|  |  |
| --- | --- |
| Gerb-BMSTU_01 | **Министерство науки и высшего образования Российской Федерации**  **Федеральное государственное бюджетное образовательное учреждение**  **высшего образования**  **«Московский государственный технический университет**  **имени Н.Э. Баумана**  **(национальный исследовательский университет)»**  **(МГТУ им. Н.Э. Баумана)** |

ФАКУЛЬТЕТ Информатика и системы управления

КАФЕДРА \_\_ Системы обработки информации и управления

**РАСЧЕТНО-ПОЯСНИТЕЛЬНАЯ ЗАПИСКА**

***К НАУЧНО-ИССЛЕДОВАТЕЛЬСКОЙ РАБОТЕ***

***НА ТЕМУ:***

***Использование алгоритма XGBoost для предсказания завершения курса обучающимся\_\_\_\_\_\_\_\_\_\_\_\_\_\_***

***\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_***

***\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_***

***\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_***

***\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_***

***\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_***

Студент \_\_\_\_ИУ5-31м\_\_\_\_ **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_**Д.А.Макаров**\_\_\_**

(Группа) (Подпись, дата) (И.О.Фамилия)

Руководитель **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_**Ю.Е.Гапанюк**\_\_\_\_**

(Подпись, дата) (И.О.Фамилия)

*2020 г.*

Оглавление

[Введение 3](#_Toc57489492)

[Исследование датасета 4](#_Toc57489493)

[Подготовка данных для машинного обучения 15](#_Toc57489494)

[Библиотека XGBoost 17](#_Toc57489495)

[Список гиперпараметров XGBoost 18](#_Toc57489496)

[Визуализация деревьев XGBoost 20](#_Toc57489497)

[Вывод 22](#_Toc57489498)

[Список использованных источников 23](#_Toc57489499)

# **Введение**

В настоящее время большой популярностью пользуются различные онлайн-курсы. Однако, распространена ситуация, когда пользователь бросает прохождение курса и не получает сертификат. В данной курсовой работе будет использован датасет, основанный на реальных данных одной из платформ онлайн курсов. Будет произведена очистка данных и их визуализация, а также анализ. С использованием библиотеки XGBoost будет произведена попытка создания модели, которая будет предсказывать, завершит ли пользователь более 50% курса или нет.

# **Исследование датасета**

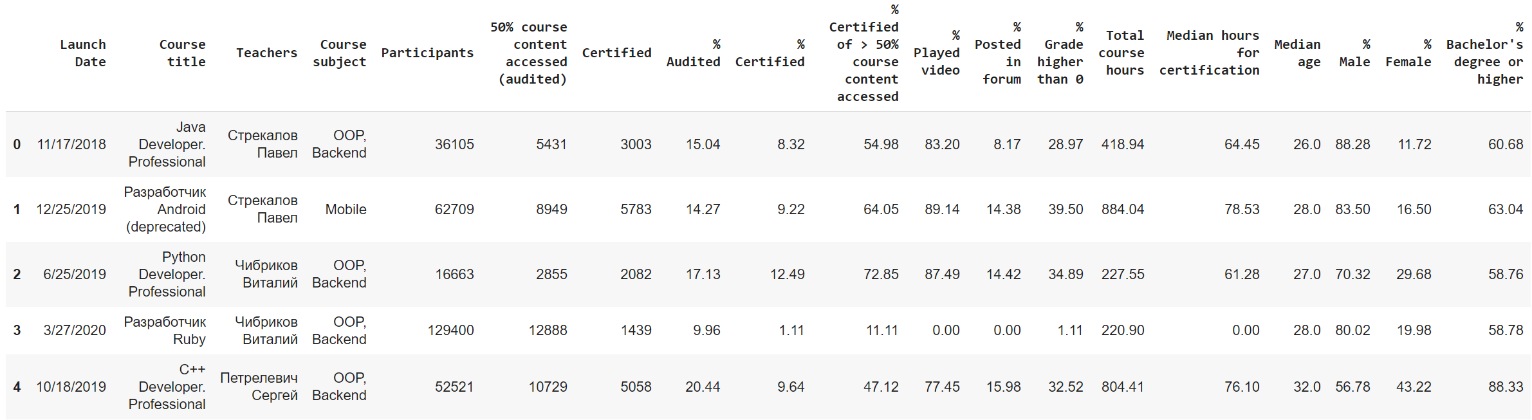
Импорт библиотек

import pandas as pd  
import numpy as np  
import matplotlib.pyplot as plt  
%matplotlib inline  
import seaborn as sns

Будет использована библиотека XGBoost для прогнозирования студентов, которые прошли более 50% курса

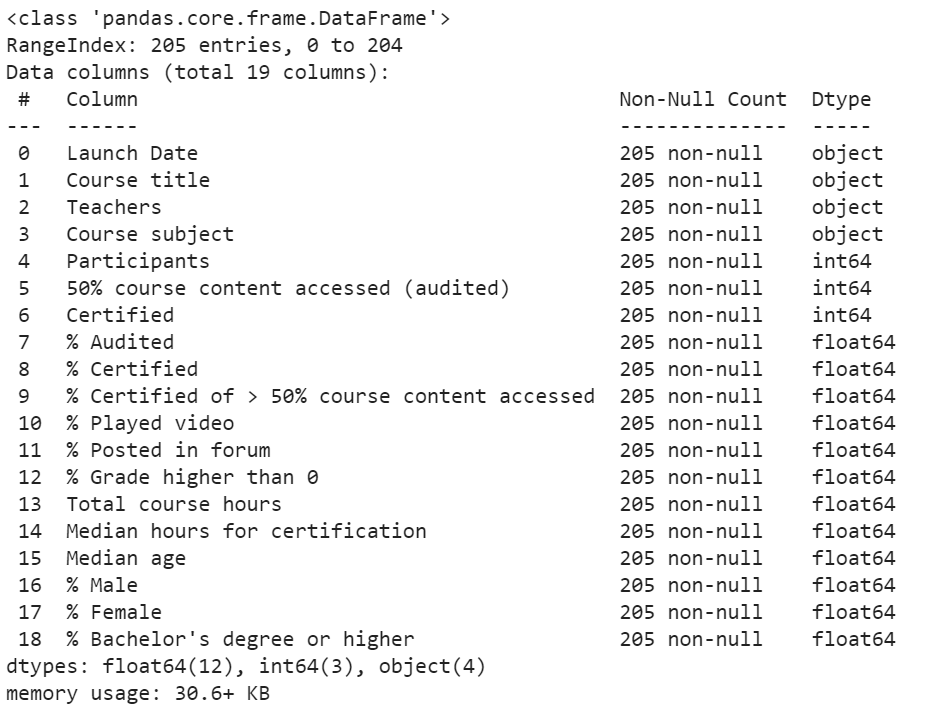
Импорт данных из csv файла

df\_course= pd.read\_csv('/Users/denis/Downloads/3.csv')  
df\_course.head()



Просмотр типов данных. Данные представлены типами object, int, float

df\_course.info()



Список категорий курсов и количество курсов в каждой категории

df\_course['Course subject'].value\_counts()

OOP, Backend 50  
Management 43  
Test 20  
Administration 19  
Security 16  
ML 15  
Mobile 11  
DB 11  
DevOps 7  
Frontend 6  
Math 5  
Design 2  
Name: Course subject, dtype: int64

