

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИМЕНИ
М.В.ЛОМОНОСОВА»**

ЭКОНОМИЧЕСКИЙ ФАКУЛЬТЕТ

«УТВЕРЖДАЮ»

Декан экономического факультета МГУ
профессор. _____ А.А.Аузан

«» 2020 год

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ

Машинное обучение и анализ данных-1

Уровень высшего образования
Магистратура

Направление подготовки (специальность)
38.04.01 Экономика

Направленность (профиль) ОПОП
Анализ данных в экономике

Форма обучения
очная

Рабочая программа рассмотрена и одобрена
Учебно-методической комиссией экономического факультета
(протокол № _____, дата)

Москва 2020

На оборотной стороне титульного листа указывается:

Рабочая программа дисциплины разработана в соответствии с самостоятельно установленным МГУ образовательным стандартом (ОС МГУ) для реализуемых основных профессиональных образовательных программ высшего образования по направлению подготовки «_____» магистратуры

**ОС МГУ утвержден решением Ученого совета МГУ имени М.В.Ломоносова от _____20
_____года (протокол №__).**

Год (годы) приема на обучение: 2020 и последующие

1. Место и статус дисциплины в структуре основной профессиональной образовательной программы подготовки магистра

Статус дисциплины: *вариативная*

Триместр: 2

2. Входные требования для освоения дисциплины

Для успешного освоения данного курса требуются знания и навыки, полученные в следующих дисциплинах:

- теория вероятностей и математическая статистика;
- основы программирования.

3. Планируемые результаты обучения по дисциплине, соотнесенные с требуемыми компетенциями выпускников

Компетенции выпускников (коды)	Индикаторы (показатели) достижения компетенций	Планируемые результаты обучения по дисциплине (модулю), сопряженные с компетенциями
ОПК-5. Способен использовать современные информационные технологии и программные средства при решении профессиональных задач	ОПК-5.И-1. Применяет общие или специализированные пакеты прикладных программ, предназначенные для обработки, визуализации и анализа данных, в том числе эконометрического анализа и имитационного моделирования	ОПК-5.И-1.У-1. Умеет применять общие или специализированные пакеты прикладных программ (MS Excel, Eviews, Stata, SPSS, AnyLogic, Tableau и др.) или языки программирования (R, Python и др.), предназначенные для обработки, визуализации и анализа данных, в том числе эконометрического анализа и имитационного моделирования в, в соответствии со своими профессиональными задачами
ПК-8. Способен использовать различные инструментальные методы расчета и анализа социально-экономических показателей	ПК-8.И-2. Применяет современные инструментальные методы расчета и анализа социально-экономических показателей при решении практических и (или) исследовательских задач	ПК-8.И-2.3-1. Знает платформы анализа данных, подходящие для реализации выбранного метода расчета и анализа, в том числе платформы бизнес-аналитики, BI-системы, платформы визуализации данных. ПК-8.И-2.У-1. Умеет применять инструментальные методы расчета и анализа показателей, в том числе с использованием различных платформ анализа данных и с применением языков

		программирования R и Python. ПК-8.И-2.У-2. Умеет адаптировать существующие инструментальные методы расчета и анализа показателей под конкретную задачу, в том числе путем написания программного кода на языках R и Python.
МПК-1. Способен разрабатывать эконометрические модели и модели машинного обучения исследуемых экономических процессов и явлений, интерпретировать полученные результаты	МПК-1.И-1. Применяет продвинутый эконометрический инструментарий и методы машинного обучения для построения моделей экономических процессов и явлений. Интерпретирует результаты проведенного моделирования, делает выводы и дает рекомендации на его основе	МПК-1.И-1.У-1. Умеет разрабатывать эконометрические модели и модели, основанные на методах машинного обучения. МПК-1.И-1.У-2. Умеет интерпретировать результаты моделирования, как эконометрического, так и основанного на применении методов машинного обучения. МПК-1.И-1.У-3. Умеет делать выводы из результатов моделирования, как эконометрического, так и основанного на применении методов машинного обучения, и давать рекомендации на основе сделанных выводов.
МПК-4. Способен видеть логические связи в системе собранной, обработанной и проанализированной информации, и на основании этого разрабатывать рекомендации для лиц, принимающих решения на микро- и макро уровне, или бизнес-решения	МПК-4.И-1. Анализирует и систематизирует собранные данные, разрабатывает рекомендации для лиц, принимающих решения на основе собранных данных	МПК-4.И-1.У-1. Умеет строить логические взаимосвязи в системе собранной информации МПК-4.И-1.У-2. Умеет разрабатывать рекомендации для лиц, принимающих управленческие решения как сфере государственного управления на разных уровнях, так и в сфере бизнеса

4. Объем дисциплины по видам занятий

Объем дисциплины составляет 6 зачетных единицы: 216 академических часов, из которых 56 академических часов составляет аудиторная нагрузка, из них 28 академических часов — лекции и 28 академических часов — семинары, 52 академических часа — групповая

контактная работа, 0 академических часов — индивидуальная контактная работа, 108 академических часов составляет самостоятельная работа магистранта.

