**1. ОБЩАЯ ХАРАКТЕРИСТИКА ПРОГРАММЫ**

Образовательная программа соответствует:

* Федеральному закону от 29 декабря 2012 г. № 273-ФЗ «Об образовании в Российской Федерации»;
* Приказу Минобрнауки России от 1 июля 2013 г. № 499 "Об утверждении Порядка организации и осуществления образовательной деятельности по дополнительным профессиональным программам";
* Приказу Минобрнауки от 23 августа 2017 г. № 816 «Об утверждении Порядка применения организациями, осуществляющими образовательную деятельность, электронного обучения, дистанционных образовательных технологий при реализации образовательных программ»

**1.1 Цель программы**

Получение новых профессиональных компетенций, необходимых для выполнения следующих видов профессиональной деятельности в рамках имеющейся квалификации:

* Способен программировать на языке Python;
* Способен применять базовые методы машинного обучения для решения задач профессиональной деятельности;
* Способен реализовывать обучение нейронных сетей на фреймворке PyTorch.

**1.2 Планируемые результаты обучения**

В результате освоения программы слушатель должен приобрести следующие знания и умения, необходимые для качественного изменения компетенций, указанных в п. 1.1:

**Слушатель должен знать:**

* базовые операторы и структуры данных языка Python;
* реализацию основных алгоритмов на языке Python.
* основные задачи и базовые методы машинного обучения;
* принципы выбора методов машинного обучения при решении практических задач;
* современные сервисы, построенные на основе ИИ;
* основы построения и обучения нейронных сетей различного типа с использованием фреймворка PyTorch.

**Слушатель должен уметь:**

* использовать операторы и базовые структуры данных языка Python при решении задач;
* подбирать наиболее подходящие алгоритмы при решении задач на языке Python;
* решать основные задачи машинного обучения;
* строить базовые модели машинного обучения;
* использовать сервисы на основе ИИ для решения практических задач;
* использовать фреймворк PyTorch для построения и обучения нейронных сетей различного типа.

**Слушатель должен владеть:**

* навыками использования средств языка Python для решения практических задач;
* методами подбора наиболее релевантного алгоритма для решения практических задач;
* навыками анализа условий задач машинного обучения;
* навыками реализации методов машинного обучения на языке Python;
* навыками работы с различными онлайн-сервисами, в основе которых лежит ИИ;
* навыками решения практических задач с использованием нейронных сетей и фреймворка PyTorch.

**Программа разработана с учетом** профессионального стандарта «Программист» (утвержден Приказом Минтруда от 18 ноября 2013 г. N 679н) в соответствии со следующими параметрами функциональной карты вида профессиональной деятельности:

|  |  |
| --- | --- |
| Обобщенная трудовая функция | Трудовая функция |
| Код | Наименование | Уровень квалификации | Наименование | Код | Уровень (подуровень) квалификации |
| A | Разработка и отладка программного кода | 3 | Формализация и алгоритмизация поставленных задач | A/01.3 | 3 |
| Написание программного кода с использованием языков программирования, определения и манипулирования данными | A/02.3 | 3 |
| Оформление программного кода в соответствии с установленными требованиями | A/03.3 | 3 |
| Проверка и отладка программного кода | A/05.3 | 3 |
| Обобщенная трудовая функция | Трудовая функция |
| Код | Наименование | Уровень квалификации | Наименование | Код | Уровень (подуровень) квалификации |
| B | Проверка работоспособности и рефакторинг кода программного обеспечения | 4 | Проверка работоспособности программного обеспечения | B/03.4 | 4 |

Программа полностью реализуется с применением дистанционных образовательных технологий.

**1.3 Требования к уровню подготовки поступающего на обучение**

К освоению программы повышения квалификации допускаются лица:

* Имеющие среднее профессиональное и (или) высшее образование.