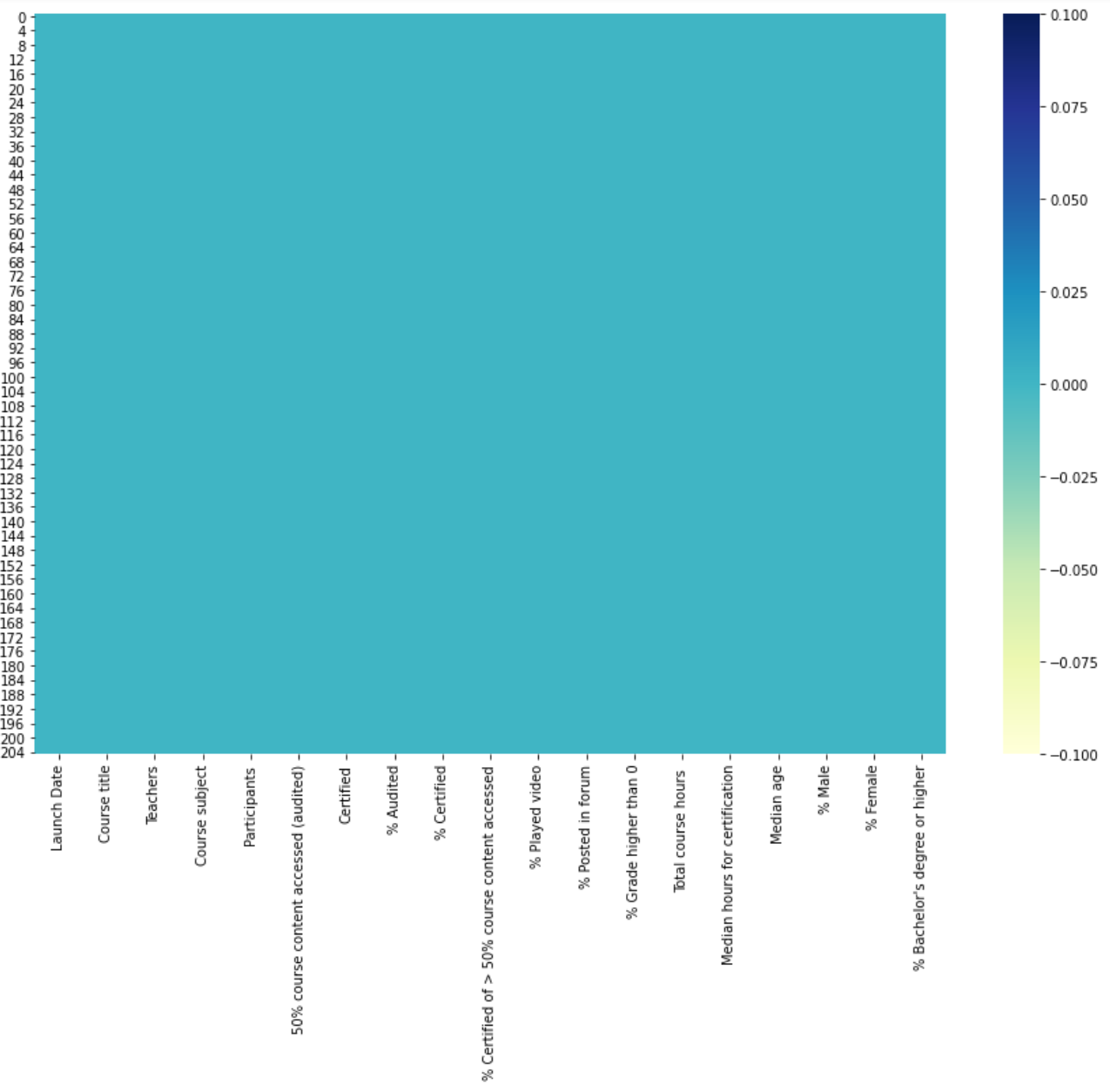
Список преподавателей и количество курсов, на которых он преподает

df\_course['Teachers'].value\_counts()

Волосатов Евгений 4  
Темирханова Эльвира 3  
Петрелевич Сергей 3  
Дроздецкий Владимир 3  
Цыкунов Алексей 3  
 ..  
Гуторов Владимир 1  
Левчук Мартин 1  
Курочкин Игорь 1  
Пулявин Артем 1  
Швец Олег 1

Просмотр значений NULL в данных, установленных с помощью тепловой карты:

plt.figure(figsize=(15,10))  
sns.heatmap(df\_course.isnull(),cmap="YlGnBu")



Мы убедились, что пустых ячеек нет.

**Визуализация данных**

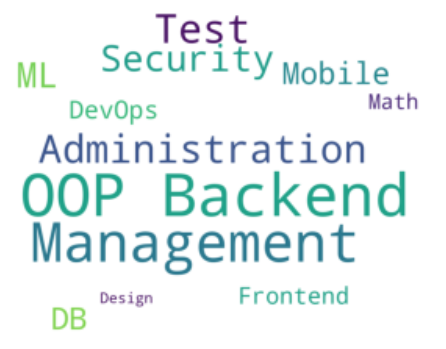
Представим названия курсов в виде облака слов, где более часто используемые слова имеют больший размер, чем другие.

from wordcloud import WordCloud, STOPWORDS  
  
wordcloud = WordCloud(  
 stopwords=STOPWORDS,  
 background\_color='white',  
 width=1200,  
 height=1000  
 ).generate(" ".join(df\_course['Course title']))  
  
  
plt.imshow(wordcloud)  
plt.axis('off')  
plt.show()



Отобразим категории курсов в виде облака слов.

wordcloud = WordCloud(  
 stopwords=STOPWORDS,  
 background\_color='white',  
 width=1200,  
 height=1000  
 ).generate(" ".join(df\_course['Course subject']))  
  
  
plt.imshow(wordcloud)  
plt.axis('off')  
plt.show()

