



МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное образовательное учреждение
высшего образования
«Дальневосточный федеральный университет»
(ДВФУ)

ШКОЛА ЦИФРОВОЙ ЭКОНОМИКИ

СОГЛАСОВАНО
Руководитель ОП

Р.И. Дремлюга

«17» июня 2019 г.

УТВЕРЖДАЮ

Директор Школы цифровой

экономики



И.Г. Мирин

2019 г.

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ

«МАШИННОЕ ОБУЧЕНИЕ»

направления 09.04.01 Информатика и вычислительная техника

Магистерская программа «Кибербезопасность»

Форма подготовки очная

Школа цифровой экономики

курс 1 семестр 1,2

лекции 72 час.

практические занятия 72 час.

лабораторные работы 0 час.

всего часов аудиторной нагрузки 144 час.

самостоятельная работа 90 час.

контрольные работы программой не предусмотрены

курсовая работа/проект – не предусмотрено

зачет – не предусмотрен

экзамен – 1,2 семестр

Рабочая программа составлена в соответствии с требованиями Федерального государственного образовательного стандарта высшего образования по направлению подготовки/специальности 09.04.01 Информатика и вычислительная техника, утвержденного приказом Министерства образования и науки Российской Федерации от 19.09.2017 г. № 918.

Рассмотрена и утверждена на заседании Дирекции Школы цифровой экономики «17» июня 2019 года (протокол № 124-01-07-05).

Составитель: с.пр. Кленин А.С.

Оборотная сторона титульного листа РПД

I. Рабочая программа пересмотрена на заседании Дирекции Школы цифровой экономики:

Протокол от «_____» _____ 20__ г. № _____

Заместитель директора ШЦЭ

по учебной и воспитательной работе _____
(подпись) (И.О. Фамилия)

II. Рабочая программа пересмотрена на заседании Дирекции Школы цифровой экономики:

Протокол от «_____» _____ 20__ г. № _____

Заместитель директора ШЦЭ

по учебной и воспитательной работе _____
(подпись) (И.О. Фамилия)

АННОТАЦИЯ

Б1.О.02.01 МАШИННОЕ ОБУЧЕНИЕ (MACHINE LEARNING)

Рабочая программа учебной дисциплины «Машинное обучение» предназначена для студентов, обучающихся по направлению подготовки 09.04.01 Информатика и вычислительная техника (уровень магистратуры), профиль «Кибербезопасность».

Дисциплина «Машинное обучение» входит в обязательную часть блока «Дисциплины (модули) Б.1» (Б1.О.02.01) учебного плана подготовки магистров.

Общая трудоемкость освоения дисциплины составляет 8 зачетных единиц, 288 часов. Дисциплина реализуется на 1 курсе в 1 и 2 семестре.

Семестр	Аудиторные Занятия		Самостоя- тельная работа	Контроль	Форма контроля	Всего по дисциплине	
	Лекции	Практи- ческие занятия				Часы	Зачетные единицы
1 семестр	36	36	45	27	Экзамен	144	4
2 семестр	36	36	45	27	Экзамен	144	4
Всего	72	72	90	54		288	8

Цель – изучение основных разделов теории машинного обучения (Machine Learning) и овладение навыками практического решения задач интеллектуального анализа данных - майнинга данных (Data Mining).

Задачи:

- Изучить основные инструменты математического анализа, линейной алгебры, методов оптимизации и теории вероятностей;
- Получить базовые навыки программирования на языках C++ и Python применительно к работе с большими объемами данных;
- Изучить основные модели машинного обучения и методики оценки их качества;
- Изучить основные способы организации искусственных нейронных сетей;

- Овладеть методологией управления data-science проектами;
- Научиться строить модели машинного обучения для решения профессиональных задач.

В результате освоения дисциплины обучающийся должен:

Знать:

- современное состояние исследований в области машинного обучения;
- принципы построения систем машинного обучения;
- модели представления и описания технологий машинного обучения.

Уметь:

- проводить анализ предметной области;
- определять назначение, выбирать методы и средства для построения систем машинного обучения;
- строить системы машинного обучения.

Иметь навыки и (или) опыт деятельности (владеть):

- использования аппарата простейшего анализ данных;
- применения методов классификации информации;
- реализации алгоритмов машинного обучения.

Связь курса с другими дисциплинами

Для успешного изучения дисциплины «Машинное обучение» необходимы знания базовой программы курса «Высшая математика» и основ программирования (желательно Python).

В результате данной дисциплины у обучающихся формируются следующие общепрофессиональные компетенции (элементы компетенций):

Код и формулировка компетенции	Этапы формирования компетенции
ОПК-1. Способен самостоятельно приобретать, развивать и применять математические, естественнонаучные, социально-экономические и профессиональные знания для решения	<p>ОПК-1.1 Знать: математические, естественнонаучные и социально-экономические методы для использования в профессиональной деятельности</p> <p>ОПК-1.2 Уметь: решать нестандартные профессиональные задачи, в том числе в новой или незнакомой среде и в междисциплинарном контексте, с применением математических, естественнонаучных, социально-</p>

