Московский Авиационный Институт

(Национальный Исследовательский Университет)

Кафедра 806 «Вычислительная информатика и программирование»

Факультет: «Информационные технологии и прикладная математика»

Реферат

Дисциплина: «Вычислительная системы»

I семестр

Реферат: «Изучение языка программирования Python на примере решения задачи с использованием алгоритмов машинного обучения»

|  |  |
| --- | --- |
| Группа: | М8О-107Б-20 |
| Студент: | Чекменев Вячеслав Алексеевич |
| Преподаватель: | Найденов Иван Евгеньевич |
| Оценка: |  |
| Дата: | 25.12.2020 |

Москва, 2020

**Содержание**

1.Задачи……………………………………………………..………....2

2.Введение. Выбор утилиты для написания программы. Использованные библиотеки…………………………...…………….3

3.Использованные классы. Сценарий работы программы……….....4

4.Заключени…………………………...………………………………13

5.Список используемых источников………………………………...17

Задача  
  
 Нужно с хорошей точностью определить пользователей курса, которые точно уйдут или точно закончат курс.  
 Подробнее про курс: В рамках данного курса подробно рассматриваются все основные этапы анализа данных при помощи R. В данном курсе рассматриваются как стандартные методы R и Rstudio, так и специальные пакеты и библиотеки. На курсе применяются основные методы статистического анализа: t-тест, корреляция, регрессия, дисперсионный и регрессионный анализ и др. Особое внимание в курсе уделяется визуализации получаемых результатов.

**Введение**

Итак, **Машинное обучение** — класс методов [искусственного интеллекта](https://ru.wikipedia.org/wiki/%D0%98%D1%81%D0%BA%D1%83%D1%81%D1%81%D1%82%D0%B2%D0%B5%D0%BD%D0%BD%D1%8B%D0%B9_%D0%B8%D0%BD%D1%82%D0%B5%D0%BB%D0%BB%D0%B5%D0%BA%D1%82), характерной чертой которых является не прямое решение задачи, а обучение в процессе применения решений множества сходных задач. Для построения таких методов используются средства [математической статистики](https://ru.wikipedia.org/wiki/%D0%9C%D0%B0%D1%82%D0%B5%D0%BC%D0%B0%D1%82%D0%B8%D1%87%D0%B5%D1%81%D0%BA%D0%B0%D1%8F_%D1%81%D1%82%D0%B0%D1%82%D0%B8%D1%81%D1%82%D0%B8%D0%BA%D0%B0), [численных методов](https://ru.wikipedia.org/wiki/%D0%A7%D0%B8%D1%81%D0%BB%D0%B5%D0%BD%D0%BD%D1%8B%D0%B5_%D0%BC%D0%B5%D1%82%D0%BE%D0%B4%D1%8B), [математического анализа](https://ru.wikipedia.org/wiki/%D0%9C%D0%B0%D1%82%D0%B5%D0%BC%D0%B0%D1%82%D0%B8%D1%87%D0%B5%D1%81%D0%BA%D0%B8%D0%B9_%D0%B0%D0%BD%D0%B0%D0%BB%D0%B8%D0%B7), [методов оптимизации](https://ru.wikipedia.org/wiki/%D0%9E%D0%BF%D1%82%D0%B8%D0%BC%D0%B8%D0%B7%D0%B0%D1%86%D0%B8%D1%8F_(%D0%BC%D0%B0%D1%82%D0%B5%D0%BC%D0%B0%D1%82%D0%B8%D0%BA%D0%B0)), [теории вероятностей](https://ru.wikipedia.org/wiki/%D0%A2%D0%B5%D0%BE%D1%80%D0%B8%D1%8F_%D0%B2%D0%B5%D1%80%D0%BE%D1%8F%D1%82%D0%BD%D0%BE%D1%81%D1%82%D0%B5%D0%B9), [теории графов](https://ru.wikipedia.org/wiki/%D0%A2%D0%B5%D0%BE%D1%80%D0%B8%D1%8F_%D0%B3%D1%80%D0%B0%D1%84%D0%BE%D0%B2), различные техники работы с [данными в цифровой форме](https://ru.wikipedia.org/wiki/%D0%94%D0%B0%D0%BD%D0%BD%D1%8B%D0%B5_(%D0%B2%D1%8B%D1%87%D0%B8%D1%81%D0%BB%D0%B8%D1%82%D0%B5%D0%BB%D1%8C%D0%BD%D0%B0%D1%8F_%D1%82%D0%B5%D1%85%D0%BD%D0%B8%D0%BA%D0%B0))

**Выбор утилит для написания программы.**

Для написания нашей программы мы использовали интерактивную среду разработки Jupyter notebook.