5. Формат обучения: используется электронная информационная среды экономического факультета МГУ имени М.В.Ломоносова «ON.ECON».

6. Содержание дисциплины, структурированное по темам (разделам) с указанием отведенного на них количества академических часов и виды учебных занятий

Название раздела/темы	Всего, часы	В том числе			
		Контактная работа с преподавателем			Самостоятельная работа магистранта, часы
		Лекции, часы	Семинары, часы	Групповая, часы	
Тема 1. Основные задачи машинного обучения и анализа данных		2	2	4	0
Тема 2: Основные библиотеки для работы с данными в Python		2	2	4	9
Тема 3. Линейные модели в задаче регрессии		2	2	4	9
Тема 4. Линейные модели в задаче классификации		2	2	4	9
Тема 5. Оценки качества моделей в задачах регрессии и классификации		2	2	4	9
Тема 6. Выбор модели. Кросс-валидация. Отбор признаков		2	2	4	9
Тема 7. Регуляризация. Преобразование признаков		2	2	4	9
Тема 8. Метод опорных векторов		2	2	4	9
Тема 9. Деревья решений		2	2	4	9
Тема 10. Ансамбли моделей		2	2	4	9
Тема 11. Введение в нейронные сети		4	4	4	9
Тема 12. Анализ временных рядов		4	4	8	18
Текущая аттестация: — домашние задания					

Всего	216	28	28	52	
-------	-----	----	----	----	--

Краткое содержание тем дисциплины

Тема 1. Основные задачи машинного обучения и анализа данных. Типы задач: обучение с учителем (регрессия, классификация), обучение без учителя (кластеризация, поиск аномалий, снижение размерностей), частичное обучение, обучение с подкреплением. Основные области: компьютерное зрение, обработка естественного языка, рекомендательные системы, анализ временных рядов, обучение ранжированию, построение выводов по данным. Типы данных. Библиотеки научных вычислений: NumPy, SciPy. Библиотека для работы с табличными данными Pandas. Библиотеки визуализации: Matplotlib, Seaborn. Среда интерактивных вычислений Jupyter Notebook: настройка и установка, основные принципы работы.

Тема 2: Регрессия и классификация. Математическая постановка задач регрессии и классификации. Метрические методы регрессии и классификации: метод ближайших соседей, взвешенный метод ближайших соседей. Расстояния в пространстве признаков. Генеративная модель: идеальный байесовский классификатор, наивный байесовский классификатор.

Тема 3. Линейные модели в задаче регрессии. Метод наименьших квадратов, метод максимального правдоподобия, свойства оценок параметров модели. Аналитическое решение, итерационные методы обучения. Обобщения: взвешенный МНК, локальная регрессия.

Тема 4. Линейные модели в задаче классификации. Логистическая регрессия для бинарной классификации: метод максимального правдоподобия, метод наименьших квадратов с итеративным пересчетом весов, градиентный спуск. Многоклассовая классификация: один против всех, Softmax.

Тема 5. Оценки качества моделей в задачах регрессии и классификации. Метрики в задаче регрессии: MAE, MSE, MAPE, R^2 . Метрики в задаче классификации: кросс-энтропия, precision, recall, F-мера, ROC-кривая, AUC ROC. Информационные метрики: AIC, BIC, SBC.

Тема 6. Выбор модели. Кросс-валидация. Отбор признаков. Обобщающая способность и ее оценка: отложенная выборка, кросс-валидация. Отбор признаков в линейных моделях: stepwise-регрессия.

Тема 7. Регуляризация. Преобразование признаков. Проблема переобучения. L_1 - и L_2 -регуляризация. Гребневая регрессия, LASSO-регрессия, регрессия наименьшего угла, ElasticNet. Методы обучения моделей с регуляризацией. Виды признаков: категориальные, вещественные.