**2. СОДЕРЖАНИЕ ПРОГРАММЫ**

**2.1 Учебный план**

дополнительной профессиональной программы повышения квалификации

«Машинное обучение и искусственный интеллект: с нуля до результата»

Общая трудоемкость программы – 72 академических часа

Форма обучения – очно-заочная с применением дистанционных образовательных технологий.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| № п/п | Наименование раздела,дисциплины(модуля) | Трудоемкость, час. | Дистанционныезанятия | СРС, час. | Промежу-точнаяаттестация2)  |
| Всего, час | из них | Зачет | Экзамен |
| лекции | лабор. работы | практ.занятия, семинары |
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| 1. | Основы ИИ | 8 | 8 | 4 | 0 | 3 |  | 1Д |  |
| 2. | Основы Python | 16 | 16 | 4 | 0 | 10 |  | 2Д |  |
| 3. | Базовые методы машинного обучения | 20 | 20 | 5 | 0 | 14 |  | 1Д |  |
| 4. | Нейронные сети | 14 | 14 | 3 | 0 | 10 |  | 1Д |  |
| 5. | Современные методы ИИ и их применение | 12 | 12 | 4 | 0 | 7 |  | 1Д |  |
| Итого: | 70 | 70 | 20 | 0 | 44 |  | 6 |  |
| Итоговая аттестация | 2 | 2 | 0 | 0 | 0 |  | 2Д |  |
| Всего: | 72 | 72 | 20 | 0 | 44 |  | 8 |  |

Проведение текущего контроля не предусмотрено.

**2.2 Учебно-тематический план**

дополнительной профессиональной программы повышения квалификации

«Машинное обучение и искусственный интеллект: с нуля до результата»

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| № п/п | Наименование раздела,дисциплины(модуля) | Трудоемкость, час. | Дистанционныезанятия | СРС, час. | Промежу-точнаяаттестация2)  |
| Всего, час | из них | Зачет | Экзамен |
| лекции | лабор. работы | практ.занятия, семинары |
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| **1.** | **Основы ИИ** | **8** | **8** | **4** | **0** | **3** |  | **1Д** |  |
| 1.1 | История развития ИИ | 3 | 3 | 1 | 0 | 2 |  | 0 |  |
| 1.2 | Современные сервисы на основе ИИ | 2 | 2 | 1 | 0 | 1 |  | 0 |  |
| 1.3 | Кто разрабатывает ИИ? | 2 | 2 | 2 | 0 | 0 |  | 0 |  |
| 1.4 | Промежуточная аттестация по модулю 1 | 1 | 1 | 0 | 0 | 0 |  | 1Д |  |
| **2.** | **Основы Python** | **16** | **16** | **4** | **0** | **10** |  | **2Д** |  |
| 2.1 | Введение в Python | 2 | 2 | 0 | 0 | 2 |  | 0 |  |
| 2.2 | Структуры данных в Python | 3 | 3 | 1 | 0 | 2 |  | 0 |  |
| 2.3 | Циклы и условия | 3 | 3 | 1 | 0 | 2 |  | 0 |  |
| 2.4 | Чтение данных | 3 | 3 | 1 | 0 | 2 |  | 0 |  |
| 2.5 | Создание и работа с датафреймом. Функции | 3 | 3 | 1 | 0 | 2 |  | 0 |  |
| 2.6 | Промежуточная аттестация по модулю 2 | 2 | 2 | 0 | 0 | 0 |  | 2Д |  |
| **3.** | **Базовые методы машинного обучения** | **20** | **20** | **5** | **0** | **14** |  | **1Д** |  |
| 3.1 | Понятие машинного обучения | 2 | 2 | 1 | 0 | 1 |  | 0 |  |
| 3.2 | Задачи машинного обучения | 3 | 3 | 2 | 0 | 1 |  | 0 |  |
| 3.3 | Построение модели машинного обучения | 4 | 4 | 2 | 0 | 2 |  | 0 |  |
| 3.4 | Линейная и логистическая регрессия | 4 | 4 | 0 | 0 | 4 |  | 0 |  |
| 3.5 | Деревья решений | 4 | 4 | 0 | 0 | 4 |  | 0 |  |
| 3.6 | Ансамблевые методы машинного обучения | 2 | 2 | 0 | 0 | 2 |  | 0 |  |
| 3.7 | Промежуточная аттестация по модулю 3 | 1 | 1 | 0 | 0 | 0 |  | 1Д |  |
| **4.** | **Нейронные сети** | **14** | **14** | **3** | **0** | **10** |  | **1Д** |  |
| 4.1 | Модель нейрона. Применение нейронных сетей. | 1 | 1 | 1 | 0 | 0 |  | 0 |  |
| 4.2 | Обучение нейронных сетей | 3 | 3 | 1 | 0 | 2 |  | 0 |  |
| 4.3 | Глубокое обучение. Многослойные нейронные сети. | 5 | 5 | 1 | 0 | 4 |  | 0 |  |
| 4.4 | Фреймворк PyTorch. Реализация нейронов и сетей на PyTorch. | 4 | 4 | 0 | 0 | 4 |  | 0 |  |
| 4.5 | Промежуточная аттестация по модулю 4 | 1 | 1 | 0 | 0 | 0 |  | 1Д |  |
| **5.** | **Современные методы ИИ и их применение** | **12** | **12** | **4** | **0** | **7** |  | **1Д** |  |
| 5.1 | Сверточные нейронные сети | 2 | 2 | 1 | 0 | 1 |  | 0 |  |
| 5.2 | Рекуррентные нейронные сети | 2 | 2 | 1 | 0 | 1 |  | 0 |  |
| 5.3 | Обработка естественного языка | 1 | 1 | 0 | 0 | 1 |  | 0 |  |
| 5.4 | Классификация изображений | 5 | 5 | 1 | 0 | 4 |  | 0 |  |
| 5.5 | Будущее ИИ | 1 | 1 | 1 | 0 | 0 |  | 0 |  |
| 5.6 | Промежуточная аттестация по модулю 5 | 1 | 1 | 0 | 0 | 0 |  | 1Д |  |
| Итого: | 70 | 70 | 20 | 0 | 44 |  | 6 |  |
| Итоговая аттестация | 2 | 2 | 0 | 0 | 0 |  | 2Д |  |
| Всего: | 72 | 72 | 20 | 0 | 44 |  | 8 |  |

