Московский Авиационный Институт

(Национальный Исследовательский Университет)

Факультет информационных технологий и прикладной математики

Кафедра вычислительной математики и программирования

Лабораторная работа №1 по курсу

«Искусственный интеллект (Машинное обучение)»

Студент: Алексеев В.Е.
Группа: М8О–306Б–19
Преподаватель: Ахмед Самир Халид
Оценка: _____
Дата: _____
Подпись: _____

Москва, 2022

1. Постановка задачи

1) Реализовать следующие алгоритмы машинного обучения: Linear/ Logistic Regression, SVM, KNN, Naive Bayes в отдельных классах.

2) Данные классы должны наследоваться от BaseEstimator и ClassifierMixin, иметь методы fit и predict.

3) Вы должны организовать весь процесс предобработки, обучения и тестирования с помощью Pipeline.

4) Вы должны настроить гиперпараметры моделей с помощью кросс валидации, вывести и сохранить эти гиперпараметры в файл, вместе с обученными моделями.

5) Проделать аналогично с коробочными решениями.

6) Для каждой модели получить оценки метрик: Confusion Matrix, Accuracy, Recall, Precision, ROC_AUC curve.

7) Проанализировать полученные результаты и сделать выводы о применимости моделей.

8) Загрузить полученные гиперпараметры модели и обученные модели в формате pickle на гит вместе с jupyter notebook ваших экспериментов.

2. Подготовка

В прошлой лабораторной работе мы подготовили данные для подачи их в модели машинного обучения: закодировали категориальные признаки, удалили линейно зависимые признаки, пронормировали величины. Загрузим обработанные таблицы через pd.read_csv().

3. Linear Regression (Линейная регрессия)

В работе реализованы сопсоб линейной регрессии на основе аналитического решения. То есть мы можем точно вычислить формулу, по которой будет находится матрица весов.

4. Logistic Regression (Логистическая регрессия)

Логистическая регрессия описывает распределение признаков. Лосс функция – кросс-энтропия. На выходе получаем вероятность принадлежности объекта к классу.

5. SVM (Метод опорных векторов)

SVM или метод опорных векторов отличается от линейной регрессии тем, что мы ищем такую гиперплоскость, которая максимально удалена от каждой группы классов. Таким образом, мы решаем проблему, когда наш объект вблизи границы класса. Для этого достаточно поменять функцию ошибки.

6. KNN (Метод k-ближайших соседей)

Для метода k-ближайших соседей мы задаем метрику на пространстве и ищем первые k ближайших соседей к нашему классу. Присваиваем нашему объекту класс, который имеет больше всего соседей, используя обычную евклидову метрику.

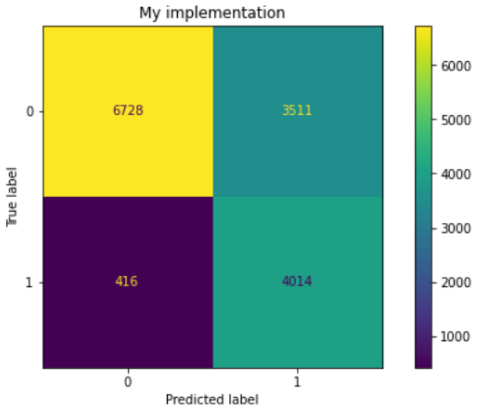
7. Naive Bayes (Наивный байесовский классификатор)

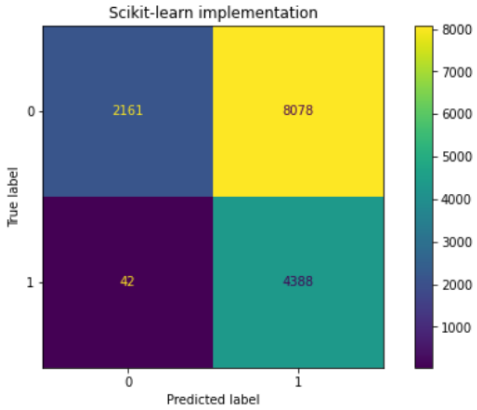
В основе наивного байесовского классификатора лежит теорема Байеса. Теорема Байеса позволяет переставить местами причину и следствие. Зная, с какой вероятностью причина приводит к некоему событию, теорема позволяет рассчитать вероятность того, что именно эта причина привела к наблюдаемому событию. Алгоритм называется наивным, потому что делается предположение условной независимости.