<p>нестандартных задач, в том числе в новой или незнакомой среде и в междисциплинарном контексте</p>	<p>экономических и профессиональных знаний ОПК-1.3 Владеть: навыками теоретического и экспериментального исследования объектов профессиональной деятельности, в том числе в новой или незнакомой среде и в междисциплинарном контексте</p>
<p>ОПК-2. Способен разрабатывать оригинальные алгоритмы и программные средства, в том числе с использованием современных интеллектуальных технологий, для решения профессиональных задач</p>	<p>ОПК-2.1 Знать: современные информационно-коммуникационные и интеллектуальные технологии, инструментальные среды, программно-технические платформы для решения профессиональных задач ОПК-2.2 Уметь: обосновывать выбор современных информационно-коммуникационных и интеллектуальных технологий, разрабатывать оригинальные программные средства для решения профессиональных задач ОПК-2.3 Владеть: навыками разработки оригинальных программных средств, в том числе с использованием современных информационно-коммуникационных и интеллектуальных технологий, для решения профессиональных задач</p>

Для формирования вышеуказанных компетенций в рамках дисциплины «Машинное обучение» применяется следующий метод интерактивного обучения: метод автоматизированного обучения в системе автоматического тестирования программ CATS, предъявляющей задания и позволяющей оценить решение.

При выполнении различных видов работ используются следующие технологии:

1. *Проблемное обучение* – стимулирование обучающихся к самостоятельному приобретению знаний, необходимых для решения конкретной проблемы.

2. *Контекстное обучение* – мотивация магистрантов к усвоению знаний путём выявления связей между конкретным знанием и его применением.

3. *Обучение на основе опыта* – активизация познавательной деятельности магистрантов за счёт ассоциации и собственного опыта с предметом обучения.

I. СТРУКТУРА И СОДЕРЖАНИЕ ТЕОРЕТИЧЕСКОЙ ЧАСТИ КУРСА

Тема 1. Введение. Основы машинного обучения

Задача обучения с учителем и без учителя.

Классификация и регрессия.

Линейные модели.

Обработка данных.

Кросс-валидация.

Подбор гиперпараметров.

Визуализация данных.

Тема 2. Методы оптимизации. Градиентный спуск

Задача регрессии, классификации.

Функция потерь. Оптимизация. Перебор по сетке.

Производная, частные производные, градиент.

Градиентный спуск, проблема выбора шага.

Стохастический градиентный спуск. Использование момента.

Adagrad, Adadelta, Adam.

RMSProp*.

Тема 3 Линейная регрессия

Постановка задачи линейной регрессии.

Метод наименьших квадратов.

Ковариация, корреляция.

Критерий R^2 .

Анализ остатков.

Тема 4. Глобальная оптимизация. Генетический алгоритм

Многопараметрическая оптимизация.

Доминанция и оптимальность по Парето.

Функция качества (fitness). Аппроксимация качества.

Общая идея генетического алгоритма.

Представление генома.

Методы селекции: пропорционально качеству, универсальная выборка (stochastic universal sampling), с наследием (reward-based), турнир. Стратегия элитизма.

Методы кроссовера. Двух и многоточечный, равномерный (по подмножествам), для перестановок.

Мутация. Влияние на скорость обучения.

Управление популяцией. Сегрегация, старение, распараллеливание.

Генетическое программирование.

Тема 5. Метод ближайших соседей (k-NN)

Понятие и свойства метрики. Ослабление требования к неравенству треугольника.

Базовый алгоритм классификации методом 1-NN и k-NN. Преимущества и недостатки.

Метрики L1, L2, Хемминга, Левенштейна, косинусное расстояние.

Потеря точности нормы в высоких размерностях.

Нормализация координат. Предварительная трансформация пространства признаков.

Метрика Махаланобиса.

Кросс-валидация методом "без одного" (leave one out).

Определение границ, показатель пограничности.

Сжатие по данным. Понятия выброса, прототипа, усвоенной точки.

Алгоритм Харта (Hart).

Регрессия методом k-NN.

Взвешенные соседи.

Связь с градиентным спуском. Стохастическая формулировка, softmax.

Метод соседних компонент (neighbour component analysis)*.

Связь с выпуклой оптимизацией. Метод большого запаса (Large margin NN)

Оптимизация классификатора, k-d деревья

Хеши чувствительные к локальности, хеши сохраняющие локальность*.

Тема 6. Наивный байесов классификатор

Условная вероятность. Байесово решающее правило. Обновление вероятностей.

Наивный классификатор, предположение о независимости признаков.

Оценка плотности распределения для числовых признаков.

Алгоритмические оптимизации.

Алгоритм EM.

Тема 7. Логистическая регрессия

Сигмоид

Метод наибольшего правдоподобия

Логистическая регрессия для меток $-1, 1$

Тема 8. Деревья решений

Понятие дерева решений.

Борьба с оверфиттингом: bagging, выборки признаков.

Ансамбли, случайный лес (Random Forest).

Понятие энтропии, определение информации по Шеннону.

Метрики: примеси Джини (Gini impurity), добавленная информация (information gain).

Деревья регрессии. Метрика вариации.

Непрерывные признаки. Использование главных компонент вместо признаков.

Сокращение дерева (pruning).

Тема 9. Кластеризация

Задача обучения без учителя, применения при эксплораторном анализе

Неметрическая кластеризация: функция схожести, компоненты связности и остовные деревья, иерархическая кластеризация снизу вверх

Метрики, понятие центроида и представителя класса

Центроидные алгоритмы: k-means, k-medoid

Алгоритмы, основанные на плотности: DBSCAN, OPTICS

Алгоритмы, основанные на распределении: сумма гауссиан

Нечёткая кластеризация, алгоритм c-means

Метрики качества: leave-one-out, силуэт, индекс Дэвиса-Болдина (Davies-Bouldin), индекс Данна (Dunn)

II. СТРУКТУРА И СОДЕРЖАНИЕ ПРАКТИЧЕСКОЙ ЧАСТИ КУРСА

Лабораторные работы (54 часа)

Лабораторная работа №1. Работа с текстом (8 часов)

Задачи обработки текста: извлечение, поиск, классификация (тематическая, эмоциональная), перевод. Разбиение на слова, пунктуация, лексический и морфологический анализ. Определение частей речи, имён, основ слов.