**Использованные библиотеки.**

import pandas as pd

- для предобработки данных

import seaborn as sns

- для построения графиков

import numpy as np

- для работы с массивами и математическими функциями

%matplotlib inline

- для отображения графиков в Jupyter notebook

**Использованные классы.**

from sklearn import tree

- для создания классификатора и дерева решений

from sklearn.model\_selection import train\_test\_split

- разделение данных на тестовое и тренировочное множества

from sklearn.model\_selection import GridSearchCV

- для поиска лучших параметров для дерева решений

from sklearn.metrics import precision\_score, recall\_score

- для вывода значений точности и полноты модели

from IPython.display import SVG

from graphviz import Source

from IPython.display import display

- для вывода дерева в формате SVG Jupyter notebook

from IPython.display import HTML

style = "<style>svg{width:100% !important;height:100% !important;}</style>"

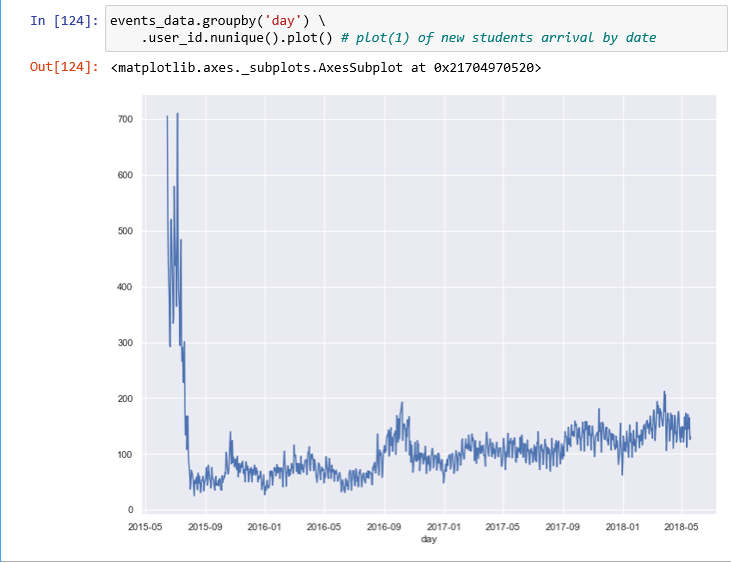
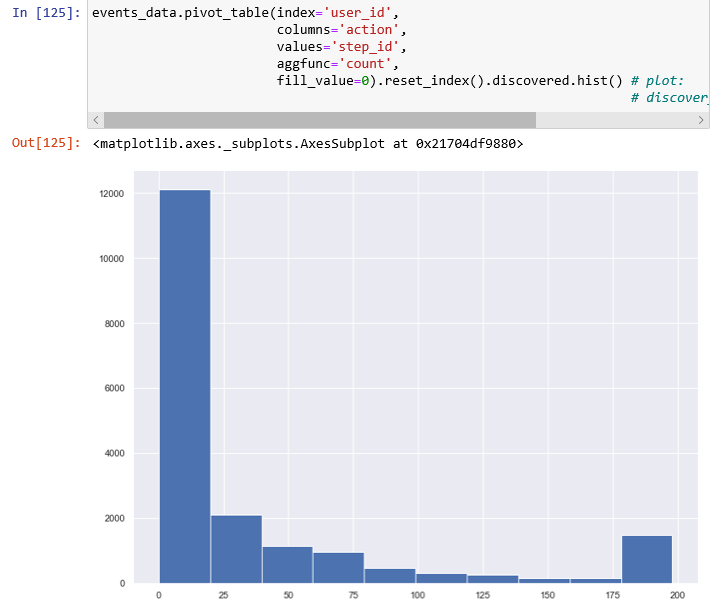
HTML(style)

**Сценарий работы программы**

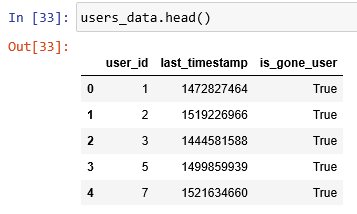
1. Перевод времени из Unix date в нормальный вид.