8. Результат работы алгоритмов

Linear Regression (Линейная регрессия)

```
Accuracy: 0.7322925898152567
Accuracy sklearn model: 0.44645170086577135
Recall: 0.9060948081264109
Recall sklearn model: 0.44645170086577135
Precision: 0.575310193183603
Precision sklearn model: 0.44645170086577135
```

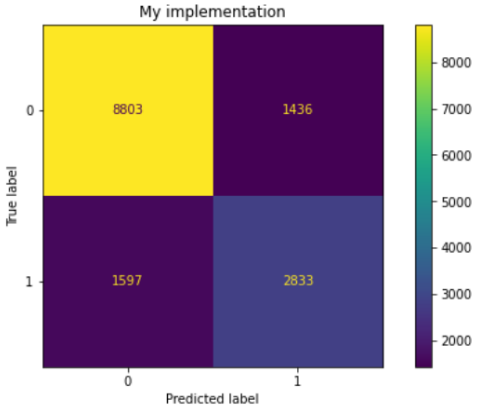


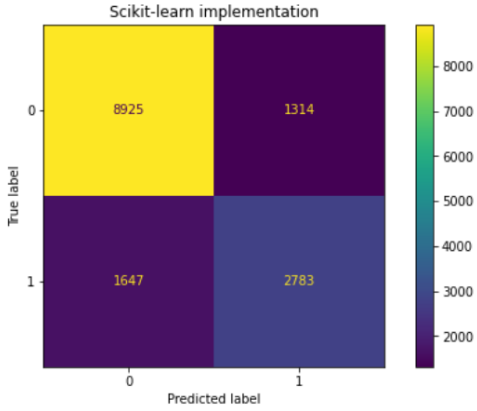


Как показали тесты, линейная модель из коробки не справляется с задачей. А моя реализация, хоть и не идеальные результаты, но явно лучше.

Logistic Regression (Логистическая регрессия)

```
Accuracy: 0.7932374394982616
Accuracy sklearn model: 0.798145749539846
Recall: 0.6395033860045146
Recall sklearn model: 0.798145749539846
Precision: 0.575310193183603
Precision sklearn model: 0.798145749539846
```

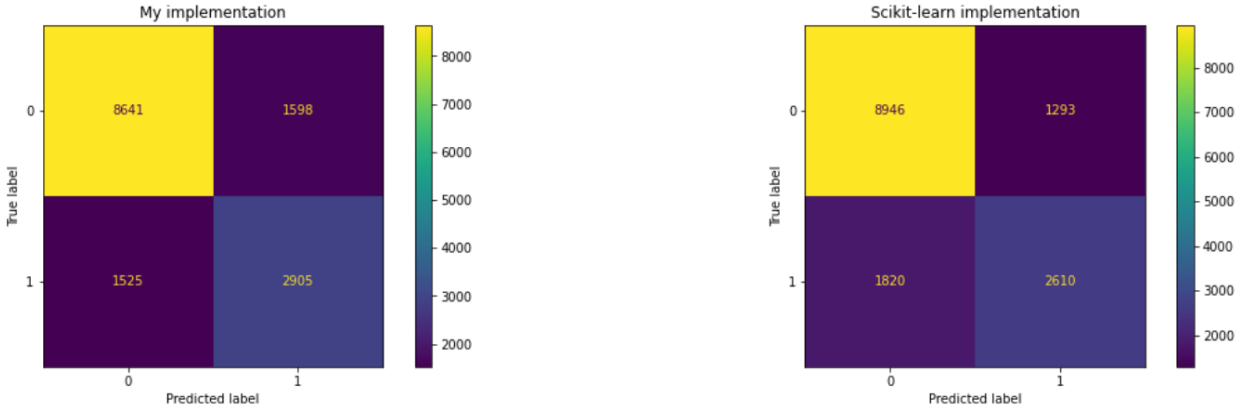




Результаты логистических регрессий почти идентичны. Это значит, что модели хорошо справляются с задачей.

SVM (Метод опорных векторов)