Частотный анализ, представление bag-of-words, TF-IDF и его варианты. N-граммы, byte-pair encoding. Векторные представления, семантическая интерпретация алгебраических операций. Унитарный код (One-hot encoding). Алгоритмы Word2Vec и FastText. Алгоритм GloVe*.

Лабораторная работа №2. Снижение размерности (10 часов)

Постановка задачи, причины и цели снижения размерности. Выбор и извлечение признаков. Подходы к выбору признаков: filtering, wrapping, embedding. Расстояние между распределениями. Расстояние Кульбака-Лейблера. Взаимная информация.

Алгоритмы выбора признаков: на основе корреляции (CFS), взаимной информации, Relief. Метод главных компонент (PCA). Нелинейные обобщения метода главных компонент. Kernel PCA. Неотрицательное матричное разложение (NMF). Стохастическое вложение соседей с t-распределением (t-SNE).

Лабораторная работа №3. Метод опорных векторов (8 часов)

Постановка задачи линейного SVM для линейно разделимой выборки. Задача оптимизации с ограничениями. Двойственная задача Лагранжа. Условия Каруша-Куна-Такера. Функция Лагранжа для линейного SVM.

Опорный вектор. Типы опорных векторов. Kernel trick. Полиномиальное ядро. Радиально-базисное ядро (RBF). SVM для задачи регрессии.

Лабораторная работа №4. Работа с изображениями (10 часов)

Сверхточные фильтры, непрерывное и дискретное определение свёртки. Сглаживающие фильтры. Фильтр Гаусса. Дифференцирующие фильтры: Roberts cross, Sobel, Prewitt, Scharr.

Лабораторная работа №5. Поиск границ (8 часов)

Алгоритм Кенни (Canny). Адаптивное сглаживание. Определение порога методом Отцу (Otsu).

Лабораторная работа №6. Оптимизация (10 часов)

Оптимизация с учётом направления градиента. Преобразование Nough. Обобщения на многопараметрический и многомерный случай. Извлечение признаков. Признаки Хаара (Haar).

Практические задания (18 часов)

Практическое задание №1 (4 часа)

Многопараметрическая оптимизация.

Практическое задание №2 (6 часов)

Управление популяцией. Сегрегация, старение, распараллеливание.

Практическое задание №3 (4 часа)

Оценка плотности распределения для числовых признаков.

Практическое задание №3 (4 часа)

Нормализация координат. Предварительная трансформация пространства признаков.

III. УЧЕБНО-МЕТОДИЧЕСКОЕ ОБЕСПЕЧЕНИЕ

САМОСТОЯТЕЛЬНОЙ РАБОТЫ ОБУЧАЮЩИХСЯ

Учебно-методическое обеспечение самостоятельной работы обучающихся по дисциплине «Машинное обучение» представлено в Приложении 1 и включает в себя:

план-график выполнения самостоятельной работы по дисциплине, в том числе примерные нормы времени на выполнение по каждому заданию;

характеристика заданий для самостоятельной работы обучающихся и методические рекомендации по их выполнению;

требования к представлению и оформлению результатов самостоятельной работы;

критерии оценки выполнения самостоятельной работы.

IV. КОНТРОЛЬ ДОСТИЖЕНИЯ ЦЕЛЕЙ КУРСА

№ п/п	Контролируемые разделы / темы дисциплины	Коды и этапы формирования компетенций		Оценочные средства	
				текущий контроль	промежуточная аттестация
1	Основы машинного обучения	ОПК-1 ОПК-2	знает теорию	ПР-2 ПР-11 ТС	УО
			умеет применять на практике		
			владеет навыками использования		
2	Методы оптимизации. Градиентный спуск	ОПК-1 ОПК-2	знает теорию	ПР-2 ПР-11 ТС	УО-1
			умеет применять на практике		
			владеет навыками использования		
3	Линейная регрессия Глобальная оптимизация.	ОПК-1 ОПК-2	знает теорию	ПР-2 ПР-11 ТС	УО-1
			умеет применять на практике		
			владеет навыками использования		
4	Генетический алгоритм	ОПК-1 ОПК-2	знает теорию	ПР-4 ПР-11	УО-2

			умеет применять на практике	ТС	
			владеет навыками использования		
5	Метод ближайших соседей	ОПК-1 ОПК-2	знает теорию	ПР-2 ПР-11 ТС	УО-1
			умеет применять на практике		
			владеет навыками использования		
6	Наивный байесов классификатор	ОПК-1 ОПК-2	знает теорию	ПР-1 ПР-11 ТС	УО-1
			умеет применять на практике		
			владеет навыками использования		
7	Логистическая регрессия	ОПК-1 ОПК-2	знает теорию	ПР-2 ПР-11 ТС	УО-1
			умеет применять на практике		
			владеет навыками использования		
8	Деревья решений	ОПК-1 ОПК-2	знает теорию	ПР ПР-11 ТС	УО-2
			умеет применять на практике		
			владеет навыками использования		
9	Кластеризация	ОПК-1 ОПК-2	знает теорию	ПР-2 ПР-11 ТС	УО-1
			умеет применять на		

			практике		
			владеет навыками использования		

1. устный опрос (УО): собеседование (УО-1), коллоквиум (УО-2); итоговая презентация (УО-3); круглый стол (УО-4);
2. технические средства контроля (ТС);
3. письменные работы (ПР): тесты (ПР-1), контрольные работы (ПР-2), эссе (ПР-3), рефераты (ПР-4), курсовые работы (ПР-5), научно-учебные отчеты по практикам (ПР-6), конспект (ПР-7), проект (ПР-9). Разноуровневые задачи и задания (ПР-11) и т.п.

