I. Паспорт программы:

Специалист по машинному обучению

Автор программы: Пьяненкова Анна Владимировна.

Уровень сложности	Формат проведения	Время проведения	Возрастная категория	Доступность для участников с ОВЗ
ознакомительный	очная	30 минут	8-9, 10-11 классы	- Общие заболевания (нарушение дыхательной системы, пищеварительной, эндокринной систем, сердечно-сосудистой системы и т.д.; - Использование специальных образовательных программ и методов обучения и воспитания, специальных учебников, учебных пособий и дидактических материалов, специальных технических средств обучения коллективного и индивидуального пользования, предоставление услуг ассистента (помощника), оказывающего обучающимся необходимую техническую помощь не требуется - Возможность проведения пробы в смешанных группах «участники без ОВЗ + участники с ОВЗ»

II. Содержание программы

Введение (5 мин)

1. Краткое описание профессионального направления

Машинное обучение – это реализация анализа данных, не используя четких детерминированных алгоритмов. За последнее десятилетие машинное обучение было реализовано в беспилотных автомобилях, распознавании речи, эффективных поисковых системах и т.д. На данный момент машинное обучение прочно вошло в повседневную жизнь.

В рамках компетенции применяются наиболее эффективные алгоритмы машинного обучения, реализуется опыт их практического применения. Рассматривается применение машинного обучения к практическим новым задачам, требующим быстрого и эффективного решения. Компетенция охватывает следующие направления машинного обучения:

- *обучение с учителем;*
- *обучение без учителя;*
- *обучение с подкреплением;*
- *ансамблевые методы;*
- *нейронные сети и глубокое обучение.*

2. Место и перспективы профессионального направления в современной экономике региона, страны, мира

За последние 20 лет развитие компьютерных технологий шагнуло гораздо дальше представлений фантастов прошлого века. Навигатор строит маршрут лучше нас, приложения «угадывают» наши предпочтения в музыке. Системы искусственного интеллекта (далее – ИИ) преуспели в творчестве и областях, требующих от человека незаурядных способностей: обыгрывают профессиональных игроков в го и DOTA, обнаруживают раковые опухоли, точнее человека определяют личность по внешности или голосу и даже создают музыку и произведения искусства.

Однако эти системы еще не могут считаться ИИ. Они, как правило, хорошо справляются только с тем типом задач, на который натренированы. Например, антиспам-системы хоть и включают в себя анализ текста, ничего не знают об устройстве естественного языка и эмоциональной окраске сообщений. Кроме того, для машинного обучения все еще необходим человек: модель способна только на то, что в нее изначально заложили разработчики. Однако технологии не стоят на месте.

3. Необходимые навыки и знания для овладения профессией

Для овладения данной профессией необходимы навыки программирования, основы статистики, математического анализа, высшей математики.

4. 1-2 интересных факта о профессиональном направлении

Проект «Нейролирика» доцента Школы лингвистики НИУ ВШЭ Бориса Орехова вызывает тотальную фрустрацию у всех апологетов прекрасного. Дело в том, что он натренировал нейронную сеть на стихах великих поэтов — Пушкина, Гомера, Ахматовой и т.д., а потом заставил писать собственные вирши. Так ИИ создал некоторое количество опусов, поразительно напоминающих оригиналы.

Пример творчества нейросети, обученной на четырехстопных ямбах разных авторов:

Он беспощадной головой
Волной и волосом волненья
Не чувствовать не упадет.
В пределах воздух красный смех.

Сеть соблюдала все особенности стиля того или иного поэта. В итоге получился сборник стихов под названием «Нейролирика», о котором Орехов отозвался как о «легитимации нейронных стихов в литературе». На слух стихи нейронной сети от стихов истинных поэтов фокус-группа отличить не смогла.

5. Связь профессиональной пробы с реальной деятельностью

Использование машинного обучения для предсказания ситуации по исходным данным.

Постановка задачи (3 мин)

1. Постановка цели и задачи в рамках пробы

Цель: Построение модели машинного обучения для прогнозирования результата.

Задачи:

1. Визуализировать данные с использованием языка Python;
2. Обучить модель средствами scikit-learn;
3. Проверить модели на данных от пользователя.

2. Демонстрация итогового результата, продукта.

Нотебук с последовательностью команд, необходимых для выполнения поставленных задач.

Выполнение задания (15 мин)

1. Подробная инструкция по выполнению задания

Методическая разработка Приложение № 3.

