRegressor, хотя остальные две метрики ниже.

Листинг программы находится в соответствующей папке Lab3 по ссылке на репозиторий: https://github.com/peremichka256/NSiIP