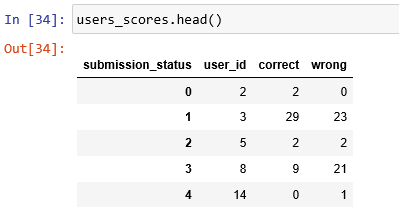


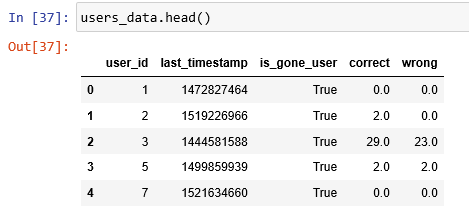
1. Предварительная проверка данных на корректность при. помощи построения графиков: зависимости кол-ва прибывших пользователей от даты, зависимости кол-ва действий от даты.

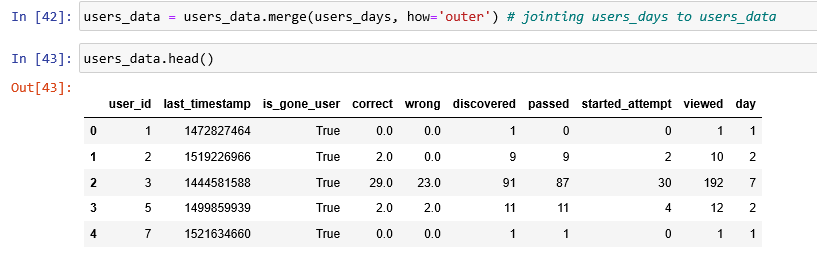


1. Будем считать человека ушедшим с курса, если он не совершал действий 15 дней.
2. Создаем датафрейм users\_data (последний заход на курс, ушел ли человек с курса), users\_scores (Кол-во удачных попыток и неудачных). Объединяем эти датафреймы в один (users\_data).



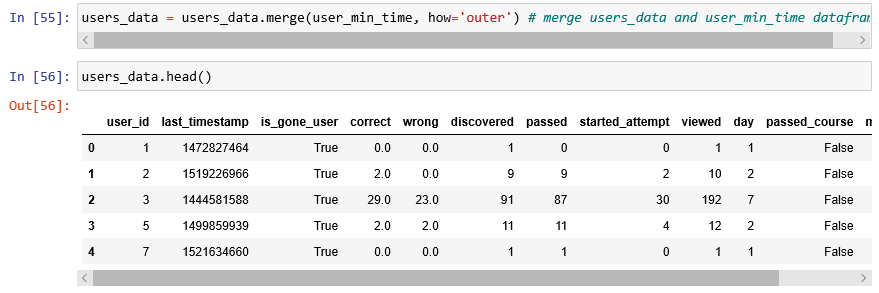




1. Присоединяем к датафрейму users\_datа датафрейм с кол-вом уникальных дней проведенных на курсе по общему столбцу user\_id.
2. Определяем условие прохождения курса (необходимо набрать больше 170 баллов).



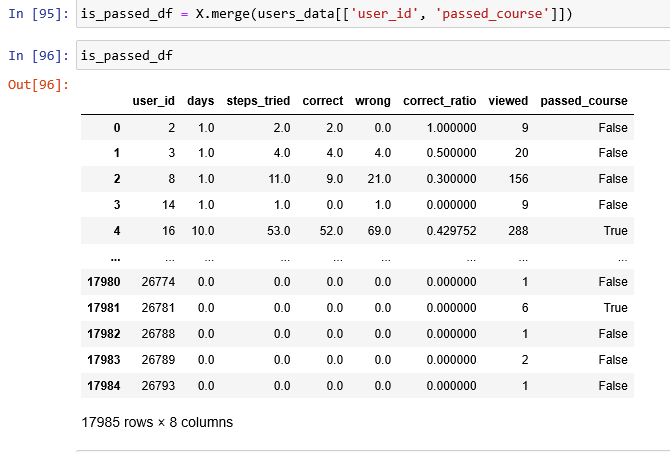
1. Присоединяем к датафрейму users\_datа датафрейм с временем первого посещения курса для каждого пользователя по общему столбцу user\_id.



