**Задание для Машинное обучение 4\_2**

Для выполнения практического задания будем использовать данные о диабете у женщин. Данные хранятся в файле «diabetes.csv».

Данные содержат следующие характеристики:

1. Pregnancies – число случаев беременности
2. Glucose – концентрация глюкозы в крови
3. BloodPressure – артериальное диастолическое давление (мм рт. ст.)
4. SkinThickness – толщина кожной складки трехглавой мышцы (мм)
5. Insulin – 2-х часовой сывороточный инсулин
6. BMI – индекс массы тела
7. DiabetesPedigreeFunction – числовой параметр наследственности диабета
8. Age – возраст

Outcome – **целевая переменная**: 1 – наличие заболевания, 0 – отсутствие

**Задача:** построить линейные модели (логистическая регрессия, метод опорных векторов), определяющие есть или нет заболевание по набору характеристик 1-8. Сравнить качество моделей.

Для каждой модели:

1. Загрузите данные в DataFrame с помощью функции read\_csv библиотеки pandas.

2. Как наблюдения (объекты) распределились по классам?

Сколько наблюдений в каждом классе? Для ответа на вопрос используйте метод

value\_counts().

3. Разделите данные на признаки и ответы, а затем на обучающую и тестовую выборки.

4. Обучите модель.

5. Оцените качество модели на тестовой выборке.

Используйте для этого функцию classification\_report. Что можно сказать о модели?

6. Стандартизируйте данные и постройте модель на стандартизированных данных. Используйте для стандартизации класс StandardScaler. Оцените

качество модели с помощью classification\_report. Улучшилась ли модель?

7. Воспользуйтесь перекрестной проверкой, чтобы оценить качество моделей. Используйте функцию cross\_val\_score.

8. Если усреднённые показатели качества линейной модели для нестандартизованных и стандиртизованных данных почти равны, то следует для улучшения качества SVM-модели брать другое значение kernel. Найдите наилучший вариант. Для модели логистической регрессии организуйте поиск по сетке, используя

from sklearn.model\_selection import GridSearchCV.

1. Решите эту задачу с помощью метода k ближайших соседей, подобрав оптимальное значение k.
2. Постройте дерево решений с помощью класса DecisionTreeClassifier.
3. Оцените качество модели с помощью функции classification\_report.
4. Подберите оптимальное значение гиперпараметра max\_depth с помощью поиска по сетке (класс GridSearchCV).
5. Обучите модель с оптимальным max\_depth и оцените результат.
6. Сравните все модели и сделайте выводы.