2. Рекомендации для наставника по организации процесса выполнения задания
   - В начале пробы включить видеоролик «Машинное обучение и искусственный интеллект» (Приложение 1)
   - Подготовить данные для выполнения задания – файл с исходными данными train (Приложение 2). Разместить их в папку ресурсов на рабочих местах;
   - На рабочих местах запустить работу ПО Anaconda и в папку ресурсов разместить инструкцию по выполнению заданий (Приложение 4);
   - Подготовить волонтеров из числа студентов, провести с ними инструктаж;
   - Подготовить критерии успешного выполнения задания.

Контроль, оценка и рефлексия (7 мин)

1. Критерии успешного выполнения задания

   Все этапы машинного обучения выполнены в полном объеме:
   - программный код не выдает ошибок;
   - графики отображаются;
   - обученная модель выдает результат.

2. Рекомендации для наставника по контролю результата, процедуре оценки:

Проводить оценку процесса выполнения заданий согласно критериям успешного выполнения задания

3. Вопросы для рефлексии учащихся

   - На сколько просто происходило обучение модели?

6. Инфраструктурный лист

Наименование	Рекомендуемые технические характеристики с необходимыми примечаниями	Количество	на 1 чел.
Компьютер	Processor - Intel Core i7 Ethernet - 10/100/1000 mbps RAM - 8GB SSD 128 Gb	12	1
Клавиатура	Клавиатура без кнопки Power, подключение по USB	12	1
Мышь	подключение по USB	12	1
Монитор	Мониторы LCD 17"	24	2
Источник бесперебойного питания	Источник бесперебойного питания мощьностью от 600ВА	12	1
Сетевой фильтр 6 розеток, 5 метров		12	1
Патч-корд 2 м		12	1
Салфетки для чистки экранов и оргтехники		1	1
Anaconda	Anaconda	12	1
R Studio	R Studio	12	1

PyCharm	PyCharm	12	1

7. Приложение и дополнения

Литература

1. Бринк Х. Машинное обучение. — пер. с англ. Рузмайкина И. — Санкт-Петербург: Питер, 2017 — 336 с.

2. Рашка С. Python и машинное обучение - пер. с англ. А. В. Логунова. - М.: ДМК Пресс, 2017. - 418 с.: ил

Ссылка	Комментарий
https://www.coursera.org/learn/vvedenie-mashinnoe-obuchenie /home/welcome	Курс «Введение в машинное обучение» [Электронный ресурс]

Приложение № 1 – Видеоролик «Машинное обучение и искусственный интеллект» - 3 мин;

Приложение № 2 – файл с исходными данными;

Приложение № 3 – Презентация «Специалист по машинному обучению»

Приложение № 4 – Подробная инструкция выполнения задания

Приложение № 4. Инструкция по выполнению заданий

1. Подключение необходимых библиотек.

Для выполнения анализа данных, визуализации и обучения необходимо подключить библиотеки, которые обладают необходимым аппаратом.

В нотебуке наберите представленный на листинге код.

```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import numpy as np
from sklearn.model_selection import train_test_split
```

Для выполнения кода необходимо нажать на кнопку «Выполнить»

2. Подключение файла с данными

Файл с данными должен находиться в папке ресурсов.

В нотебуке наберите представленный на листинге код.

```
df=pd.read_csv("train.csv");
```

Для выполнения кода необходимо нажать на кнопку «Выполнить»

3. Просмотр данных

В нотебуке наберите представленный на листинге код.

```
df.head()
```

При выполнении данной команды будут выводиться данные из таблицы.

Для выполнения кода необходимо нажать на кнопку «Выполнить»

4. Визуализация данных

Анализ данных лучше проводить с использованием графиков и диаграмм.

a. Построим гистограмму, в которой определим количество мужчин и женщин каждого пола. Для построения гистограммы возьмем только часть данных – первые 20 строк.

В ячейке нотебука наберите представленный на листинге код.

```
df2=df.head(20)
df2.plot(kind='bar',x='Sex',y='Age')
plt.show()
```

Для выполнения кода необходимо нажать на кнопку «Выполнить»

b. Построим диаграмму рассеивания, показывающую зависимость выбора класса каюты от возраста пассажира.

В ячейке нотебука наберите представленный на листинге код.

```
df3=df.head(20)
df3.plot(kind='scatter',x='Pclass',y='Age',color='red')
plt.show()
```

c. Тепловая карта

Построим матрицу зависимости всех параметров. Используется математическое понятие Корреляция.