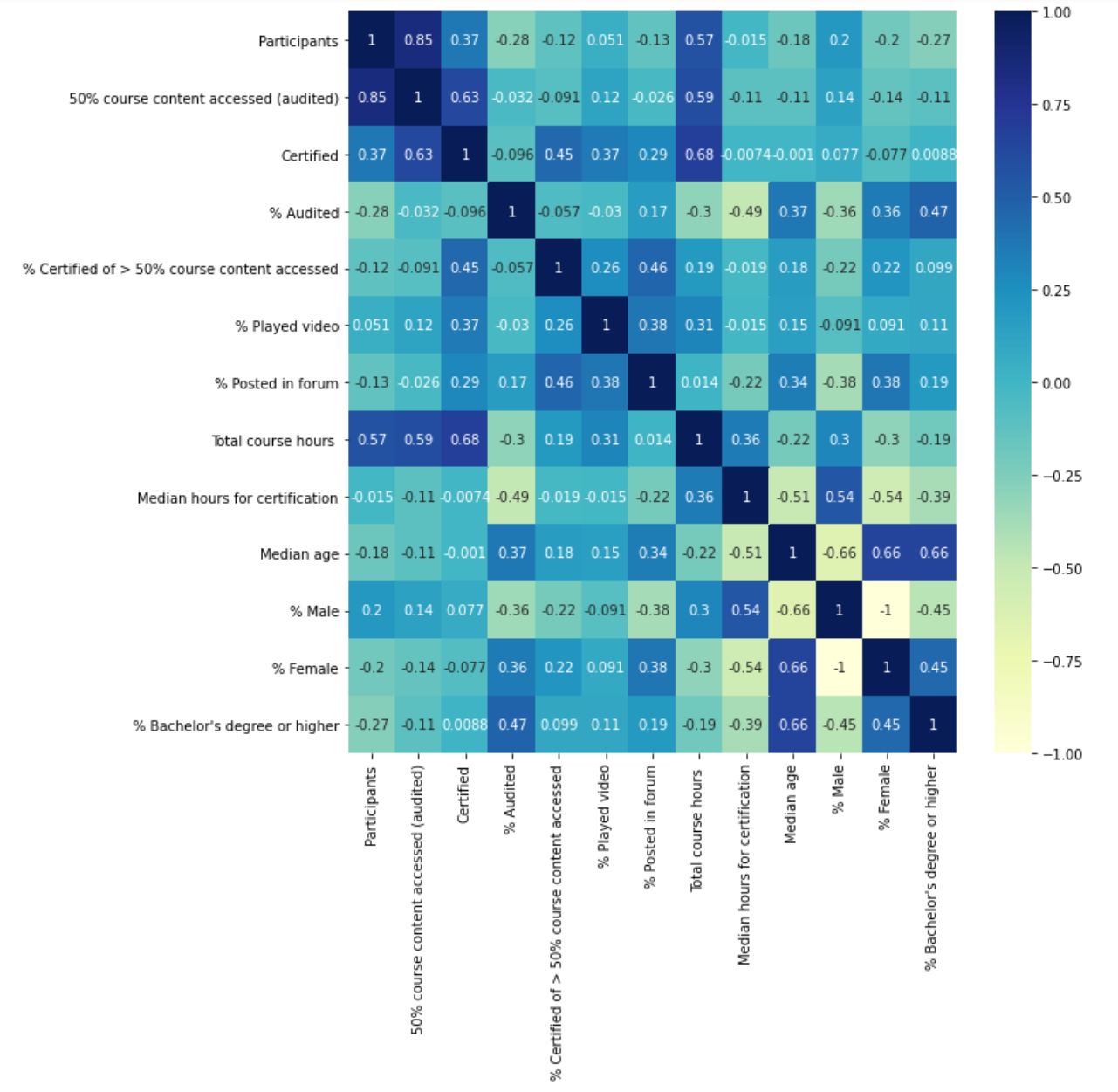


Судя по облакам слов, в названии курса чаще всего встречается слово «разработчик», а самой популярной категорией является бэкенд разработка и объектно-ориентированное программирование.

Построим матрицу корреляций между различными признаками.

df\_course=df\_course.drop(['% Certified','Course title','% Grade higher than 0'],axis=1)  
df\_course

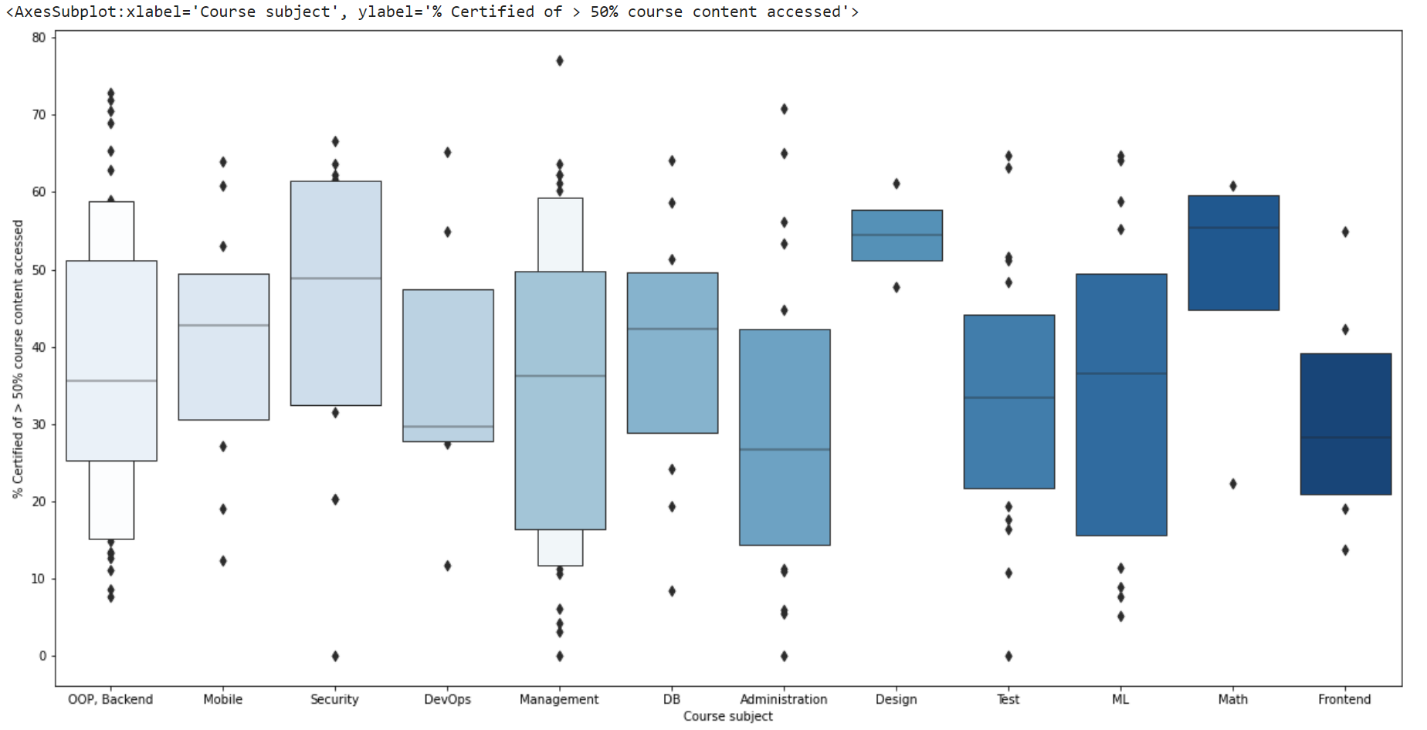
figure= plt.figure(figsize=(10,10))  
sns.heatmap(df\_course.corr(), annot=True,cmap="YlGnBu")