Проведение текущего контроля не предусмотрено.

**2.3 Календарный учебный график**

Календарный учебный график формируется до начала обучения и утверждается отдельным распоряжением.

|  |  |  |  |
| --- | --- | --- | --- |
| №п/п | Наименованиеразделов | **Объем учебной нагрузки, час.** | **Учебные недели** |
| 1 день / неделя | 2 день / неделя | 3 день / неделя | 4 день/ неделя | 5 день/неделя | 6 день / неделя | 7 день / неделя |
| 1 | Основы ИИ | 8 | 8 |  |  |  |  |  |  |
| 2 | Основы Python | 16 | 12 | 4 |  |  |  |  |  |
| 3 | Базовые методы машинного обучения | 20 |  | 16 | 4 |  |  |  |  |
| 4 | Нейронные сети | 14 |  |  | 14 |  |  |  |  |
| 5 | Современные методы ИИ и их применение | 12 |  |  | 2 | 10 |  |  |  |
| 6 | Итоговая аттестация | 2 |  |  |  | 2 |  |  |  |
|  | ИТОГО | **72** |  |  |  |  |  |  |  |

**2.4 Рабочая программа**

**РАЗДЕЛ 1. Основы ИИ (8 часов)**

**Тема 1.1 История развития ИИ**

История развития информационных технологий и компьютерной техники как фактор появления и становления искусственного интеллекта. Понятие генетического алгоритма как математической основы искусственного интеллекта. Примеры работы генетических алгоритмов. Соотношение искусственного интеллекта и машинного обучения. Современный искусственный интеллект в играх (шахматы, го).

**Тема 1.2 Современные сервисы на основе ИИ**

Обзор сервисов, работающих на основе искусственного интеллекта и их возможностей:

* <https://www.remove.bg/ru> Удаление фона на изображениях
* <https://www.jukedeck.com/> Сервис для создания музыкальных треков различных жанров. Всё, что требуется от пользователя, – это определить начальные параметры будущей композиции (жанр, темп, настроение, длительность, состав инструментов), после чего щёлкнуть по клавише Create Track и дождаться завершения обработки запроса. Сочинённую искусственным интеллектом музыку можно прослушать в браузере, скачать на компьютер либо отправить на доработку, откорректировав характеристики трека.
* <https://www.autodraw.com/> Сервис, превращающий рисунки от руки в высококачественные клип-арты. Положенный в основу AutoDraw искусственный интеллект в реальном времени анализирует пользовательские наброски, распознаёт их и предлагает аналогичные картинки, нарисованные профессиональными художниками. Созданные иллюстрации можно разместить в социальных сетях либо скачать на компьютер для дальнейшего использования. Важно отметить, что разработанный компанией Google сервис прекрасно подходит не только для развлечения, но и для решения вполне реальных задач. Например, добрую службу AutoDraw может сослужить дизайнерам-оформителям презентаций, иллюстраторам, фоторедакторам и представителям прочих творческих профессий.
* <https://deepart.io/> Сервис, предназначенный для работы с графикой и создания оригинальных картин на основе пользовательских изображений. Техника работы с Deepart.io предельно простая: загружаем на сервер сервиса фотографию, указываем предпочтительный художественный стиль и дожидаемся завершения процесса отрисовки картины, который может занять продолжительное время. Для тех, кто не желает ждать, разработчики сервиса предлагают несколько вариантов платных подписок, позволяющих не только свести к минимуму время рендеринга шедевров цифрового искусства, но и снять ограничения на размер выходных изображений.
* <https://www.beautiful.ai/> Онлайн инструмент для создания презентаций, использующий технологии искусственного интеллекта с целью автоматизации и упрощения работы пользователя со слайдами. «Умные» алгоритмы сервиса контролируют каждый шаг при работе с презентацией и делают так, чтобы просмотр слайдов был более комфортным. Beautiful.ai анализирует расположение элементов презентации и автоматически перестраивает слайды, корректирует их цветовое оформление, перерисовывает графики, подбирает анимационные переходы, рекомендует подходящие по тематике контента шаблоны и выполняет прочие действия, стараясь, чтобы подача материала на слайдах была профессиональной с точки зрения дизайна.