Тема 8. Метод опорных векторов. Метод опорных векторов в задаче классификации. Ядерный переход. Метод опорных векторов для задачи регрессии.

Тема 9. Деревья решений. Деревья решений для задачи классификации. Алгоритмы построения деревьев: ID3, C4.5, CART. Стрижка дерева. Связь с линейными моделями.

Тема 10. Ансамбли моделей. Bootstrap-метод. Бэггинг. Стэкинг. Случайный лес. Дилемма смещения-дисперсии. Бустинг: Adaboost, градиентный бустинг.

Тема 11. Введение в нейронные сети. История нейронных сетей. Искусственный нейрон. Многослойная нейронная сеть в задачах регрессии и классификации. Дифференцирование функции, заданной графом. Метод обратного распространения ошибок. Методы обучения нейронных сетей, основанные на стохастическом градиентном спуске.

Тема 12. Анализ временных рядов. Временные ряды. Стационарность временных рядов. Модели авторегрессии–скользящего среднего (модели ARMA, ARIMA, SARIMA, SARIMAX). Гетероскедастичность временных рядов (модель GARCH). Временные ряды с трендом (модель Хольта–Уинтерса).

Фонд оценочных средств для оценивания результатов обучения по дисциплине

Шкала оценивания результатов (баллы) по дисциплине:

Результаты обучения по дисциплине	Виды оценочных средств
ОПК-5.И-1.У-1. Умеет применять общие или специализированные пакеты прикладных программ (MS Excel, Eviews, Stata, SPSS, AnyLogic, Tableau и др.) или языки программирования (R, Python и др.), предназначенные для обработки, визуализации и анализа данных, в том числе эконометрического анализа и имитационного моделирования в, в соответствии со своими профессиональными задачами	Домашние работы Выполнение проектных заданий
ПК-8.И-2.З-1. Знает платформы анализа данных, подходящие для реализации выбранного метода расчета и анализа, в том числе платформы бизнес-аналитики, BI-системы, платформы визуализации данных.	Домашние работы Выполнение проектных заданий
ПК-8.И-2.У-1. Умеет применять инструментальные методы расчета и анализа показателей, в том числе с использованием различных платформ анализа данных и с применением языков программирования R и Python.	Домашние работы Выполнение проектных заданий
ПК-8.И-2.У-2. Умеет адаптировать существующие инструментальные методы расчета и анализа показателей под конкретную задачу, в том числе путем написания программного кода на языках R и Python.	Домашние работы Выполнение проектных заданий
МПК-1.И-1.У-1. Умеет разрабатывать эконометрические модели и модели, основанные на методах машинного обучения.	Домашние работы Выполнение проектных заданий
МПК-1.И-1.У-2. Умеет интерпретировать результаты моделирования, как эконометрического, так и основанного на применении методов машинного обучения.	Домашние работы Выполнение проектных заданий

МПК-1.И-1.У-3. Умеет делать выводы из результатов моделирования, как эконометрического, так и основанного на применении методов машинного обучения, и давать рекомендации на основе сделанных выводов.	Домашние работы Выполнение проектных заданий
МПК-4.И-1.У-1. Умеет строить логические взаимосвязи в системе собранной информации.	Домашние работы Выполнение проектных заданий
МПК-4.И-1.У-2. Умеет разрабатывать рекомендации для лиц, принимающих управленческие решения как в сфере государственного управления на разных уровнях, так и в сфере бизнеса	Домашние работы Выполнение проектных заданий

Виды оценочных средств	Баллы
Домашние работы	200
Проектные задания	100

Оценка по дисциплине выставляется, исходя из следующих критериев:

Оценка	Минимальное количество баллов	Максимальное количество баллов
<i>Отлично</i>	255	300
<i>Хорошо</i>	195	254,9
<i>Удовлетворительно</i>	120	194,9
<i>Неудовлетворительно</i>	60	119,9

Примечание: в случае, если магистрант за триместр набирает менее 20% баллов от максимального количества по дисциплине, то уже на промежуточном контроле (и далее на пересдачах) действует следующее правило сдачи: «магистрант может получить только оценку «Удовлетворительно», и только если получит за промежуточный контроль, включающий весь материал дисциплины, не менее, чем 85% от баллов за промежуточный контроль».