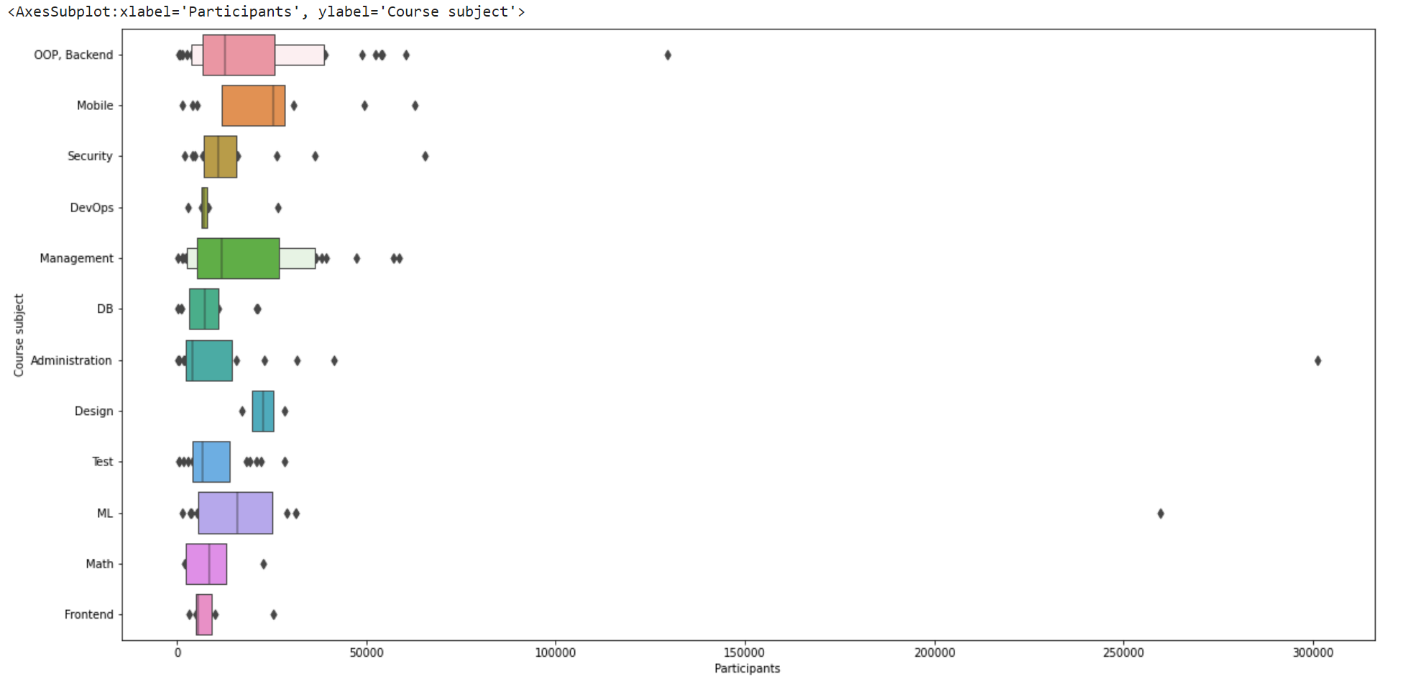


Как видно из матрицы, сильной корреляции между признаками нет, будем   
продолжать исследование дальше.

Построим диаграммы размаха («ящик с усами») для признаков Course subject, Certified   
of > 50% course content accessed и Participants, Course subject

figure= plt.figure(figsize=(20,10))  
sns.boxenplot(x='Course subject',y='% Certified of > 50% course content accessed,data=df\_course,palette="Blues")

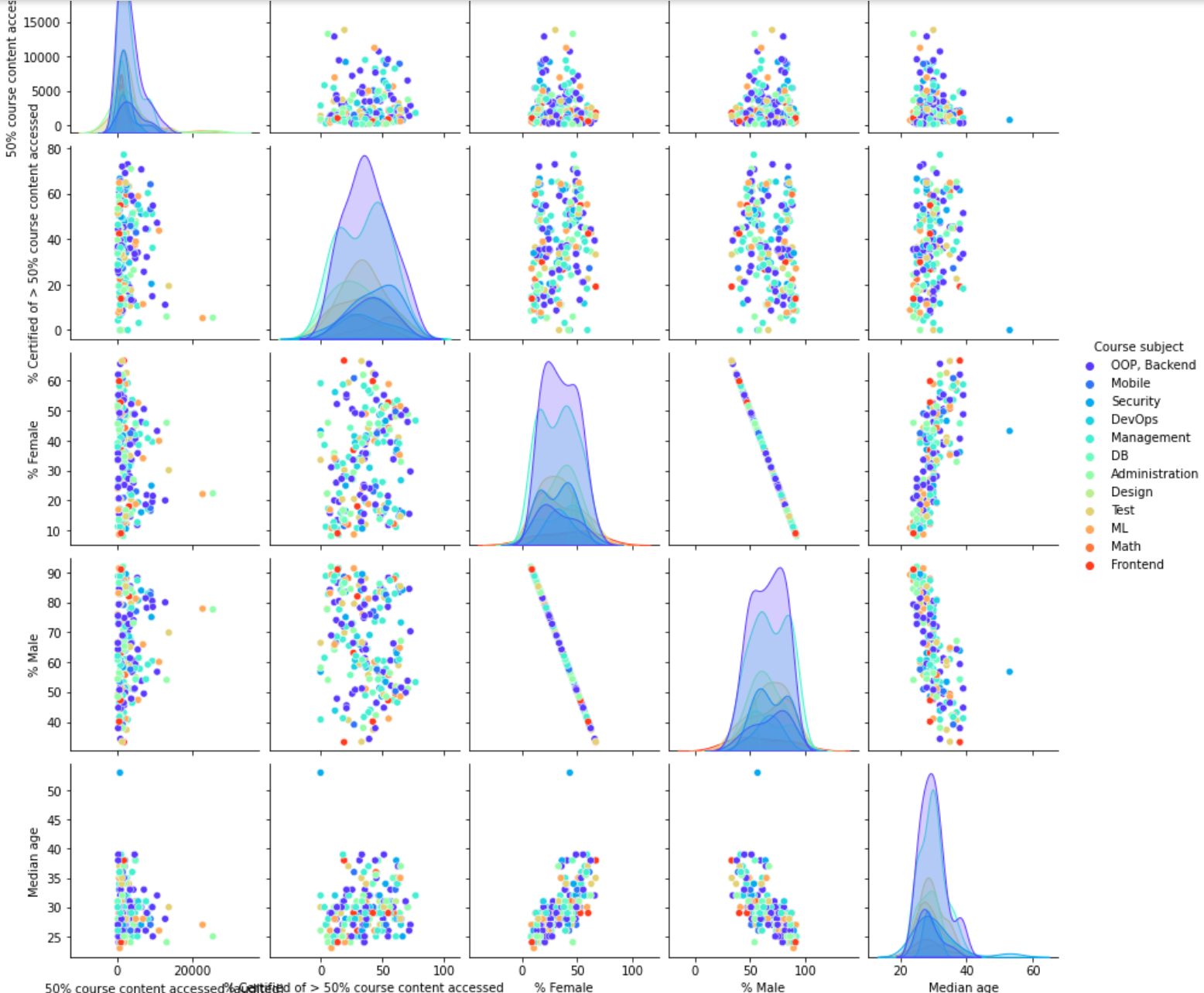
figure= plt.figure(figsize=(20,10))  
sns.boxenplot('Participants','Course subject',data=df\_course)



Данные графики в удобной форме показывают медиану (или, если нужно, среднее), нижний и верхний квартили, минимальное и максимальное значение выборки и выбросы. Расстояния между различными частями ящика позволяют определить степень разброса (дисперсии) и асимметрии данных и выявить выбросы.

График типа pairplot показывает отношения между всеми парами переменных.

df\_pairplot\_cols=df\_course[['Course subject','50% course content accessed (audited)','% Certified of > 50% course content accessed','% Female','% Male','Median age']]  
plt.figure(figsize=(20,20))  
sns.pairplot(df\_pairplot\_cols,hue='Course subject',palette='rainbow')