Типовые индивидуальные задания, методические материалы, определяющие процедуры оценивания знаний, умений и навыков и (или) опыта деятельности, а также критерии и показатели, необходимые для оценки знаний, умений, навыков и характеризующие этапы формирования компетенций в процессе освоения образовательной программы, представлены в Приложении 2.

V. СПИСОК УЧЕБНОЙ ЛИТЕРАТУРЫ И ИНФОРМАЦИОННО-МЕТОДИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ

Основная литература (электронные и печатные издания)

1. Коэльо, Л.П. Построение систем машинного обучения на языке Python [Электронный ресурс] / Л.П. Коэльо, В. Ричарт; пер. с англ. Слинкин А. А.. — Электрон. дан. — Москва: ДМК Пресс, 2016. — 302 с. — Режим доступа: <https://e.lanbook.com/book/82818>. — Загл. с экрана.
2. Неделько В.М. Основы статистических методов машинного обучения [Электронный ресурс]: учебное пособие/ Неделько В.М.— Электрон. текстовые данные. — Новосибирск: Новосибирский государственный технический университет, 2010. — 72 с.— Режим доступа: <http://www.iprbookshop.ru/45418.html>. — ЭБС «IPRbooks»
3. Флах, П. Машинное обучение. Наука и искусство построения алгоритмов, которые извлекают знания из данных [Электронный ресурс] / П. Флах. — Электрон. дан. — Москва: ДМК Пресс, 2015. — 400 с. — Режим доступа: <https://e.lanbook.com/book/69955>. — Загл. с экрана.
4. Рашка, С. Python и машинное обучение: крайне необходимое пособие по новейшей предсказательной аналитике, обязательное для более глубокого понимания методологии машинного обучения [Электронный ресурс]: руководство / С. Рашка; пер. с англ. Логунова А.В. — Электрон. дан. — Москва: ДМК Пресс, 2017. — 418 с. — Режим доступа: <https://e.lanbook.com/book/100905>. — Загл. с экрана.
5. Шарден, Б. Крупномасштабное машинное обучение вместе с Python [Электронный ресурс]: учебное пособие / Б. Шарден, Л. Массарон, А. Боскетти; пер. с англ. А. В. Логунова. — Электрон. дан. — Москва: ДМК Пресс, 2018. — 358 с. — Режим доступа: <https://e.lanbook.com/book/105836>. — Загл. с экрана.

6. Кук, Д. Машинное обучение с использованием библиотеки H2O [Электронный ресурс] / Д. Кук; пер. с англ. Огурцова А.Б.. — Электрон. дан. — Москва: ДМК Пресс, 2018. — 250 с. — Режим доступа: <https://e.lanbook.com/book/97353>. — Загл. с экрана.

Дополнительная литература
(печатные и электронные издания)

1. Информационные аналитические системы [Электронный ресурс]: учебник / Т. В. Алексеева, Ю. В. Амириди, В. В. Дик и др.; под ред. В. В. Дика. - М.: МФПУ Синергия, 2013. - 384 с. - (Университетская серия). - ISBN 978-5-4257-0092-6,
<http://www.znaniyum.com/bookread.php?book=451186>
2. Домингос, П. Верховный алгоритм: как машинное обучение изменит наш мир [Электронный ресурс] Москва: Манн, Иванов и Фербер, 2016. 336 с. <https://e.lanbook.com/book/91645>.
3. Гаврилова, И.В. Основы искусственного интеллекта [Электронный ресурс]: учеб. пособие / И.В. Гаврилова, О.Е. Масленникова. Москва: ФЛИНТА, 2013. 282 с. <https://e.lanbook.com/book/44749>. 4. Ясницкий, Л.Н. Интеллектуальные системы [Электронный ресурс]: учеб. пособ./ Москва : Издательство 'Лаборатория знаний', 2016. 224 с. Режим доступа: <https://e.lanbook.com/book/90254>.

Перечень ресурсов
информационно-телекоммуникационной сети «Интернет»

1. Байесовские_методы_машинного_обучения_(курс_лекций)_/_2017 Д.П. Ветров - [http://www.machinelearning.ru/wiki/index.php?title=Байесовские_методы_машинного_обучения_\(курс_лекций\)_/_2017_Д.П.Ветров](http://www.machinelearning.ru/wiki/index.php?title=Байесовские_методы_машинного_обучения_(курс_лекций)_/_2017_Д.П.Ветров)
2. Машинное обучение (курс лекций, Н.Ю. Золотых) - <http://www.uic.unn.ru/~zny/ml/>

3. Машинное_ обучение_(курс_ лекций С.К.Воронцов). - [http://www.machinelearning.ru/wiki/index.php?title=Машинное_обучение_\(курс_лекций%2С_К.В.Воронцов\)](http://www.machinelearning.ru/wiki/index.php?title=Машинное_обучение_(курс_лекций%2С_К.В.Воронцов))
4. [Курс «Введение в машинное обучение», К.В.Воронцов \(ВШЭ и Яндекс\).Хабр об этом курсе.](#)
5. [Специализация «Машинное обучение и анализ данных» \(МФТИ и Яндекс\). Хабр об этом курсе.](#)
6. [Машинное обучение \(семинары,ФУПМ МФТИ\)](#)
7. [Машинное обучение \(семинары, ВМК МГУ\)](#)
8. [Машинное обучение \(курс лекций, Н.Ю.Золотых\)](#)
9. [Машинное обучение \(курс лекций, СГАУ, С.Лисицын\)](#)

