Московский Авиационный Институт

(Национальный Исследовательский Университет)

Факультет информационных технологий и прикладной математики

Кафедра вычислительной математики и программирования

**Лабораторная работа №1 по курсу**

**«Машинное обучение»**

**Разработка линейных моделей**

Студент: Гаптулхаков Руслан Рамилевич

Группа: М80 - 308Б -19

Дата: 17.05.2022

Оценка: \_\_\_\_\_\_\_\_\_\_\_

Подпись: \_\_\_\_\_\_\_\_\_\_\_

Москва, 2022

# **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 Вы должны организовать весь процесс предобработки, обучения и тестирования с помощью Pipeline Вы должны настроить гиперпараметры моделей с помощью кросс валидации, вывести и сохранить эти гиперпараметры в файл, вместе с обученными моделями;
3. Проделать аналогично с коробочными решениями; Для каждой модели получить оценки метрик: Confusion Matrix, Accuracy, Recall, Precision, ROC\\_AUC curve;
4. Проанализировать полученные результаты и сделать выводы о применимости моделей;
5. Загрузить полученные гиперпараметры модели и обученные модели в формате pickle на гит вместе с Jupyter Notebook ваших экспериментов.

# **Подготовка**

В прошлой лабораторной работе мы подготовили данные для подачи их в модели машинного обучения. А именно: добавили новых признаков, закодировали категореальные признаки, удалили линейно зависимые признаки, отнормировали величины. Загрузим обработанные таблицы через pd.read\_csv(). Теперь мы можем разделить нашу выборку на train/val части. Для этого воспользуемся функцией train\_test\_split из scikit-learn.

# **Метод k-ближайших соседей**

Задаём метрику на пространстве и ищем первые k ближайших соседей к нашему классу. Присваиваем нашему объекту класс, который имеют больше всего соседей. Мы будем использовать обычную евклидову метрику.



# **Линейная регрессия**

В работе реализованы два вида линейной регрессии. Первый способ использует градиентный спуск. Второй способ -- аналитическое решение. То есть мы можем точно вычислить формулу, по которой будет находится матрица весов.





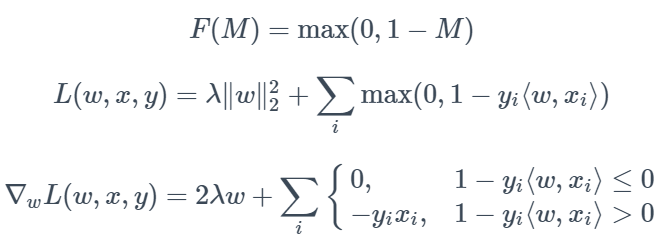
# **Логистическая регрессия**

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



# **Метод опорных векторов**

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





# **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, эта теорема позволяет расчитать вероятность того что именно эта причина привела к наблюдаемому событию. Алгоритм называется наивным, потому что делается предположение условной независимости.



# **Подбор гиперпараметров**

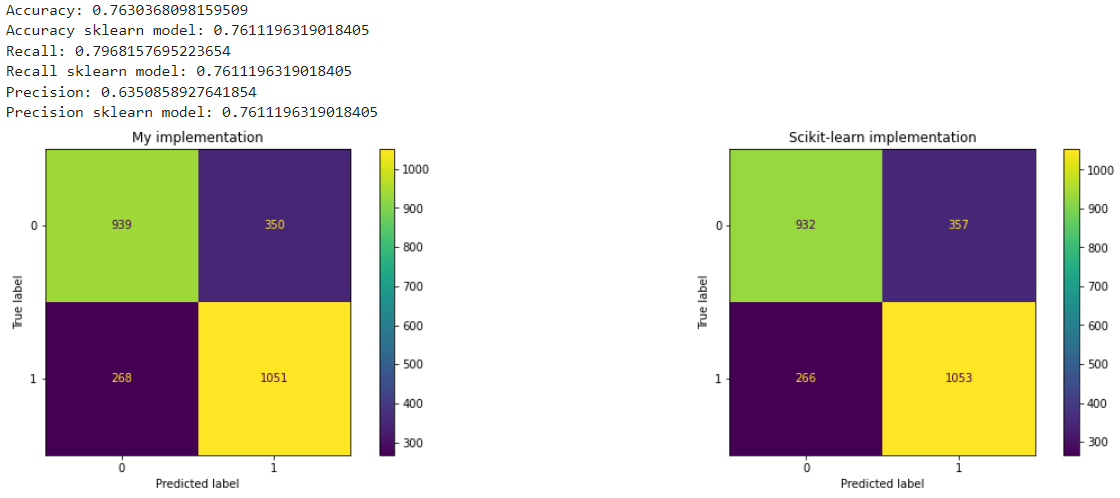
Для наглядной проверки работ модели будем строить сonfusion matrix для нашего и коробочного алгоритмов. Для этого нам понадобится функция.