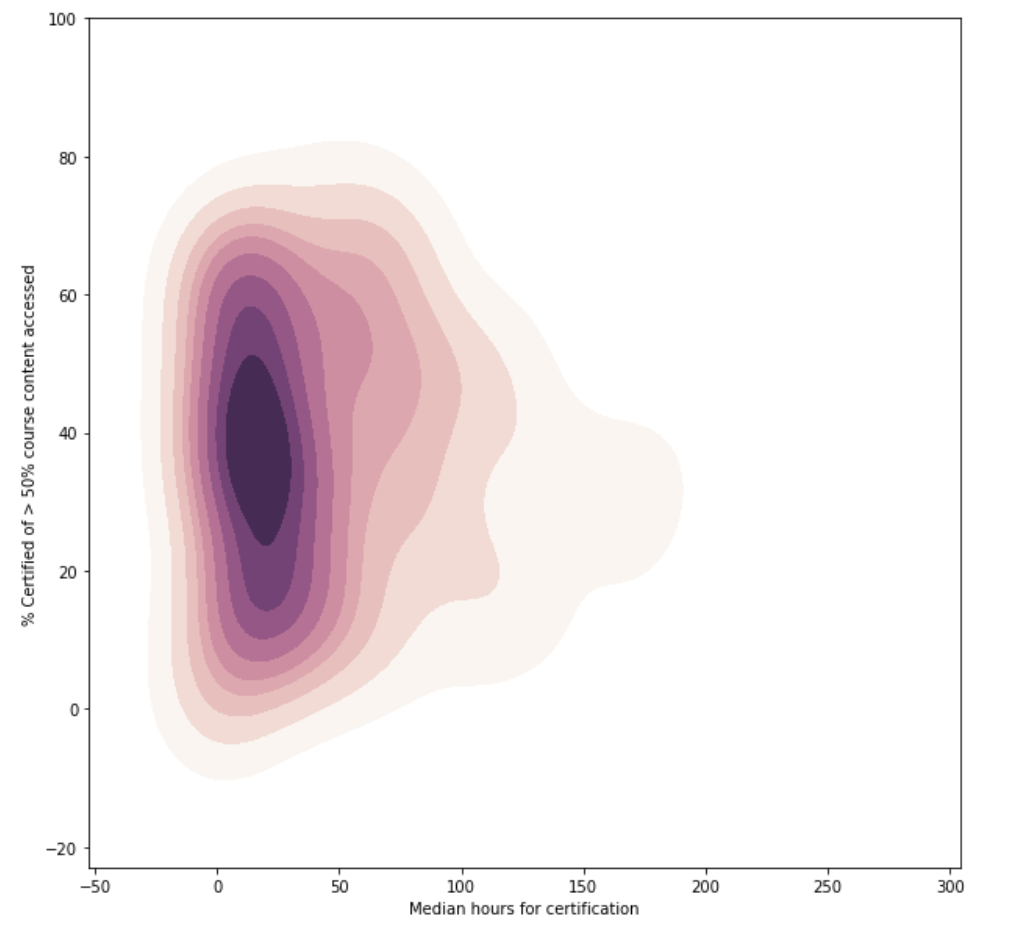


Построим kdeplot для Median hours for certification и % Certified of > 50% course content accessed.

График оценки плотности ядра (KDE) - это метод визуализации распределения наблюдений в наборе данных, аналогичный гистограмме. KDE представляет данные с помощью непрерывной кривой плотности вероятности в одном или нескольких измерениях.

По сравнению с гистограммой KDE может создавать график, который менее загроможден и более понятен, особенно при рисовании нескольких распределений. Но он может внести искажения, если основное распределение ограничено или негладко. Как и в случае гистограммы, качество представления также зависит от выбора хороших параметров сглаживания.

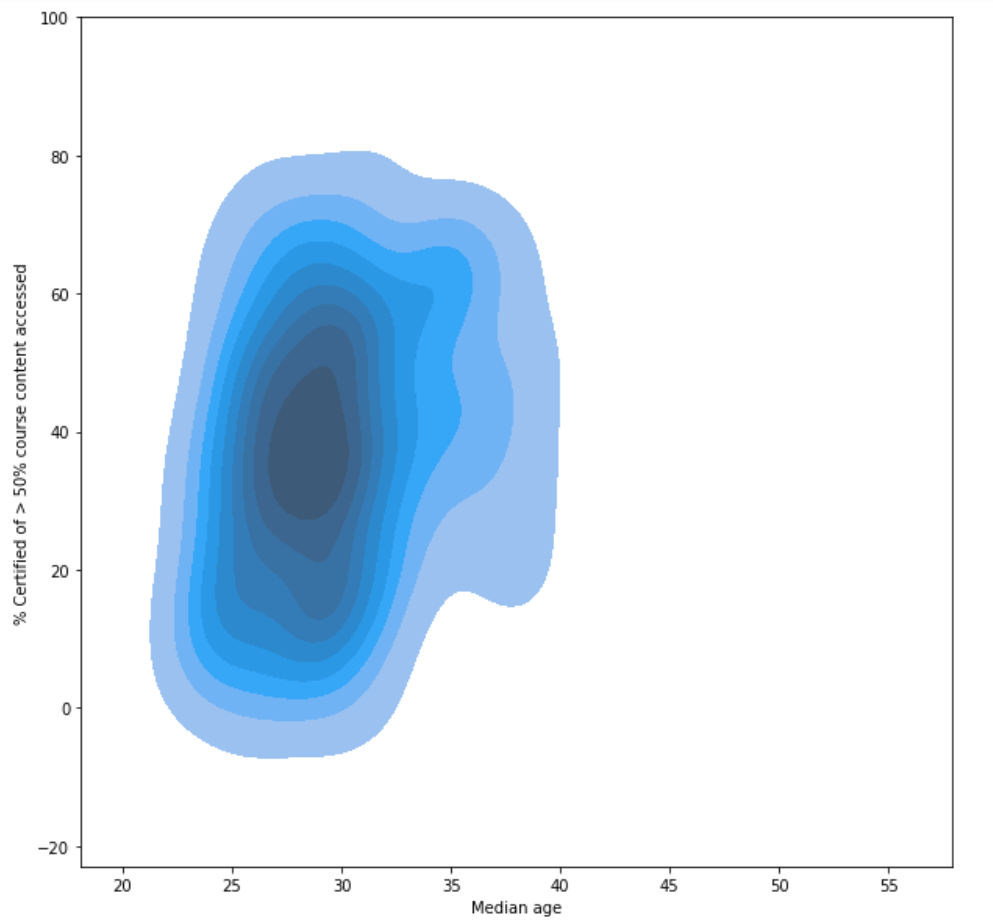
x= df\_course['Median hours for certification']  
y= df\_course['% Certified of > 50% course content accessed']  
cmap = sns.cubehelix\_palette(light=1, as\_cmap=True)  
plt.figure(figsize=(10,10))  
sns.kdeplot(x, y, cmap=cmap, shade=True);



Построим kdeplot для Median age и % Certified of > 50% course content accessed

По графику видно, что среднее время для получения сертификата составляет 25 часов?

x= df\_course['Median age']  
y= df\_course['% Certified of > 50% course content accessed']  
plt.figure(figsize=(10,10))  
sns.kdeplot(x, y, shade=True);

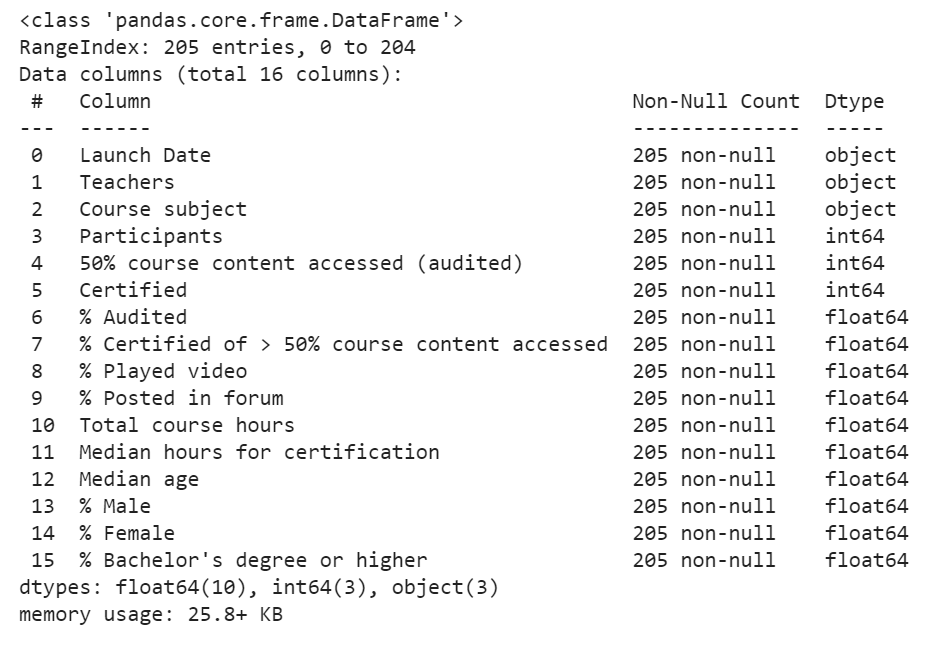


По графику видно, что средний возраст составляет около 27 лет.