Типовые задания, методические рекомендации по их подготовке и требования к их выполнению:

Домашние работы представляют собой практические задания, ориентированные на закрепление навыков машинного обучения и анализа данных на языке Python в рамках различных тем. Задание сдается в формате Jupyter-ноутбука .ipynb с кодом, комментариями и ответами на вопросы задания. В комментариях следует описать используемые для решения задачи методы и данные, подробно обосновать выбор алгоритма, представить результаты

расчётов (при необходимости используя таблицы и рисунки). Домашние задания позволяют набрать до 200 баллов.

Проектные работы представляют собой практические задания, требующие от студента применения полученных навыков машинного обучения для решения проблем. Задание сдается в формате устной презентации, а также исходного кода на языке Python в удобном формате (например, Jupyter-ноутбук, код на репозитории Github, zip-архив). Проект может быть как индивидуально, так и в команде. В ходе защиты проекта требуется кратко описать решаемую задачу, описать применяемые методы машинного обучения и анализа данных, сформулировать и объяснить полученные результаты, ответить на вопросы принимающих. Проектные работы позволяют набрать до 100 баллов.

Ниже приведены примеры заданий, которые будут задаваться студентам в процессе обучения.

Пример 1: Реализация линейной регрессии для предсказания цен на недвижимость.

Обучим нашу модель предсказывать среднюю цену квартиры по среднему числу комнат в доме.

```
In [5]: predictors = ['rm']

# формируем данные для модели
# метод .values возвращает данные в виде np.array
X = boston[predictors].values
y = boston['medv'].values

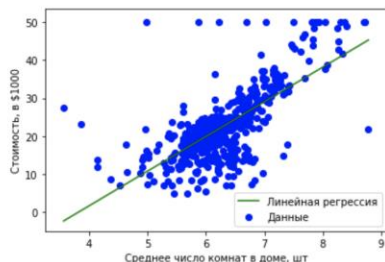
model_lm1 = LinearRegression()
model_lm1.fit(X, y)
y_pred = model_lm1.predict(X)
```

Теперь посмотрим как выглядит предсказание обученной модели.

```
In [6]: # сделаем сетку от минимального до максимального значения среднего числа комнат
X_low, X_high = X.min(), X.max()
X_plt = np.linspace(X_low, X_high, 64)
y_pred_plt = model_lm1.predict(X_plt.reshape(-1, 1))

# график предсказаний и диаграмма рассеяния
plt.plot(X_plt, y_pred_plt, color='green', label='Линейная регрессия')
plt.scatter(X[:, 0], y, color='blue', label='Данные')

plt.xlabel('Среднее число комнат в доме, шт')
plt.ylabel('Стоимость, в $1000')
plt.legend()
plt.show()
```



Пример 2: Проблема переобучения в задаче полиномиальной регрессии (с решением).

Задание №3 (2 балла): Определить функцию, которая строит матрицу признаков:

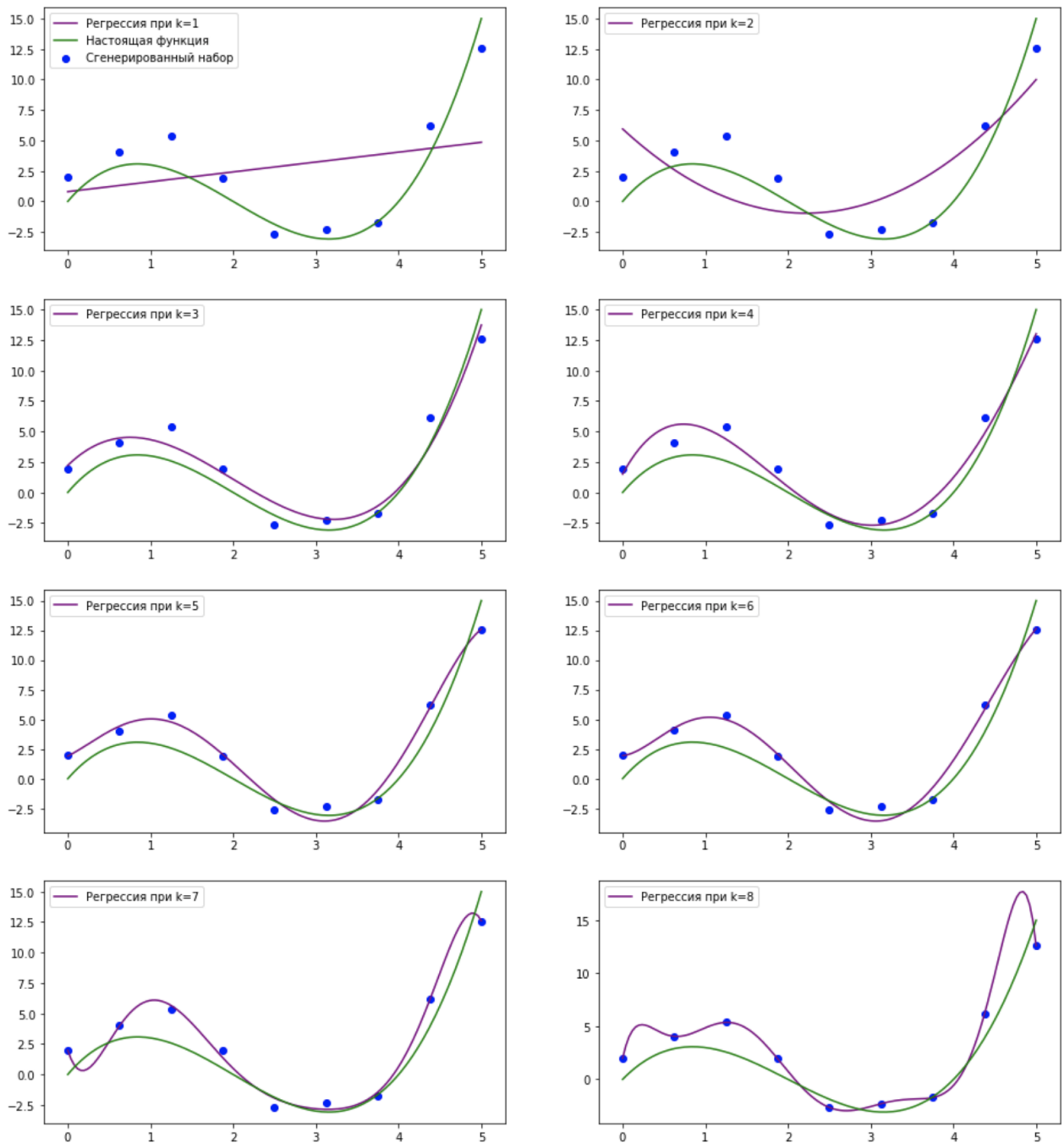
```
In [20]: # на вход передается вектор x размера N
# на выходе выдать матрицу Xp размера Nxk,
# где i-ая строка соответствует степеням x_i
# с 1 до k включительно (нулевая степень будет добавлена в методе fit)
def get_polynomial_features(X, k):
    ### МЕСТО ДЛЯ ВАШЕГО КОДА ###

    # РЕШЕНИЕ
    X = np.array(X)
    return np.column_stack([X ** i for i in range(1, k+1)])
    # КОНЕЦ РЕШЕНИЯ
```

Обучим модель полиномиальной регрессии и посмотрим на график предсказаний для различных k .

```
In [21]: # настоящая зависимость
x_plt = np.linspace(0, 5, 128)
y_plt = func(x_plt)

_, axes = plt.subplots(4, 2, figsize=(16, 18))
for k, ax in zip(range(1, 9), axes.flatten()):
    mdl = LinearRegression()
    Xp = get_polynomial_features(X, k)
    mdl.fit(Xp, y)
    yp_plt = mdl.predict(get_polynomial_features(x_plt, k))
    ax.plot(x_plt, yp_plt, color='purple', label=f'Регрессия при k={k}')
    ax.plot(x_plt, y_plt, color='green', label='Настоящая функция' if k == 1 else None)
    ax.scatter(X, y, color='blue', label='Сгенерированный набор' if k == 1 else None)
    ax.legend()
plt.show()
```



Пример 3: реализация алгоритма гребневой регрессии на языке Python (с решением).

- **L2-регуляризация (Ridge-регрессия или гребневая регрессия).** К функции потерь добавляется L_2 норма весов θ :

$$\theta^{ridge} = \arg \min_{\theta \in \Theta} \left\{ \sum_{k=1}^n S(y_k, f_{\theta}(x_k)) + \alpha \sum_{i=1}^d \theta_i^2 \right\},$$

где $\alpha > 0$ -- гиперпараметр. Аналитическое решение задачи оптимизации в случае L2-регуляризации:

$$\theta^{ridge} = (X^T X + \alpha I)^{-1} X^T y.$$

Задание №4 (12 баллов): применить гребневую регрессию на наборе данных по недвижимости в Бостоне и сравнить результаты с обычной линейной регрессией.

Требуется:

- Реализовать метод **fit** настройки параметров (6 баллов).
- Реализовать метод **predict**, который предсказывает целевую переменную (4 балла).
- Сравнить метрики гребневой регрессии и обычной линейной регрессии (2 балла).