Также будем выводить результаты различных известных метрик.

# **Результаты работ алгоритмов**

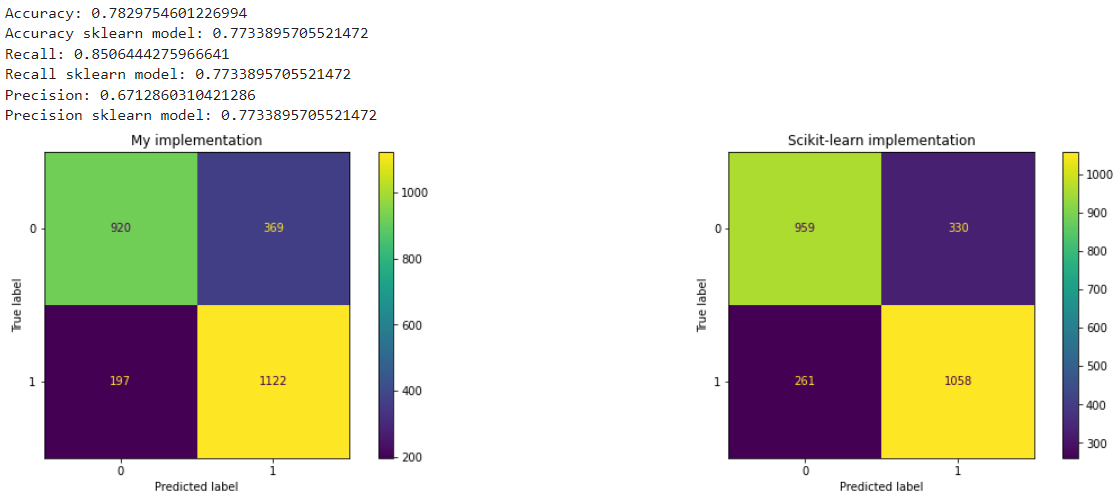
## **K-ближайших соседей**





KNN довольно хорошо справляется с поставленной задачей.

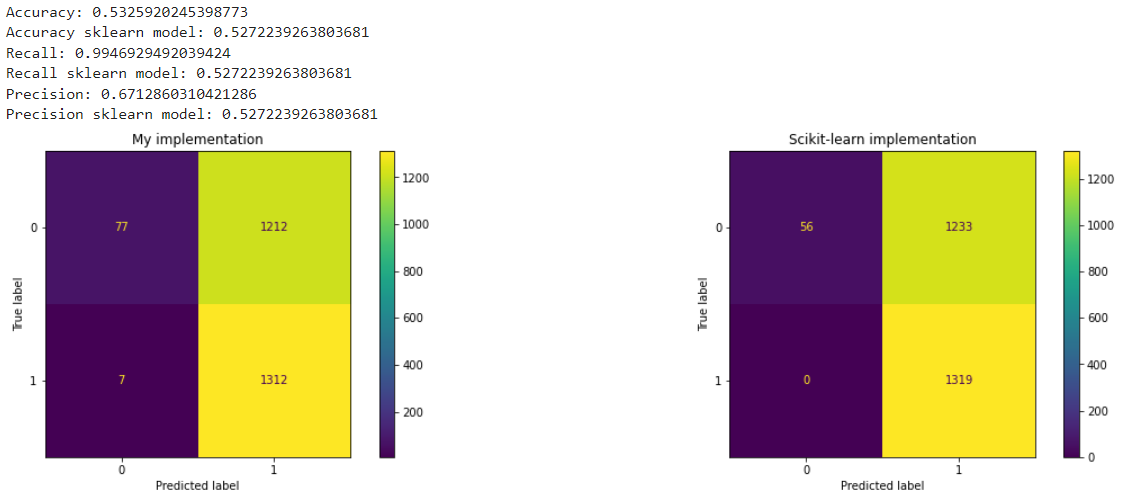
## **Логисточеская ркгркссия**





Результаты логистических регрессий схожи. Модели хорошо справляются с задачей.