VI. МЕТОДИЧЕСКИЕ УКАЗАНИЯ ПО ОСВОЕНИЮ ДИСЦИПЛИНЫ

На изучение дисциплины отводится 90 часов аудиторных занятий. Формы работы: лекции, самостоятельная работа с учебной и научной литературой, самостоятельное выполнение индивидуальных заданий, консультации. На занятиях перед выдачей индивидуальных заданий преподаватель объясняет теоретический материал по заданной теме. Вводит основные требования к его выполнению. Приводит примеры.

По ряду тем студентам предлагается работать самостоятельно, выполняя полный обзор по теме. Преподаватель контролирует работу студентов, отвечает на возникающие вопросы, предоставляет список литературных источников для освоения темы, а также перечень вопросов для самопроверки.

После выполнения задания, студент оформляет материал в форме программного кода и отправляет его на проверку преподавателю по электронной почте, либо предъявляет на компьютере во время занятия. Студент отвечает устно во время занятия по заданной теме.

Рекомендации по подготовке к экзамену

Рекомендуется регулярное посещение всех учебных занятий в течение всего семестра: лекций, консультаций и т.п., а также активное изучение

рекомендованной литературы, и выполнение в установленные сроки всех индивидуальных заданий.

При ответе на каждый вопрос зачета/экзамена студент должен продемонстрировать знание определения указанного понятия, связанных с ним особенностей реализации и применения, умение реализовать указанную операцию, а также навыки иллюстрации теоретических принципов на предложенных простых примерах.

VII. МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ

Компьютерный класс: Проектор DLP, 3000 ANSI Lm, WXGA 1280x800, 2000:1 EW330U Mitsubishi; Моноблок HP ProOne 440 G3 23.8" All-in-One, диагональ экрана 23.8", разрешение экрана 1920x1080, Bluetooth, Wi-Fi, операционная система: Windows 10 Enterprise, оптический привод DVD, процессор: Intel Core i5-7500T, размер оперативной памяти: 8 ГБ, видеопроцессор: Intel HD Graphics 630, объем жесткого диска: 1Тб. Беспроводные ЛВС для обучающихся обеспечены системой на базе точек доступа 802.11a/b/g/n 2x2 MIMO(2SS). Специализированное ПО: Matlab, Simulink, Visual Studio 2019, Система автоматического тестирования CATS/	690922, Приморский край, г. Владивосток, о. Русский, п. Аякс, 10, г. Владивосток, о. Русский, п. Аякс , корпус G, ауд. G468
---	--



МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное образовательное учреждение
высшего образования
«Дальневосточный федеральный университет»
(ДВФУ)

ШКОЛА ЦИФРОВОЙ ЭКОНОМИКИ

**УЧЕБНО-МЕТОДИЧЕСКОЕ ОБЕСПЕЧЕНИЕ
САМОСТОЯТЕЛЬНОЙ РАБОТЫ ОБУЧАЮЩИХСЯ**

по дисциплине «Машинное обучение»

Направление подготовки 09.04.01 «Информатика и вычислительная техника»

Форма подготовки очная

Владивосток

2019

План-график выполнения самостоятельной работы по дисциплине

№ п/п, название	Дата/сроки выполнения	Вид СРС	Примерные нормы времени на выполнение	Форма контроля
1. Работа с текстом	3 неделя 2 семестра	ИДЗ	3 недели	Проверка программы
2. Снижение размерности	9 неделя 2 семестра	ИДЗ	3 недели	Проверка программы
3. Метод опорных векторов	3 неделя 2 семестра	ИДЗ	3 недели	Проверка программы
4. Работа с изображениями	2 неделя 3 семестра	ИДЗ	4 недели	Проверка программы
5. Поиск границ	10 неделя 3 семестра	ИДЗ	4 недели	Проверка программы
6. Оптимизация	15 неделя 3 семестра	ИДЗ	3 недели	Проверка программы

Критерии оценивания

Действует балльно-рейтинговая система оценки знаний обучающихся. Суммарно по дисциплине (модулю) можно получить максимум 100 баллов за семестр.

В течение каждого семестра студентам последовательно выдается набор лабораторных работ, каждая из которых имеет вес от 10% до 20%. В 2-м семестре также предлагается для выполнения набор практических заданий. Каждое из заданий имеет вес от 10% до 15%.

Посещаемость занятий также учитывается и имеет вес 2%. Для получения зачета во 2 семестре необходимо набрать не менее 60%, Для получения экзамена в 3-ом семестре необходимо:

86 % и более – "отлично",

71-85 % – "хорошо",

56-70 % – "удовлетворительно",

55 % и менее – "неудовлетворительно".

ТИПОВЫЕ ИНДИВИДУАЛЬНЫЕ ЗАДАНИЯ

Задача А. Градиентный спуск

Входной файл: Стандартный вход

Ограничение времени: 1 сек

Выходной файл: Стандартный выход

Ограничение памяти: 512 Мб

- **Условие**

Требуется реализовать класс на языке Python, который соответствует следующему интерфейсу.

```
class GradientOptimizer:
    def __init__(self, oracle, x0):
        self.oracle = oracle
        self.x0 = x0

    def optimize(self, iterations, eps, alpha):
        pass
```

В конструктор принимаются два аргумента — оракул, с помощью которого можно получить градиент оптимизируемой функции, а также точку, с которой необходимо начать градиентный спуск.