1. Будем предсказывать уход человека с курса по первым 15 дням пребывания.
2. Найдем пороговое значение времени для каждого пользователя.
3. Отберем действия, которые совершили пользователи до своих пороговых значений времени.
4. Составив датафрейм для последующего создания модели is\_passed\_df включающий в себя: кол-во уникальных дней проведенных за определенный ранее промежуток времени (15 дней), кол-во пройденых степов, кол-во удачных и неудачных попыток и точность выполнения заданий.

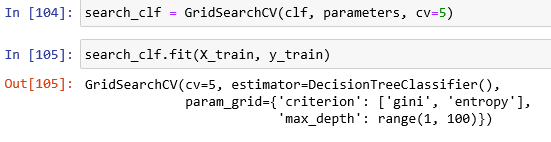


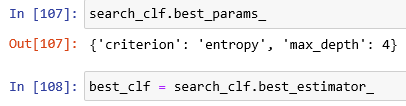


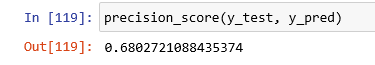
1. Присоединяем к датафрейму is\_passed\_df колонку passed\_course (значения 0 1).
2. Составим серию со значениями целевой переменной и датафрейма с фичами.



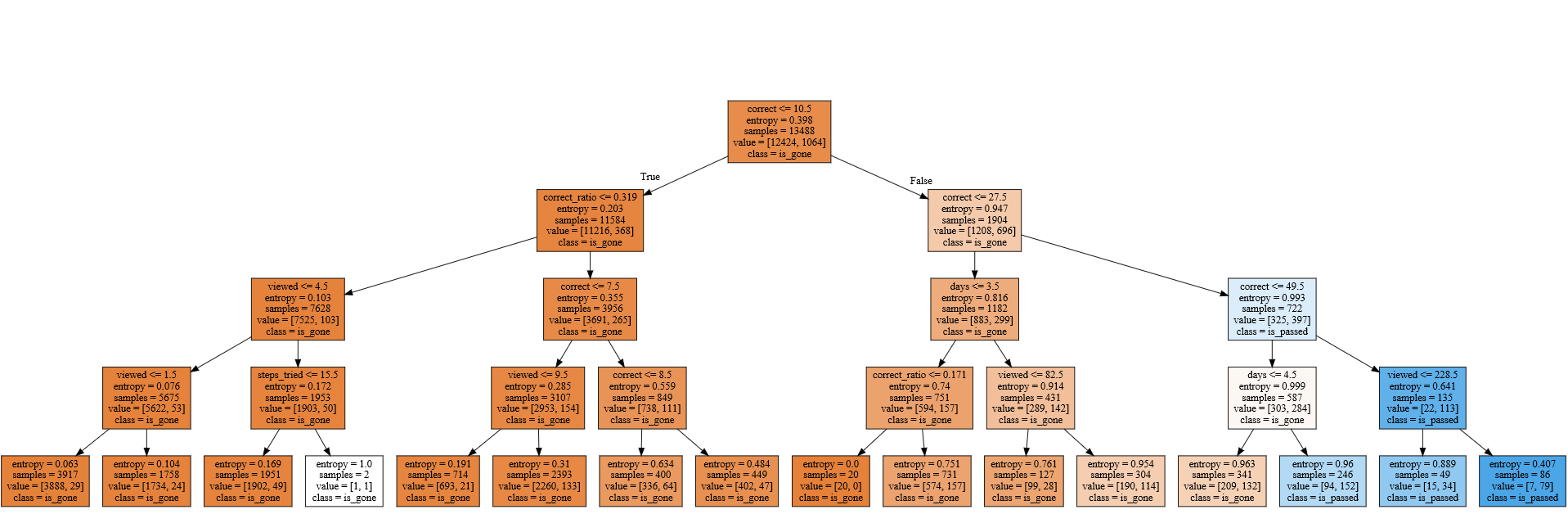
1. Найдем нужные параметры для дерева решений при помощи GridSearchCV.