Замечание: Для того, чтобы гребневая регрессия работала корректно, необходимо привести все признаки к единым величинам. Для этого для каждого признака x из него вычитают выборочное среднее и делят на корень из выборочной дисперсии:

$$\tilde{x} = \frac{x - \text{mean}(x)}{\text{std}(x)}, \quad \text{mean}(x) = \frac{1}{n} \sum_{k=1}^n x_k, \quad \text{std}(x) = \sqrt{\frac{1}{n} \sum_{k=1}^n (x - \text{mean}(x))^2}$$

```
In [22]: # Реализация гребневой регрессии линейной регрессии
class RidgeRegression:
    # настройка модели
    def __init__(self, alpha):
        self.alpha = alpha
        self._is_fitted = False
        return

    # метод fit
    # на вход принимает матрицу X размера Nxd с объясняющими признаками
    # и вектор y размера N с предсказанными значениями целевой переменной
    # настраивает веса модели
    # возвращает сам себя (для совместимости с библиотекой sklearn)
    def fit(self, X, y):
        # добавляем фиктивный признак
        X = np.array(X)
        y = np.array(y)

        # выполним нормализацию признака
        # важно сделать это ДО добавления фиктивного признака
        # иначе он превратится в 0 после нормализации
        self._mean = X.mean(axis=0, keepdims=True)
        self._std = X.std(axis=0, keepdims=True)

        X = (X - self._mean) / self._std

        # дописываем фиктивный признак
        X = np.column_stack([np.ones(len(X)), X])

        # настроить параметры theta
        # self._theta = ### МЕСТО ДЛЯ ВАШЕГО КОДА ###

        # РЕШЕНИЕ
        d = X.shape[1]
        I = np.eye(d)
        self._theta = np.linalg.inv(X.T @ X + self.alpha * I) @ X.T @ y
        # КОНЕЦ РЕШЕНИЯ

        self._is_fitted = True
        return self

    # на вход принимает матрицу X размера Nxd с объясняющими признаками
    # на выход дает вектор y размера N с предсказанными значениями целевой переменной
    # примечание: не забудьте нормализовать признаки
    def predict(self, X):
        ### МЕСТО ДЛЯ ВАШЕГО КОДА ###

        # РЕШЕНИЕ
        X = (np.array(X) - self._mean) / self._std

        X = np.column_stack([np.ones(len(X)), X])
        return X @ self._theta
        # КОНЕЦ РЕШЕНИЯ
```

8. Ресурсное обеспечение

8.1. Перечень основной и дополнительной литературы

Основная литература:

- Christopher M. Bishop. Pattern Recognition and Machine Learning

- T.Hastie, R. Tibshirani, J. Friedman. Elements of statistical learning
- Kevin P. Murphy. Machine Learning: A Probabilistic Perspective
- Петер Флах. Машинное обучение. Наука и искусство построения алгоритмов, которые извлекают знания из данных
- В.Н.Варник. Восстановление зависимостей по эмпирическим данным.
- В.Н.Варник, А.Я.Червоненкис. Теория распознавания образов.
- Vladimir N. Vapnik. The Nature of Statistical Learning Theory.

8.2.Перечень лицензионного программного обеспечения

- Среда разработки Anaconda для языка Python.
- Библиотеки анализа данных для языка Python.

8.3.Перечень профессиональных баз данных и информационных справочных систем

- Школа Анализа Данных Яндекса
- Академия MADE
- Coursera

8.4.Перечень ресурсов информационно-телекоммуникационной сети «Интернет»

- [Курс CS229 Stanford](#)
- [Видеолекции курса CS299 Stanford](#)
- [Towards Data Science](#)
- [Kaggle](#)
- [Блог А.Г. Дьяконова](#)
- [Курс Open Data Science](#)

8.5. Описание материально-технической базы

Для организации занятий по дисциплине необходимы следующие технические средства обучения: компьютерный класс с проектором.

Для организации дистанционных занятий по дисциплине необходимы следующие технические средства: компьютер с доступом в интернет, камера и микрофон, аккаунт Zoom и установленная среда разработки Anaconda у каждого студента.

- 1. Язык преподавания:** русский.
- 2. Преподаватель (преподаватели):** аспирант Р.А. Габдуллин, аспирант В.А. Макаренко
- 3. Автор (авторы) программы:** аспирант Р.А. Габдуллин, аспирант В.А. Макаренко