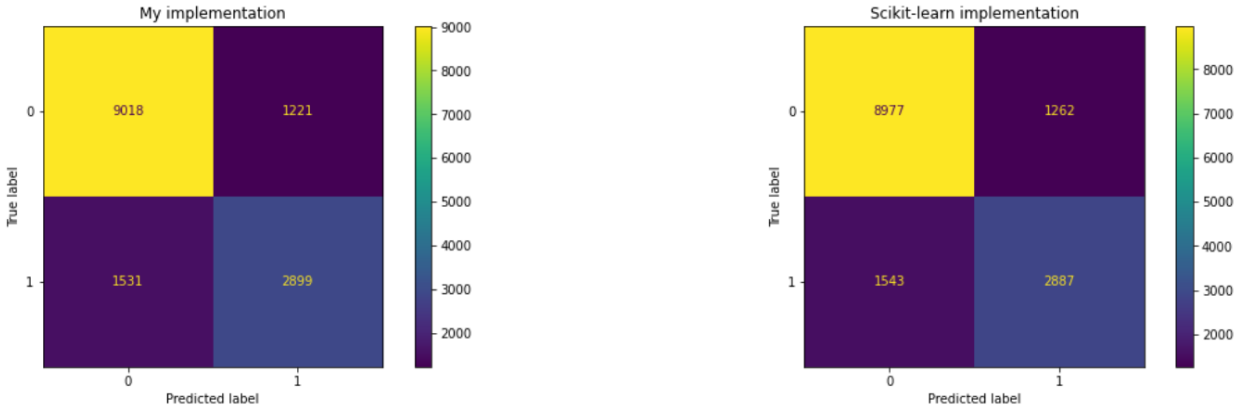
```
Accuracy: 0.7871020519462812
Accuracy sklearn model: 0.7877837616742791
Recall: 0.6557562076749436
Recall sklearn model: 0.7877837616742791
Precision: 0.575310193183603
Precision sklearn model: 0.7877837616742791
```



Коробочный SVM получает хорошие результаты классификации. Мой SVM оказалася немного хуже, но все равно не так все печально.

KNN (Метод k-ближайших соседей)

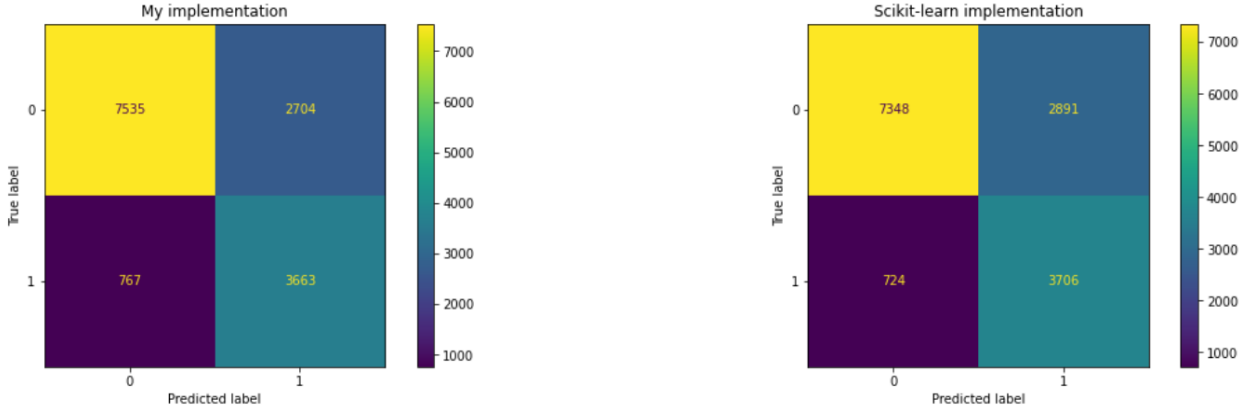
```
Accuracy: 0.8123934828550003
Accuracy sklearn model: 0.8087804212966119
Recall: 0.6544018058690745
Recall sklearn model: 0.8087804212966119
Precision: 0.575310193183603
Precision sklearn model: 0.8087804212966119
```



KNN, можно сказать, очень хорошо справляется с поставленной задачей.

Наивный байесовский классификатор

```
Accuracy: 0.7633785534119572
Accuracy sklearn model: 0.7535619333287886
Recall: 0.82686230248307
Recall sklearn model: 0.7535619333287886
Precision: 0.575310193183603
Precision sklearn model: 0.7535619333287886
```



Результаты наивных байесовских классификаторов почти идентичны. Это значит, что модели хорошо справляются с задачей.

9. Выводы

В данной лабораторной работе я познакомился с sklearn и базовыми алгоритмами машинного обучения. Также я попробовал реализовать свои версии этих алгоритмов. Так же я изучил линейные модели классического машинного обучения. В результате работы были реализованы все алгоритмы поставленной задачи. Для каждого алгоритма были подобраны лучшие гиперпараметры. Для этого использовалась функция GridSearchCV. Как показали результаты, ручные реализации базовых классификаторов не сильно хуже тех, что есть в sklearn (а некоторые показали себя даже лучше). Все модели, кроме линейной, дали довольно хорошие результаты. Я думаю, можно существенно поднять accuracy, если мы дополнительно поработаем с данными. А именно выделим больше признаков.