# **Подготовка данных для машинного обучения**

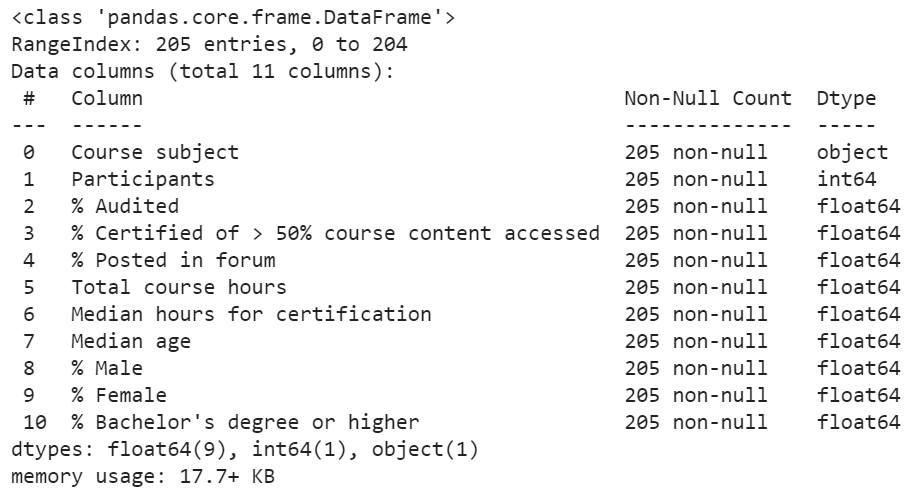
Произведем удаление ненужных столбцов (Certified','50% course content accessed (audited)','Teachers','Launch Date','% Played video')

df\_course.info()



df\_XGB = df\_course.drop(['Certified','50% course content accessed (audited)','Teachers','Launch Date','% Played video'],axis=1)

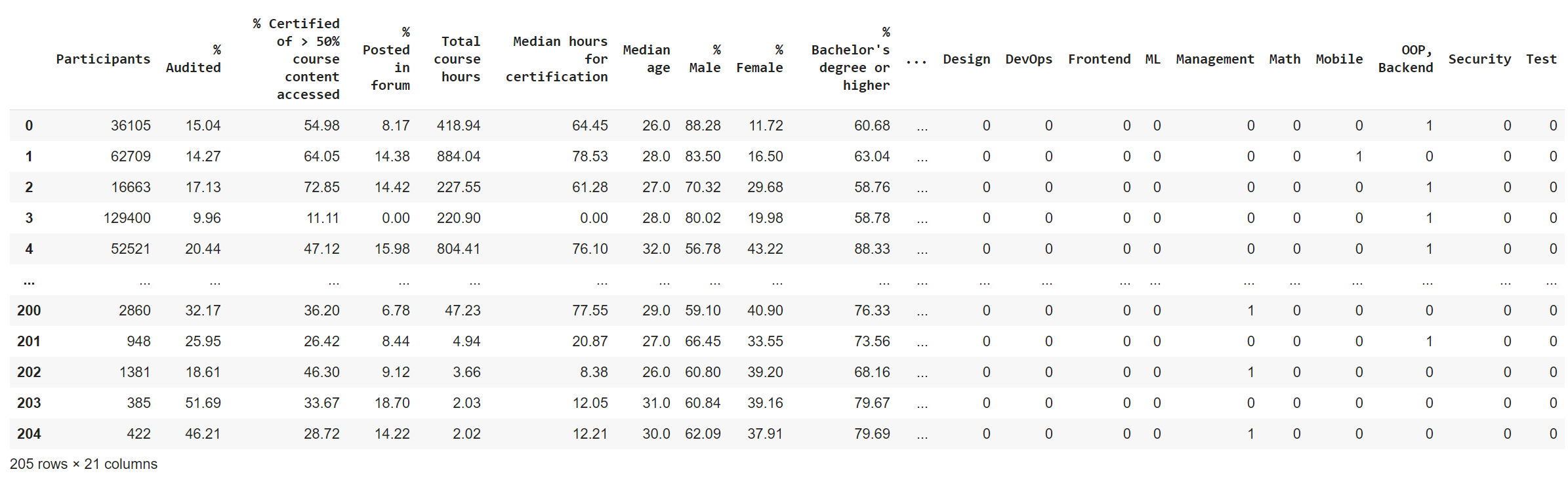
df\_XGB.info()



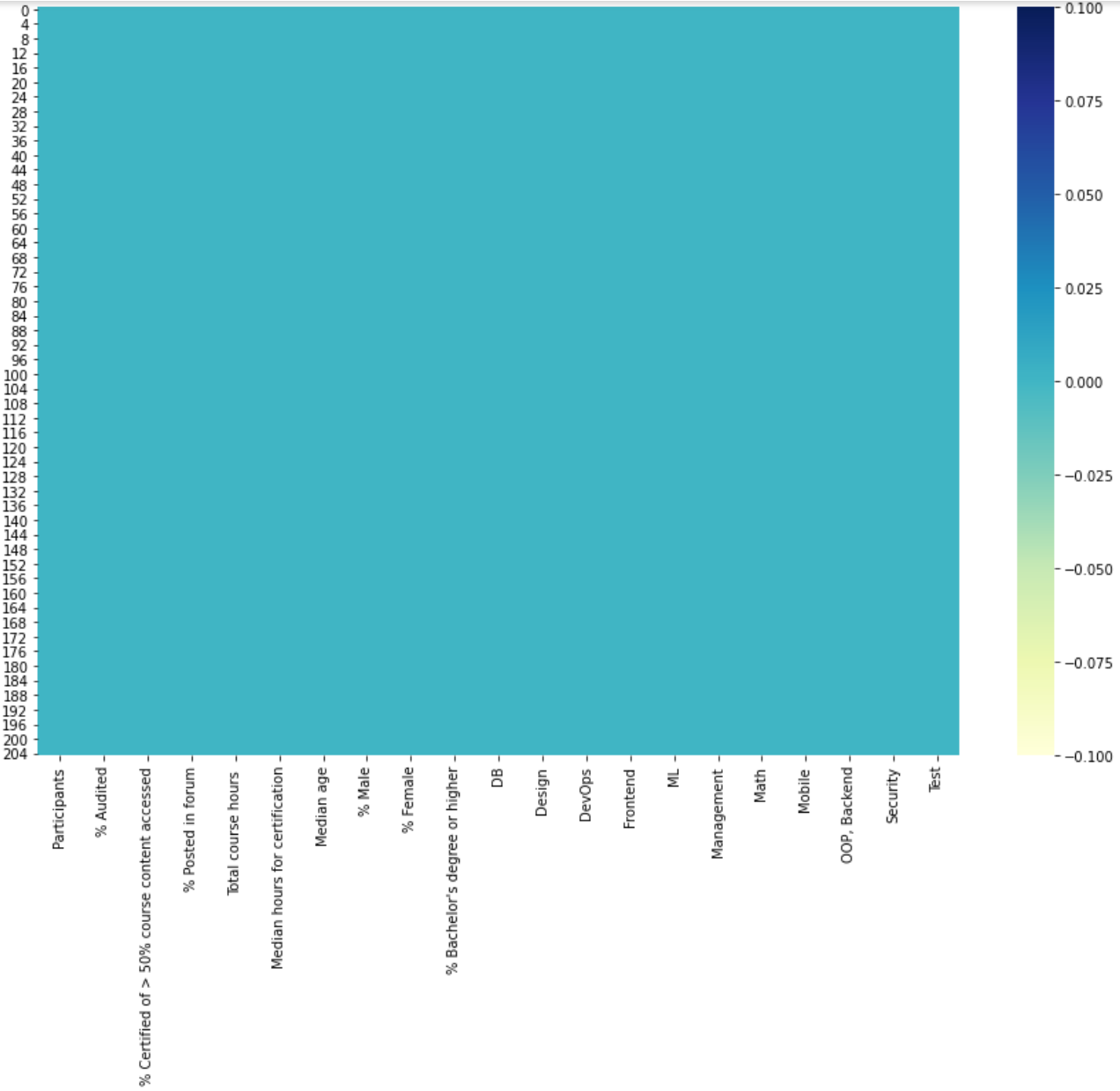
Произведем преобразование категориальных переменных в серии нулей и единиц, что значительно упрощает их количественное определение и сравнение.