**Тема 1.3 Кто разрабатывает ИИ?**

Базовый состав команды ИИ проекта.

* Специалист по данным.
* Инженер данных.
* Дизайнеры продукта
* Специалисты по этике ИИ и социологии

**Тема 1.4. Промежуточная аттестация по модулю 1.**

Контрольный тест по пройденному материалу.

**РАЗДЕЛ 2. Основы Python (16 часов)**

**Тема 2.1 Введение в Python**

Операции с целыми числами. Сообщения об ошибках. Числа с плавающей точкой. Переменные, их обозначение. Логические операции. Условия. Строковые переменные. Комментарии в программном коде.

**Тема 2.2 Структуры данных в Python**

Создание объекта Series из списка целых чисел. Создание Series из строковых значений. Другие способы создания объекта Series. Создание объекта DataFrame для списка объектов Series. Другие способы создания объекта DataFrame. Свойства объектов Series и DataFrame: количество элементов, количество уникальных элементов, индексы и значения элементов и другие. Различные способы вывода значений объектов Series и DataFrame. Поиск значений в Series и DataFrame. Транспонирование. Срезы данных.

**Тема 2.3 Циклы и условия**

Циклы for (различные виды и конфигурации). Цикл while. Операторы break и continue. Условия в циклах. Оператор if. Оператор else. Оператор elif.

**Тема 2.4 Чтение данных**

Загрузка данных из файла. Запись в файл. Пропущенные значения. Повторяющиеся значения. Замена значений. Применение функций.

**Тема 2.5 Создание и работа с датафреймом. Функции**

Оператор pandas: info. Описательная статистика (pandas: describe). Применение функций для каждого элемента (pandas: map). Применение лямбда-функций. Сортировка (pandas: sort). Соединение датафреймов: конкатенация и pandas: merge. Группировка pandas: groupby. Сводные таблицы pandas: pivot\_table.

**Тема 2.6 Промежуточная аттестация по модулю 2.**

Контрольный тест по пройденному материалу.

**РАЗДЕЛ 3. Базовые методы машинного обучения (20 часов)**

**Тема 3.1 Понятие машинного обучения**

Мотивы для создания технологии. Основные определения. Шкалы измерения различных характеристик. Источники, порождающие данные.

**Тема 3.2 Задачи машинного обучения**

* Ассоциация - нахождение постоянных составляющих (трендов), которые можно использовать для объяснения событий (например, выбор товаров и услуг, определение уровней запасов, схем складирования и др.).
* Последовательность - установление временных серий последовательных действий (транзакций), правил выполнения отдельных транзакций.
* Классификация - выявление признаков, характеризующих группу объектов, распределение объектов по группам для моделирования поведения объектов, прогнозирования значений свойств объектов.
* Кластеризация - распределение по группам или сегментам. В отличие от классификации, кластеры формируются в процессе анализа.
* Прогнозирование - предсказания будущих значений непрерывно изменяющихся переменных и др.

**Тема 3.3 Построение модели машинного обучения**

Этапы в процессе машинного обучения. Обзор методов машинного обучения.