В ячейке нотебука наберите представленный на листинге код.

```
df3.corr(method='spearman')
```

Получим матрицу, где указывается проценты зависимости каждого параметра от другого. На основе такой таблицы делаются выводы о зависимостях.

Для выполнения кода необходимо нажать на кнопку «Выполнить»

Построим тепловую карту, которая графически представит зависимости между признаками.

В ячейке нотебука наберите представленный на листинге код.

```
corrmat = df3.corr()
f, ax = plt.subplots(figsize=(9,8))
sns.heatmap(corrmat, ax=ax, cmap="YlGnBu",linewidths=0.1)
```

Для выполнения кода необходимо нажать на кнопку «Выполнить»

По карте определяются самые зависимые друг от друга признаки – чем темнее или светлее ячейка, то зависимость сильнее.

5. Подготовка данных к обучению.

Скопируйте код и вставьте в ячейку нотебука и выполнить.

```
from sklearn.model_selection import train_test_split #import split of data
 set method
```

```
df.Age.fillna(df.Age.mode()[0], inplace = True) # Fulling with mode
df.Cabin.fillna(df.Cabin.mode()[0], inplace = True)
df.Embarked.fillna(df.Embarked.mode()[0], inplace = True)

df.Age.fillna('nothing_here', inplace = True) # Fulling with mode
df.Cabin.fillna('nothing_here', inplace = True)
df.Embarked.fillna('nothing_here', inplace = True)
```

```
df.Age.fillna(df.Age.mode()[0], inplace = True) # Fulling with mode
df.Cabin.fillna(df.Cabin.mode()[0], inplace = True)
df.Embarked.fillna(df.Embarked.mode()[0], inplace = True)

target = df['Survived'] # Y
data = df.drop('Survived', axis = 1) # X
```

```
from sklearn.preprocessing import LabelEncoder #Encoder for str

le = LabelEncoder() # Initialization of Encoder

data['Sex'] = le.fit_transform(data['Sex']) # Encode of of columns where w
e can find str
data['Name'] = le.fit_transform(data['Name'])
data['Ticket'] = le.fit_transform(data['Ticket'])
data['Cabin'] = le.fit_transform(data['Cabin'])
data['Embarked'] = le.fit_transform(data['Embarked'])
```

6. Разбиение выборки на обучающую и тестовую.

Для обучения модели необходимо разбить все данные на две части – одна будет использоваться для обучения, другая для тестирования. Деление обычно выполняется на 80 и 20 процентов соответственно.

Скопируйте код и вставьте в ячейку нотебука и выполнить.

```
from sklearn.model_selection import train_test_split #import split of data
 set method

```

```
x_train, x_test, y_train, y_test = train_test_split(data, target, test_siz
e = 0.2, random_state = 0)

x_train=x_train.drop('Embarked', axis = 1)
x_test=x_test.drop('Embarked', axis = 1)
print(x_train)
```

7. Обучение модели.

Обучение модели происходит с использованием алгоритма «Случайный лес».

Скопируйте код и вставьте в ячейку нотебука и выполнить.

```
from sklearn.ensemble import RandomForestClassifier
RFC = RandomForestClassifier()
RFC.fit(x_train, y_train)
```

8. Проверка точности модели обучения.

Скопируйте код и вставьте в ячейку нотебука и выполнить.

```
from sklearn.metrics import accuracy_score #method for

pred_RFC_train = RFC.predict(x_train) #prediction for train data
pred_RFC_test = RFC.predict(x_test)
 #prediction for test data

print('Правильность на обучающем наборе:', np.round(accuracy_score(y_train
, pred_RFC_train), 2)) #Printing our acc score
print('Правильность на тестовом наборе:', np.round(accuracy_score(y_test,
pred_RFC_test), 2))
```

В результате получили процент правильности обучения модели как на тестовом так и на обучающем наборе.

9. Проверка модели на данных от пользователя.

Установим значения для признаков самостоятельно и посмотрим, выживет ли наш пассажир с заданными параметрами или нет.

Скопируйте код и вставьте в ячейку нотебука и выполнить.

```
me = {'Age': 25.0, # looking mine encoded  params from last position of da
ta
 'Cabin': 73.0,
 'Fare': 300.0,
 'Name': 563.0,
 'Parch': 0.0,
 'Pclass': 1.0,
 'Sex': 1.0,
 'SibSp': 1.0,
 'Ticket': 610.0,
     'PassengerId':58585}
me =  pd.DataFrame(data = [me])
prediction = RFC.predict(me)
print(prediction)
```

10. Вывод

Мы построили модель, обучили и проверили на других данных работу модели.