Метод `optimize` принимает максимальное число итераций для критерия остановки, L2-норму градиента, которую можно считать оптимальной, а также `learning rate`. Метод возвращает оптимальную точку.

Оракул имеет следующий интерфейс:

```
class Oracle:
    def get_func(self, x)
    def get_grad(self, x)
```

`x` имеет тип `np.array` вещественных чисел.

- **Формат выходных данных**

Код должен содержать только класс и его реализацию. Он не должен ничего выводить на экран.

Задача В. Линейная регрессия. Основы

Входной файл: Стандартный вход

Ограничение времени: 1 сек

Выходной файл: Стандартный выход

Ограничение памяти: 512 Мб

- **Условие**

Требуется реализовать следующие функции на языке Python.

```
def linear_func(theta, x) # function value
def linear_func_all(theta, X) # 1-d np.array of function values of
all rows of the matrix X
def mean_squared_error(theta, X, y) # MSE value of current regression
def grad_mean_squared_error(theta, X, y) # 1-d array of gradient by theta
```

`theta` — одномерный `np.array`

x — одномерный `np.array`

X — двумерный `np.array`. Каждая строка соответствует по размерности вектору θ

y — реальные значения предсказываемой величины

Матрица XX имеет размер $M \times NM \times N$. MM строк и NN столбцов.

Используется линейная функция вида: $h_{\theta}(x) = \theta_1 x_1 + \theta_2 x_2 + \dots + \theta_n x_n$

Mean squared error (MSE) как функция от θ : $J(\theta) = \frac{1}{M} \sum_{i=1}^M (y_i - h_{\theta}(x^{(i)}))^2$

Где $x^{(i)}$ — i -я строка матрицы XX

Градиент функции MSE: $\nabla J(\theta) = \{\partial J / \partial \theta_1, \partial J / \partial \theta_2, \dots, \partial J / \partial \theta_N\}$

- **Пример**

```
X = np.array([[1, 2], [3, 4], [4, 5]])
```

```
theta = np.array([5, 6])
```

```
y = np.array([1, 2, 1])
```

```
linear_func_all(theta, X) # -> array([17, 39, 50])
```

```
mean_squared_error(theta, X, y) # -> 1342.0
```

```
grad_mean_squared_error(theta, X, y) # -> array([215.33333333, 283.33333333])
```

- **Формат выходных данных**

Код должен содержать только реализацию функций.

Задача С. Найти линейную регрессию

Входной файл: Стандартный вход

Ограничение времени: 10 сек

Выходной файл: Стандартный выход

Ограничение памяти: 512 Мб

- **Условие**

Требуется реализовать функцию на языке Python, которая находит линейную регрессию заданных векторов, используя метрику MSE.

```
def fit_linear_regression(X, y) # np.array of linear regression coeffs
```

X — двумерный `np.array`. Каждая строка соответствует отдельному примеру.

y — реальные значения предсказываемой величины

- **Формат выходных данных**

Код должен содержать только реализацию функций.

Задача А. Распределение задач

Входной файл: `input.txt`

Ограничение времени: 1 сек

Выходной файл: `output.txt`

Ограничение памяти: 256 Мб

- **Условие**

Группа разработчиков работает над проектом. Весь проект разбит на задачи, для каждой задачи указывается ее категория сложности (1, 2, 3 или 4), а также оценочное время выполнения задачи в часах. Проект считается выполненным, если выполнены все задачи. Для каждого разработчика и для каждой категории сложности задачи указывается коэффициент, с которым, как ожидается, будет соотноситься реальное время выполнения задачи данным разработчиком к оценочному времени. Считается, что все разработчики начинают работать с проектом в одно и то же время и выделяют для работы одинаковое время. Необходимо реализовать программу, распределяющую задачи по разработчикам, с целью минимизировать время выполнения проекта (получить готовый проект за минимальный промежуток времени). Поиск решения необходимо реализовать с помощью генетического алгоритма.

- **Отправка решения и тестирование**

Данная задача будет проверяться на *ОДНОМ* входном файле. Этот файл можно скачать [ЗДЕСЬ](#).

В качестве решения принимается текстовый файл, содержащий ответ к задаче в требуемом формате (при его отправке следует выбрать в тестирующей системе среду разработки "Answer text").

Решение набирает количество баллов, вычисляемое по следующей формуле: $Score = 106 \cdot T_{max}$. T_{max} — наибольшее среди всех разработчиков время, затраченное на выполнение выданных соответствующему разработчику задач.

- **Формат входного файла**

Первая строка входного файла содержит целое число NN количество задач.

Вторая строка — NN целых чисел от 1 до 4 категорий сложности задач.

Третья строка — NN вещественных положительных чисел оценочного времени для задач.

Четвертая строка — целое число MM, количество разработчиков.

Следующие MM строк содержат по 4 вещественных положительных числа — коэффициенты каждого разработчика.

- **Формат выходного файла**

Первая и единственная строка выходного файла содержит NN целых чисел w_i — номер разработчика, назначенного на i -ю задачу.