**Тема 3.4 Линейная и логистическая регрессия**

Линейная регрессия на примере зависимости роста детей от роста родителей. Логистическая регрессия – это разновидность множественной регрессии, общее назначение которой состоит в анализе связи между несколькими независимыми переменными (называемыми также регрессорами или предикторами) и зависимой переменной. Бинарная логистическая регрессия, как следует из названия, применяется в случае, когда зависимая переменная является бинарной (т.е. может принимать только два значения). Иными словами, с помощью логистической регрессии можно оценивать вероятность того, что событие наступит для конкретного испытуемого (больной/здоровый, возврат кредита/дефолт и т.д.). Логистическая регрессия на примере данных, выживших/погибших на Титанике. Ошибки первого и второго рода. ROC-кривая.

**Тема 3.5 Деревья решений**

Задачи, приводящие к деревьям решений. Алгоритм построения дерева решений. Определение наилучшего разбиения. Прирост информации о множестве объектов. Меры неопределенности. Важность признаков. Работа с пропусками в данных. Специальные алгоритмы построения деревьев.

**Тема 3.6 Ансамблевые методы машинного обучения**

Случайный лес (Random Forest). Метод градиентного бустинга (Xgboost).

**Тема 3.7 Промежуточная аттестация по модулю 3.**

Контрольный тест по пройденному материалу.

**РАЗДЕЛ 4. Нейронные сети (14 часов)**

**Тема 4.1 Модель нейрона. Применение нейронных сетей**

Искусственный нейрон (формальный нейрон) - элемент искусственных нейронных сетей, моделирующий некоторые функции биологического нейрона. Перцептрон Розенблатта. Структура нейрона. Работа нейрона. Пороговая функция. Обучение нейрона. Области применения нейронных сетей: Компьютерное зрение, Обработка естественного языка, Управление, Прогнозирование и классификация. Недостатки нейронных сетей.

**Тема 4.2 Обучение нейронных сетей**

Реализация класса Perceptron() – нейрон с пороговой функцией активации, его обучение и трассировка на сгенерированных и реальных данных. Сравнение качества работы обученного класса с классом из библиотеки scikit-learn (sklearn.linear\_model.Perceptron()). Реализация нейрона с сигмоидной функцией.

**Тема 4.3 Глубокое обучение. Многослойные нейронные сети**

Структура полносвязной нейронной сети. Функционирование отдельного нейрона в сети. Работа первого слоя нейронной сети. Работа второго слоя нейронной сети. Работа i-го слоя. Прохождение данных через нейросеть (forward pass). Процесс распространения ошибок обратно для обновления весов сети (Backward pass).

**Тема 4.4 Фреймворк PyTorch. Реализация нейронов и сетей на PyTorch**

* Знакомство с фреймворком PyTorch
* Реализация одного нейрона и нейронных сетей на PyTorch
* Реализация многослойной нейронной сети на PyTorch

**Тема 4.5 Промежуточная аттестация по модулю 4.**

Контрольный тест по пройденному материалу.

**РАЗДЕЛ 5. Современные методы ИИ и их применение (12 часов)**

**Тема 5.1 Сверточные нейронные сети**

* Операция свёртки
* Свёрточный и пулинг слои
* Техника Transfer Learning

**Тема 5.2 Рекуррентные нейронные сети**

* Введение
* Forward pass
* Backward pass
* Примеры задач
* Проблемы данного вида сетей
* Архитектура рекуррентного нейрона-LSTM
* Схема LSTM-нейрона-GPU
* Двунаправленные рекуррентные нейронные сети.

**Тема 5.3 Обработка естественного языка**

Задачи обработки естественного языка. Определение «тональности» текстов.

**Тема 5.4 Классификация изображений**

Задачи классификации изображений/распознавания образов. Предсказание классов картинок.

**Тема 5.5 Будущее ИИ**

ИИ в робототехнике, в управлении автомобилем, в системе «умного дома».

**Тема 5.6 Промежуточная аттестация по модулю 5.**

Контрольный тест по пройденному материалу.