CourseSubject = pd.get\_dummies(df\_XGB['Course subject'],drop\_first=True)  
df\_XGB.drop(['Course subject'],axis=1,inplace=True)  
df\_XGB = pd.concat([df\_XGB,CourseSubject],axis=1)

df\_XGB

Проверка на нулевые значения:

plt.figure(figsize=(15,10))  
sns.heatmap(df\_XGB.isnull(),cmap="YlGnBu")



Убедились, что пустых ячеек нет.

# **Библиотека XGBoost**

XGBoost - это оптимизированная распределенная библиотека повышения градиента, разработанная для обеспечения высокой эффективности, гибкости и портативности. Он реализует алгоритмы машинного обучения в рамках платформы Gradient Boosting. XGBoost обеспечивает усиление параллельного дерева (также известное как GBDT, GBM), которое позволяет быстро и точно решить многие проблемы data science. Один и тот же код работает в основной распределенной среде (Hadoop, SGE, MPI) и может решать проблемы, выходящие за рамки множества примеров.

В основе **XGBoost** лежит алгоритм [градиентного бустинга](https://neerc.ifmo.ru/wiki/index.php?title=%D0%91%D1%83%D1%81%D1%82%D0%B8%D0%BD%D0%B3,_AdaBoost) [деревьев решений](https://neerc.ifmo.ru/wiki/index.php?title=%D0%94%D0%B5%D1%80%D0%B5%D0%B2%D0%BE_%D1%80%D0%B5%D1%88%D0%B5%D0%BD%D0%B8%D0%B9_%D0%B8_%D1%81%D0%BB%D1%83%D1%87%D0%B0%D0%B9%D0%BD%D1%8B%D0%B9_%D0%BB%D0%B5%D1%81). Градиентный бустинг — это техника машинного обучения для задач классификации и регрессии, которая строит модель предсказания в форме ансамбля слабых предсказывающих моделей, обычно деревьев решений. Обучение ансамбля проводится последовательно в отличие, например от [бэггинга](https://neerc.ifmo.ru/wiki/index.php?title=%D0%92%D0%B8%D0%B4%D1%8B_%D0%B0%D0%BD%D1%81%D0%B0%D0%BC%D0%B1%D0%BB%D0%B5%D0%B9" \o "Виды ансамблей). На каждой итерации вычисляются отклонения предсказаний уже обученного ансамбля на обучающей выборке. Следующая модель, которая будет добавлена в ансамбль будет предсказывать эти отклонения. Таким образом, добавив предсказания нового дерева к предсказаниям обученного ансамбля мы можем уменьшить среднее отклонение модели, котрое является таргетом оптимизационной задачи.

Разделим данные на обучающую и тестовую выборки

from sklearn.model\_selection import train\_test\_split  
x= df\_XGB  
y=df\_XGB['% Certified of > 50% course content accessed']  
x\_train, x\_test, y\_train, y\_test = train\_test\_split(x,y, test\_size=0.4, random\_state=109)

import xgboost as xgb  
train= xgb.DMatrix(x\_train,label=y\_train)  
test = xgb.DMatrix(x\_test, label= y\_test)

# **Список гиперпараметров XGBoost**

learning\_rate: уменьшение размера шага, используемое для предотвращения переобучения. Диапазон [0,1]

max\_depth: определяет, насколько глубоко каждое дерево может расти во время любого раунда повышения.

subsample: процент использованных образцов на дерево. Низкое значение может привести к неполному оснащению.

colsample\_bytree: процент функций, используемых в дереве. Высокое значение может привести к переобучению.

n\_estimators: количество деревьев, которые вы хотите построить.

objective: определяет функцию потерь, которая будет использоваться, например, линейная для задач регрессии, логистическая для задач классификации с единственным решением, двоичная для задач классификации с вероятностью.

XGBoost также поддерживает параметры регуляризации, чтобы наказывать модели по мере их усложнения и сводить их к простым (экономным) моделям

gamma: контролирует, будет ли данный узел разделен на основе ожидаемого сокращения потерь после разделения. Чем выше значение, тем меньше расщеплений.

alpha: L1 регуляризация весов листьев. Большое значение ведет к большей регуляризации.

lambda: L2 регуляризация весов листьев и более плавная, чем регуляризация L1.

xg\_reg = xgb.XGBRegressor(objective ='reg:linear', colsample\_bytree = 0.5, learning\_rate = 0.2,  
 max\_depth = 7, alpha = 10, n\_estimators = 75)

xg\_reg.fit(x\_train,y\_train)  
preds = xg\_reg.predict(x\_test)