1. Найдем дерево решений по наилучшим параметрам.
2. Найдем Результат предсказания на тестовой выборке.
3. Найдем точность (precision) нашего предсказания



1. Дерево решений ( <https://raw.githubusercontent.com/Suraba03/report_sem_1/66d0e268a0d9019b45202a54bf3815f96785164e/tree.svg> )



**Заключение**

Для реализации данной задачи нам нужно было познакомиться с языком пайтон, так как данный язык хорошо подходит для машинного обучения и в целом для Data Science.

Для реализации и понимания алгоритмов машинного обучения мы ознакомились с основами статистического анализа. Все нужные знания мы получали из MOOC курсов на платформе stepik.

**Список и информация о курсах**:

* *Основы статистики часть 1* - В рамках курса рассматриваются подходы к описанию получаемых в исследованиях данных, основные методы и принципы статистического анализа, интерпретация и визуализация получаемых результатов. Рассматриваются такие методы дисперсионный, регрессионный и кластерный анализ. Ставятся задачи сравнивения групп между собой, рассчета коэффициентов корреляции и построения регрессионных уравнений.  
  Основной акцент на курсе был сделан на математических идеях, интуиции и логике, которые обуславливают методы и расчетные формулы. Изученный материал применим для решения задач на машинное обучение и не только.
* *Основы статистики часть 2 -* В данном курсе уже рассмотрены методы, которые используются в анализе данных и наиболее часто применяются при статистической обработке результатов в широчайшем круге научных и прикладных областей. Помимо теоретических заданий были решены практические задачи, которые помогли для более глубокого понимания темы. Знаний, полученных в результате прохождения данного курса достаточно, чтобы научиться более быстро и эффективно решать различные задачи, связанные с анализом данных.
* *Курс по пайтону -* В курсе по python было много практических задач на самые базовые понятия о языке, такие как: операторы, переменные, типы данных, условия, циклы, строки, списки, функции словари, интерпретатор, файлы, модули.
* *Курс по Data Science -* Курс является ознакомительным относительно основ машинного обучения. Мы подробно разобрали основные теоретические понятия, а также начали знакомство с библиотеками Pandas и Scikit-learn — наиболее популярными инструменатами для анализа данных и машинного обучения, используя язык программирования Python.

В итоге данная задача оказалась очень интересной и охватила большое кол-во новых для нас тем. Эта работа дала нам хороший старт в изучении машинного обучения и анализа данных. Мы продолжим изучать машинное обучение и дальше, чтобы охватить более узкие темы, которые пригодятся нам для тюнинга нынешней модели.

**Список используемых источников**

1. Основы статистики - [https://stepik.org/course/76](https://vk.com/away.php?utf=1&to=https%3A%2F%2Fstepik.org%2Fcourse%2F76)
2. Основы статистики 2 - [https://stepik.org/course/524](https://vk.com/away.php?utf=1&to=https%3A%2F%2Fstepik.org%2Fcourse%2F524)
3. Программирование на Python - [https://stepik.org/course/67](https://vk.com/away.php?utf=1&to=https%3A%2F%2Fstepik.org%2Fcourse%2F67)
4. Введение в Data Science и машинное обучение - [https://stepik.org/course/4852/syllabus](https://vk.com/away.php?utf=1&to=https%3A%2F%2Fstepik.org%2Fcourse%2F4852%2Fsyllabus)
5. Брали от туда ответы на наши вопросы - [https://stackoverflow.com](https://vk.com/away.php?utf=1&to=https%3A%2F%2Fstackoverflow.com)
6. Документация по pandas – [https://pandas.pydata.org/pandas-docs/stable/referenc..](https://vk.com/away.php?to=https%3A%2F%2Fpandas.pydata.org%2Fpandas-docs%2Fstable%2Freference%2Fgeneral_functions.html&cc_key=)
7. Документация по scikit-learn - [https://scikit-learn.org/stable/ljrevtynfw](https://vk.com/away.php?to=https%3A%2F%2Fscikit-learn.org%2Fstable%2Fljrevtynfw&cc_key=)