**Итоговая аттестация:** Выполнение междисциплинарного теста.

**2.5 Перечень и описание практико-ориентированных заданий (в соответствии с тематическим планом)**

Перечень практических занятий и самостоятельной работы слушателей.

| **№ раздела / темы** | **Наименование тем программы и учебных вопросов** | **Кол-во часов** |
| --- | --- | --- |
| 1.1 | История развития ИИ.Изучение основных определений и истории развития искусственного интеллекта. | 2 |
| 1.2 | Современные сервисы на основе ИИОбзор сервисов, в основе работы которых лежит ИИ | 1 |
| 1.4 | Промежуточная аттестация по модулю 1 | 1 |
| 2.1 | Введение в PythonОсновные операции и базовые операторы Python. | 2 |
| 2.2 | Структуры данных в PythonОсновные структуры данных в Python. | 2 |
| 2.3 | Циклы и условия.Циклы for и while. Условный оператор. | 2 |
| 2.4 | Чтение данныхЧтение данных из файла и запись в файл. | 2 |
| 2.5 | Создание и работа с датафреймом. Функции.Особенности структуры данных датафрейм. Функции в Python. | 2 |
| 2.6 | Промежуточная аттестация по модулю 2 | 2 |
| 3.1 | Понятие машинного обученияОсновные определения машинного обучения. | 1 |
| 3.2 | Задачи машинного обученияОсновные задачи машинного обучения. | 1 |
| 3.3 | Построение модели машинного обученияАлгоритм построения модели машинного обучения. | 2 |
| 3.4 | Линейная и логистическая регрессияРеализация линейной и логистической регрессии | 4 |
| 3.5 | Деревья решенийРеализация деревьев решений | 4 |
| 3.6 | Ансамблевые методы машинного обученияРеализация случайного леса и xgboost | 2 |
| 3.7 | Промежуточная аттестация по модулю 3 | 1 |
| 4.2 | Обучение нейронных сетейАлгоритм обучения нейронных сетей. | 2 |
| 4.3 | Глубокое обучение. Многослойные нейронные сети.Запуск фреймворка PyTorch. Принципы работы и построения многослойных нейронных сетей | 4 |
| 4.4 | Фреймворк PyTorch. Реализация нейронов и сетей на PyTorch.Реализация отдельных нейронов и многослойных нейронных сетей на PyTorch | 4 |
| 4.5 | Промежуточная аттестация по модулю 4 | 1 |
| 5.1 | Сверточные нейронные сетиПостроение сверточных нейронных сетей | 1 |
| 5.2 | Рекуррентные нейронные сетиПостроение рекуррентных нейронных сетей | 1 |
| 5.3 | Обработка естественного языкаПостроение нейронной сети для обработки естественного языка | 1 |
| 5.4 | Классификация изображенийПостроение нейронной сети для классификации изображений | 4 |
| 5.6 | Промежуточная аттестация по модулю 5 | 1 |
|  | Итоговая аттестация | 2 |
| **ИТОГО:** | **52** |

**1. Основы ИИ**

1.1 История развития ИИ.

**Изучение основных определений и истории развития искусственного интеллекта.**

Раскрыть содержание одного из следующих вопросов (в произвольной форме):

* Мышление и искусственный интеллект
* Искусственный интеллект – настоящее и будущее
* Возможен ли искусственный интеллект?
* История и тенденции развития ИИ
* Сущность искусственного интеллекта
* Теория искусственного (машинного) интеллекта
* Построение информационно-управляющей системы с элементами искусственного интеллекта
* Проблема искусственного интеллекта.
* Проблемы построения искусственного интеллекта.
* Производственные системы с искусственным интеллектом.
* Философские проблемы искусственного интеллекта и искусственной жизни.
* Создание искусственного интеллекта.
* Сознание и искусственный интеллект.
* Классификация сейсмических сигналов на основе нейросетевых технологий.
* Искусственный интеллект и теоретические вопросы психологии.
* История развития науки о искусственном интеллекте
* Опасно разумен (ИИ)
* До бессмертия остался один шаг (ИИ)

1.2 Современные сервисы на основе ИИ

**Обзор сервисов, в основе работы которых лежит ИИ**

Выполнить работу и описать основные возможности одного из сервисов, использующих ИИ. Ответить на вопрос, чтобы вам хотелось добавить в работу данного сервиса? Какие функции, какие возможности?

Примеры сервисов приведены в соответствующей лекции, возможно использование другого сервиса:

* <https://www.remove.bg/ru>
* <https://www.jukedeck.com/>
* <https://www.autodraw.com/>
* <https://deepart.io/>
* <https://www.beautiful.ai/>

**2. Основы Python**

2.1 Введение в Python

**Основные операции и базовые операторы Python**

В Jupyter notebook протестируйте основные математические действия с переменной a:

• сложение a + 1

• вычитание a - 20

• умножение (a + 1) \* 2

• деление (a + 5) / 3

• возведение в степень (a + 1) \*\* 5

• деление нацело a//3

• остаток от деления a%3

Действия со строковыми переменными

Форматированный ввод и вывод

Операции сравнения

2.2 Структуры данных в Python

**Основные структуры данных в Python**

Создание и работа с листами (list)

Перевод листа в строку

Выбор элементов листа

Методы строк

Генерация списков

2.3 Циклы и условия

**Циклы for и while. Условный оператор**

Цикл while

Вывести все нечетные числа от 5 до 55

Напишите программу, которая считывает со стандартного ввода целые числа, по одному числу в строке, и после первого введенного нуля выводит сумму полученных на вход чисел.