- **Ограничения**

- **Примеры тестов**

№	Входной файл (input.txt)	Выходной файл (output.txt)
1	3 1 1 4 5.2 3.4 4 2 1 1 2 5 0.7 1 1.2 1.5	1 2 2

Задача А. Логистическая регрессия. Основы

Входной файл: Стандартный вход

Ограничение времени: 1 сек

Выходной файл: Стандартный выход

Ограничение памяти: 512 Мб

- **Условие**

Требуется реализовать следующие функции на языке Python.

```
def logistic_func(theta, x) # function value
def logistic_func_all(theta, X) # 1-d np.array of function values
of all rows of the matrix X
def cross_entropy_loss(theta, X, y) # cross entropy loss value of
current regression
def grad_cross_entropy_loss(theta, X, y) # 1-d array of gradient by theta
```

theta — одномерный np.array

x — одномерный np.array

X — двумерный np.array. Каждая строка соответствует по размерности вектору theta

y — реальные значения предсказываемой величины

Матрица XX имеет размер $M \times N$. MM строк и NN столбцов.

Используется линейная функция вида: $h_{\theta}(x) = \theta_1 x_1 + \theta_2 x_2 + \dots + \theta_n x_n$

- **Формат выходных данных**

Код должен содержать только реализацию функций.

Задача В. Найти логистическую регрессию

Входной файл: Стандартный вход

Ограничение времени: 10 сек

Выходной файл: Стандартный выход

Ограничение памяти: 512 Мб

- **Условие**

Требуется реализовать функцию на языке Python, которая находит логистическую регрессию заданных векторов, используя метрику cross entropy loss.

```
def fit_logistic_regression(X, y) # np.array of logistic regression coeffs
```

X — двумерный np.array. Каждая строка соответствует отдельному примеру.

y — реальные значения предсказываемой величины

- **Формат выходных данных**

Код должен содержать только реализацию функций.

Задача А. News category

Входной файл: input.txt

Ограничение времени: 1 сек

Выходной файл: output.txt

Ограничение памяти: 256 Мб

- **Условие**

Требуется обучить модель определения категории новости. Обучающую выборку можно скачать [ЗДЕСЬ](#). Категория новости в обучающей выборке представлена столбцом CAT.

- HEADER — заголовок новости
- MEDIUMNAME — название СМИ
- WEBSITE — вебсайт СМИ
- PTIME — время публикации

Для определения качества модели будет использоваться тестовая выборка, доступная [ЗДЕСЬ](#).

В тестовой выборке требуется предсказать значения столбца CAT, соответствующие каждому тестовому примеру. Категории новостей кодируются одним символом, аналогично данным в обучающей выборке.

- **Отправка решения и тестирование**

Данная задача будет проверяться на *ОДНОМ* входном файле.

В качестве решения принимается текстовый файл, содержащий ответ к задаче в требуемом формате (при его отправке следует выбрать в тестирующей системе среду разработки "Answer text").

Решение набирает количество баллов, вычисляемое по следующей формуле: $Score = 105 \cdot AccuracyScore$. AccuracyScore — доля верно классифицированных новостей относительно всех новостей в тестовой выборке.

- **Формат выходного файла**

Каждая строка выходного файла должна содержать единственный символ, задающий категорию соответствующего тестового примера.

Задача А. Качество вина

Входной файл: input.txt

Ограничение времени: 1 сек

- **Условие**

Требуется обучить модель определения качества вина. Качество вина определяется по 1010-балльной шкале. В данной задаче будем использовать бинарную модель и предсказывать, "хорошее" вино или "плохое". Хорошим будем считать вино с качеством строго выше 66. Обучающую выборку можно скачать [ЗДЕСЬ](#). Качество вина представлено столбцом `quality`.

Для определения качества модели будет использоваться тестовая выборка, доступная [ЗДЕСЬ](#).

В тестовой выборке требуется предсказать значения 11 или 00, "хорошее" вино или "плохое" соответственно, для каждого примера. Оценку качества по 1010-балльной шкале предсказывать не требуется.

- **Отправка решения и тестирование**

Данная задача будет проверяться на *ОДНОМ* входном файле.

В качестве решения принимается текстовый файл, содержащий ответ к задаче в требуемом формате (при его отправке следует выбрать в тестирующей системе среду разработки "Answer text").

Решение набирает количество баллов, вычисляемое по следующей формуле: $Score=105 \cdot F1$ $Score=105 \cdot F1$.

- **Формат выходного файла**

Каждая строка выходного файла должна содержать целое число 11 или 00. Количество строк должно быть равно количеству элементов контрольной выборки.



МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное образовательное учреждение
высшего образования
«Дальневосточный федеральный университет»
(ДВФУ)

ШКОЛА ЦИФРОВОЙ ЭКОНОМИКИ

ФОНД ОЦЕНОЧНЫХ СРЕДСТВ

по дисциплине «Машинное обучение»

Направление подготовки 09.04.01 «Информатика и вычислительная техника»

Форма подготовки очная

Владивосток
2019

Сформированность каждой компетенции в рамках освоения данной дисциплины оценивается по трех уровневой шкале:

- пороговый уровень является обязательным для всех обучающихся по завершении освоения дисциплины;
- продвинутый уровень характеризуется превышением минимальных характеристик сформированности компетенции по завершении освоения дисциплины;
- эталонный уровень характеризуется максимально возможной выраженностью компетенции и является важным качественным ориентиром для самосовершенствования.

Уровень сформированности каждой компетенции на различных этапах ее формирования в процессе освоения данной дисциплины оценивается в ходе текущего контроля успеваемости представлен различными видами оценочных средств.