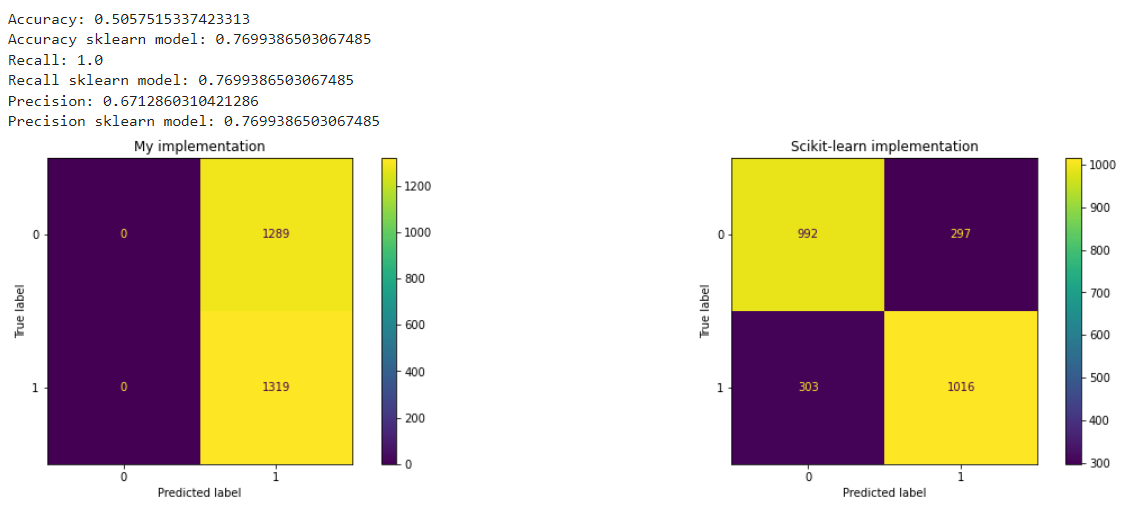
## **Линейная регрессия**





Линейная модель совсем не справляется с задаче. Можно сделать вывод, что наши классы не имеют линейную зависимость от признаков.

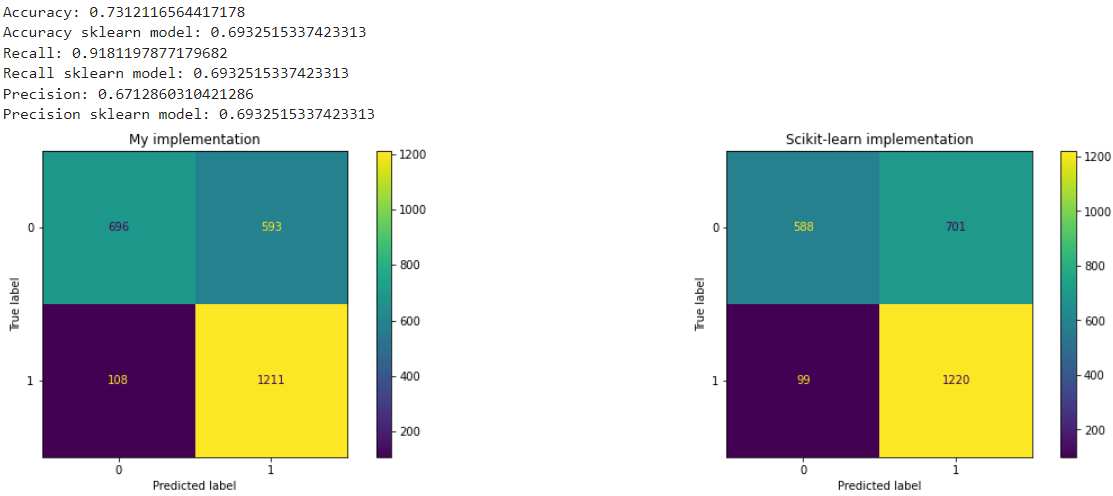
## **Метод опорнных векторов**





Коробочный SVM получает хорошие результаты классификции. Мой SVM не хочет предсказывать объекты 0 класса.

## **Наивный байесовский классификатор**



Моя модель Байесовского классификатора справляется лучше коробочного.

1. **Вывод**

В данной лабораторной работе я изучил линейные модели класического машинного обучения. Были реализованы все алгоритмы поставленные в задаче. Для каждого алгоритма были подобраны лучшие гиперпараметры. Для этого была использована функция GridSearchCV. Все модели, кроме линейной, дали результаты около 0.77 это довольно хороший результат. Я думаю, можно существенно поднять accuracy, если мы дополнительно поработаем с данными. А именно выделим больше признаков.