Напишите программу, которая считывает целые числа с консоли по одному числу в строке. Для каждого введённого числа проверить: если число меньше 10, то пропускаем это число; если число больше 100, то прекращаем считывать числа; в остальных случаях вывести это число обратно на консоль в отдельной строке.

Цикл for

Напишите программу, которая считывает список чисел lst из первой строки и число x из второй строки, которая выводит все позиции, на которых встречается число x в переданном списке lst.

2.4 Чтение данных

**Чтение данных из файла и запись в файл**

Чтение содержимого файла в лист

Чтение файлов с заголовками

Как избежать чтения первой строки

Чтение данных из файлов с несколькими столбцами

Приведение переменных к числовому типу

Подсчет суммы в файлах

Библиотека openpyxl

Чтение содержимого Excel-файла в цикле

2.5 Создание и работа с датафреймом. Функции.

**Особенности структуры данных датафрейм. Функции в Python.**

* Импортируйте файл bike.csv и создайте на его основе pandas dataframe (pandas.core.frame.DataFrame).
* Определите количество строк и столбцов в датафрейме.
* Выведите первые 6 и последние 7 строк датафрейма.
* Выведите список всех столбцов.
* Выведите транспонированный датафрейм (первые 3 элемента).
* Выведите информацию обо всех данных. Есть ли пропуски и если есть, то сколько? Какого типа данные представлены в датафрейме?
* Выведите стандартную описательную статистику по датафрему. Сделайте выводы, касательно данных в столбцах: погода, влажность, скорость ветра и количество (count).
* Изменяя параметры статистики, получите новые описательные результаты, и сделайте не менее трех утверждений на их основе.
* Выведите значения столбца «скорость ветра» начиная с 5 по 20.
* Выведите все значения столбца «temp», значения которых выше среднего.
* Создайте еще один столбец в датафреме с именем «good\_temp» и запишите в него результат сравнения текущей температуры со средним значением (True or False).
* Выведите те строки, в которых есть пустые значения (в любом из столбцов).
* Выведите первые 5 строк, для которых выполняется составное условие: температура > 20 и скорость ветра > 20.
* Выведите значения всех полей в третьей строке списка, полученного на предыдущем шаге/
* Выведите датафрейм, отсортированный по полю count, в обратном порядке.
* Создайте еще один датафрем на основе описания поля season (1 = spring, 2 = summer, 3 = fall, 4 = winter).
* Соедините исходный датафрем с даатфремом, созданным на предыдущем шаге по указанному выше правилу. Выведите полученный результат.
* Сделайте группировку по временам года и количеству (count).
* Создайте pivot\_table по любым столбцам и опишите полученный результат.
* Получите процентное соотношение count в зависимости от времен года.

**3. Базовые методы машинного обучения**

3.1 Понятие машинного обучения

**Основные определения машинного обучения**

Первичный анализ датасета, предобработка данных.

3.2 Задачи машинного обучения

**Основные задачи машинного обучения**

Выполнить сравнение зада машинного обучения между собой.

Описать структуру данных, подходящую под каждую из пяти основных задач машинного обучения.

3.3 Построение модели машинного обучения

**Алгоритм построения модели машинного обучения.**

Описать основные этапы построения модели машинного обучения с обязательным указанием источника данных.

3.4 Линейная и логистическая регрессия

**Реализация линейной и логистической регрессии**

* Вы работаете с файлом dataset\_Facebook.csv, выполните загрузку данных из него.
* Выполните предобработку данных – разберитесь с пропусками при их наличии, все категориальные признаки переведите в численный вид (их более одного). Удалять столбцы из датасета возможно только при наличии корреляции с данными из других столбцов.
* Выполните машинное обучение по модели линейной и логистической регрессии.
* Рассчитайте все основные метрики. Постройте график для ROC AUC.
* Постройте предсказания для тестового множества.