Код и формулировка компетенции	Этапы формирования компетенции
<p>ОПК-1. Способен самостоятельно приобретать, развивать и применять математические, естественнонаучные, социально-экономические и профессиональные знания для решения нестандартных задач, в том числе в новой или незнакомой среде и в междисциплинарном контексте</p>	<p>ОПК-1.1 Знать: математические, естественнонаучные и социально-экономические методы для использования в профессиональной деятельности</p> <p>ОПК-1.2 Уметь: решать нестандартные профессиональные задачи, в том числе в новой или незнакомой среде и в междисциплинарном контексте, с применением математических, естественнонаучных, социально-экономических и профессиональных знаний</p> <p>ОПК-1.3 Владеть: навыками теоретического и экспериментального исследования объектов профессиональной деятельности, в том числе в новой или незнакомой среде и в междисциплинарном контексте</p>
<p>ОПК-2. Способен разрабатывать оригинальные алгоритмы и программные средства, в том числе с использованием современных интеллектуальных технологий, для решения профессиональных задач</p>	<p>ОПК-2.1 Знать: современные информационно-коммуникационные и интеллектуальные технологии, инструментальные среды, программно-технические платформы для решения профессиональных задач</p> <p>ОПК-2.2 Уметь: обосновывать выбор современных информационно-коммуникационных и интеллектуальных технологий, разрабатывать оригинальные программные средства для решения профессиональных задач</p> <p>ОПК-2.3 Владеть: навыками разработки оригинальных программных средств, в том числе с использованием современных</p>

МАТЕРИАЛЫ ЗАЧЕТА

по дисциплине «Машинное обучение»

Основные вопросы

Введение. Методы оптимизации. Градиентный спуск

1. Задача регрессии, классификации.
2. Функция потерь. Оптимизация. Перебор по сетке.
3. Производная, частные производные, градиент.
4. Градиентный спуск, проблема выбора шага.
5. Стохастический градиентный спуск. Использование момента.
6. Adagrad, Adadelata, Adam.
7. RMSProp*.

Линейная регрессия

8. Постановка задачи линейной регрессии.
9. Метод наименьших квадратов.
10. Ковариация, корреляция.
11. Критерий R^2 .
12. Анализ остатков.

Глобальная оптимизация. Генетический алгоритм

13. Многопараметрическая оптимизация.
14. Доминанция и оптимальность по Парето.
15. Функция качества (fitness). Аппроксимация качества.
16. Общая идея генетического алгоритма.
17. Представление генома.
18. Методы селекции: пропорционально качеству, универсальная выборка (stochastic universal sampling), с наследием (reward-based), турнир. Стратегия элитизма.

19. Методы кроссовера. Двух и многоточечный, равномерный (по подмножествам), для перестановок.
20. Мутация. Влияние на скорость обучения.
21. Управление популяцией. Сегрегация, старение, распараллеливание.
22. Генетическое программирование.

Метод ближайших соседей (k-NN)

23. Понятие и свойства метрики. Ослабление требования к неравенству треугольника.
24. Базовый алгоритм классификации методом 1-NN и k-NN. Преимущества и недостатки.
25. Метрики L1, L2, Хемминга, Левенштейна, косинусное расстояние.
26. Потеря точности нормы в высоких размерностях.
27. Нормализация координат. Предварительная трансформация пространства признаков.
28. Метрика Махаланобиса.
29. Кросс-валидация методом "без одного" (leave one out).
30. Определение границ, показатель пограничности.
31. Сжатие по данным. Понятия выброса, прототипа, усвоенной точки. Алгоритм Харта (Hart).
32. Регрессия методом k-NN.
33. Взвешенные соседи.
34. Связь с градиентным спуском. Стохастическая формулировка, softmax.
35. Метод соседних компонент (neighbour component analysis)*.
36. Связь с выпуклой оптимизацией. Метод большого запаса (Large margin NN)
37. Оптимизация классификатора, k-d деревья
38. Хеши чувствительные к локальности, хеши сохраняющие локальность*.

МАТЕРИАЛЫ ЭКЗАМЕНА

по дисциплине «Машинное обучение»

Наивный байесов классификатор

1. Условная вероятность. Байесово решающее правило. Обновление вероятностей.
2. Наивный классификатор, предположение о независимости признаков.
3. Оценка плотности распределения для числовых признаков.
4. Алгоритмические оптимизации.
5. Алгоритм EM.

Логистическая регрессия

6. Сигмоид
7. Метод наибольшего правдоподобия
8. Логистическая регрессия для меток $-1, 1$

Деревья решений

9. Понятие дерева решений.
10. Борьба с оверфиттингом: bagging, выборки признаков.
11. Ансамбли, случайный лес (Random Forest).
12. Понятие энтропии, определение информации по Шеннону.
13. Метрики: примеси Джини (Gini impurity), добавленная информация (information gain).
14. Деревья регрессии. Метрика вариации.
15. Непрерывные признаки. Использование главных компонент вместо признаков.
16. Сокращение дерева (pruning).

Кластеризация

17. Задача обучения без учителя, применения при эксплораторном анализе

18. Неметрическая кластеризация: функция схожести, компоненты связности и остовные деревья, иерархическая кластеризация снизу вверх
19. Метрики, понятие центроида и представителя класса
20. Центроидные алгоритмы: k-means, k-medoid
21. Алгоритмы, основанные на плотности: DBSCAN, OPTICS
22. Алгоритмы, основанные на распределении: сумма гауссиан
23. Нечёткая кластеризация, алгоритм c-means
24. Метрики качества: leave-one-out, силуэт, индекс Дэвиса-Болдина (Davies-Bouldin), индекс Данна (Dunn)