3.5 Деревья решений

**Реализация деревьев решений**

* Вы работаете с файлом winequality-red.csv. Целевая переменная – quality.
* Выполните предобработку данных при необходимости.
* Выполните машинное обучение по модели деревьев решений.
* Исследуйте влияние глубины дерева на модель, используя 5-fold кросс-валидацию в диапазоне от 1 до 15. Особое внимание уделите подбору метрики. Примеры метрик приведены ниже: Valid options are ['accuracy', 'adjusted\_mutual\_info\_score', 'adjusted\_rand\_score', 'average\_precision', 'completeness\_score', 'explained\_variance', 'f1', 'f1\_macro', 'f1\_micro', 'f1\_samples', 'f1\_weighted', 'fowlkes\_mallows\_score', 'homogeneity\_score', 'mutual\_info\_score', 'neg\_log\_loss', 'neg\_mean\_absolute\_error', 'neg\_mean\_squared\_error', 'neg\_mean\_squared\_log\_error', 'neg\_median\_absolute\_error', 'normalized\_mutual\_info\_score', 'precision', 'precision\_macro', 'precision\_micro', 'precision\_samples', 'precision\_weighted', 'r2', 'recall', 'recall\_macro', 'recall\_micro', 'recall\_samples', 'recall\_weighted', 'roc\_auc', 'v\_measure\_score']
* Постройте графики сравнения результатов на обучающем и тестовом множестве с указанием метрики. Необходимо подобрать хотя бы две различные метрики.
* Выполните подбор гиперпараметров по методу случайного перебора. В пространство поиска включите следующие параметры: критерий, максимальная глубина, минимальное число объектов в листе, вес класса и минимальное количество объектов, необходимое для разделения внутреннего узла. Количество запусков не менее 200.
* Выведите параметры лучшей модели, наилучшее значение метрики и важность признаков.

3.6 Ансамблевые методы машинного обучения

**Реализация случайного леса и xgboost**

Решение задач при помощи случайного леса и xgboost.

**4. Нейронные сети**

4.2 Обучение нейронных сетей

**Алгоритм обучения нейронных сетей**

Нейрон с сигмоидой

Тестирование нейрона

Проверка forward\_pass()

Проверка backward\_pass()

Выполнение классификации объектов при помощи нейрона с сигмоидой

Обучение нейрона с помощью функции потерь LogLoss

Выполнение классификации объектов при помощи нейрона с LogLoss.

4.3 Глубокое обучение. Многослойные нейронные сети

**Запуск фреймворка PyTorch. Принципы работы и построения многослойных нейронных сетей**

PyTorch. Основы: синтаксис, torch.cuda и torch.autograd

Установка

Синтаксис

Базовые функции

Матричные операции

Перевод из NumPy в PyTorch

Использование технологии CUDA

Autograd

4.4 Фреймворк PyTorch. Реализация нейронов и сетей на PyTorch.

 **Реализация отдельных нейронов и многослойных нейронных сетей на PyTorch**

Один нейрон на PyTorch

Подбор параметров нейрона.

Многослойная сеть на PyTorch

Компоненты нейросети

Примеры построения многослойных нейросетей

**5. Современные методы ИИ и их применение**

5.1 Сверточные нейронные сети

**Построение сверточных нейронных сетей**

Операции Convolution и Pooling.

Свёрточные нейронные сети: MNIST

Задача классификации изображений

5.2 Рекуррентные нейронные сети

**Построение рекуррентных нейронных сетей**

Построение рекуррентных нейронных сетей на фреймворке PyTorch

5.3 Обработка естественного языка

**Построение нейронной сети для обработки естественного языка**

Библиотека Word2Vec. Принцип работы «мешка слов».

Библиотека gensim

Обучение модели

Дообучение модели

Word Similarity

t-distributed Stochastic Neighbor Embedding

5.4 Классификация изображений

**Построение нейронной сети для классификации изображений**

Свёрточные нейронные сети: CIFAR10

Предсказание классов изображений на датасете картинок - CIFAR10

**3. ОРГАНИЗАЦИОННО-ПЕДАГОГИЧЕСКИЕ УСЛОВИЯ РЕАЛИЗАЦИИ ПРОГРАММЫ**

**3.1 Кадровое обеспечение**

Кадровое обеспечение программы осуществляют сотрудники Института физико-математических наук и информационных технологий БФУ им. И. Канта. Реализация программы обеспечивается квалифицированными специалистами, имеющими высшее образование, опыт практической работы в области искусственного интеллекта.