Вычислим ошибку прогноза

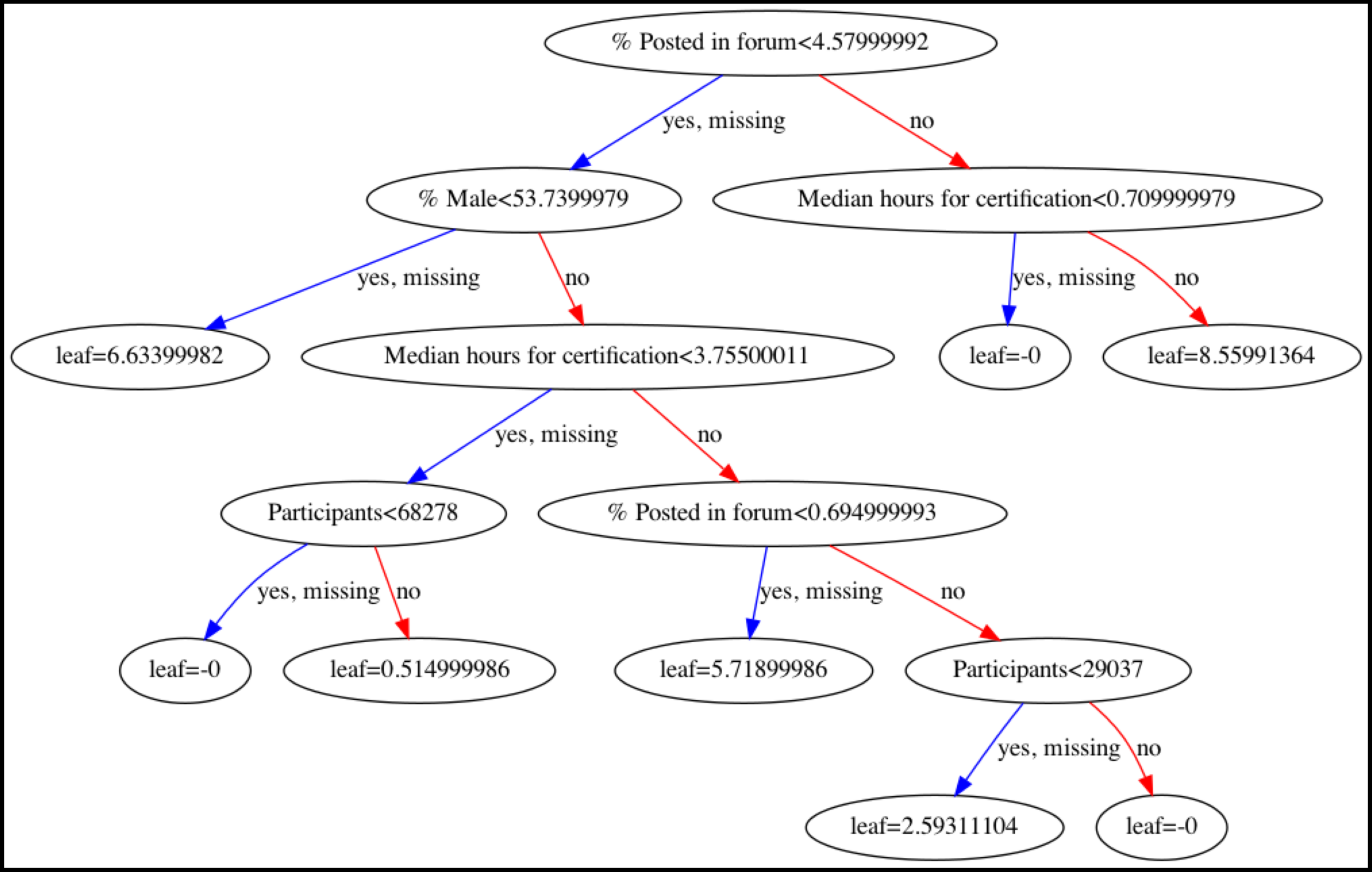
from sklearn.metrics import mean\_squared\_error  
rmse = np.sqrt(mean\_squared\_error(y\_test, preds))  
print("RMSE: %f" % (rmse))

RMSE: 6.109611

Ошибка составила 6%

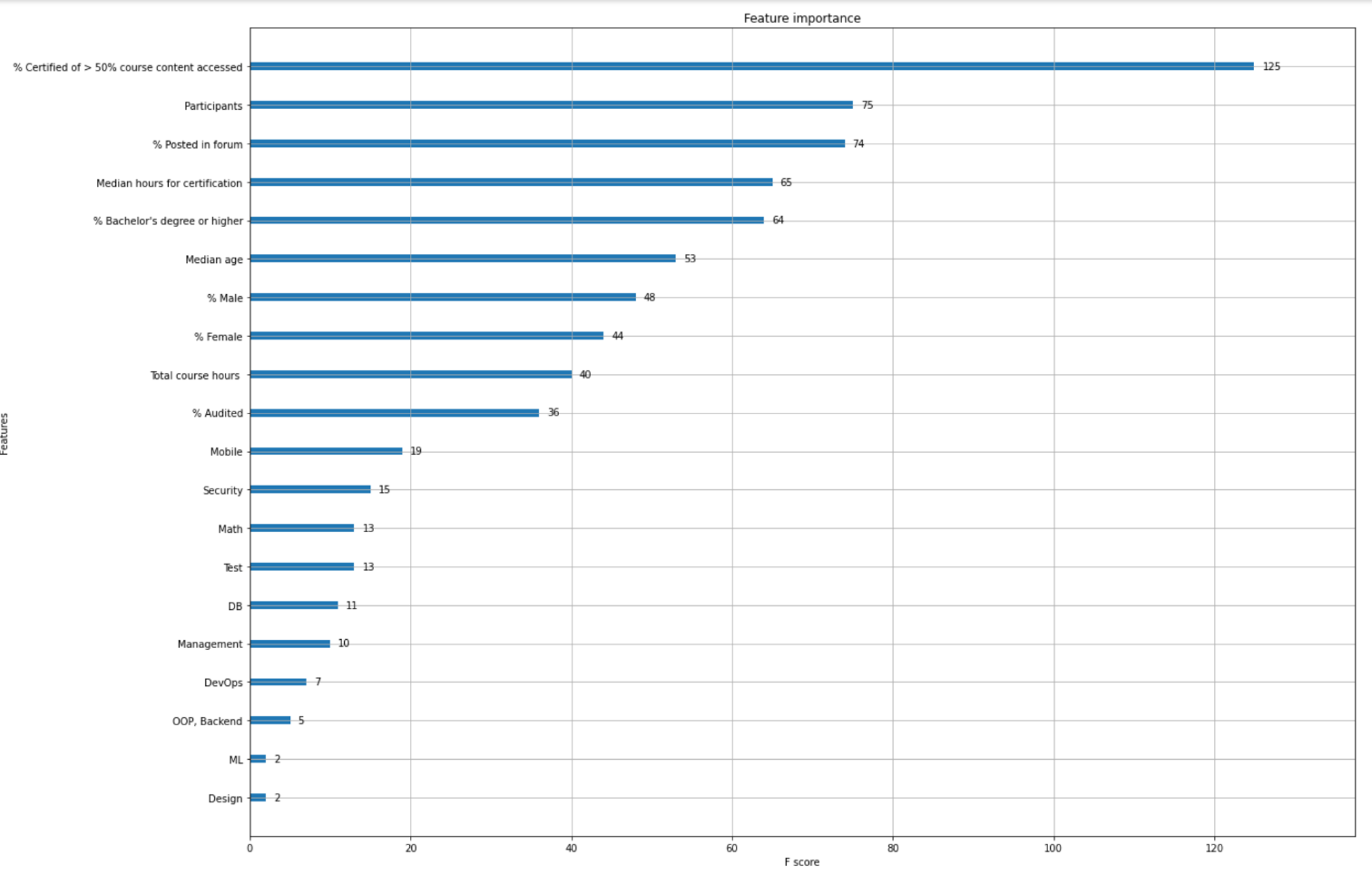
# **Визуализация деревьев XGBoost**

import matplotlib.pyplot as plt  
  
xgb.plot\_tree(xg\_reg,num\_trees=0)  
plt.rcParams['figure.figsize'] = [20, 15]  
plt.show()



Построим график важности признаков на основе подобранных деревьев.

xgb.plot\_importance(xg\_reg)  
plt.rcParams['figure.figsize'] = [15,15]  
plt.show()



По графику видно, что наибольшей важностью обладает признак «% Certified of > 50% course content accessed»

# **Вывод**

В данной курсовой работе была произведена очистка, визуализация, а также анализ данных от одной из платформ онлайн-курсов. С использованием библиотеки XGBoost была произведена попытка создания модели, которая предсказывает, завершит ли пользователь более 50% курса или нет. Точность прогноза модели составила около 94%.

# **Список использованных источников**

1. Friedman J. Greedy Function Approximation: A Gradient Boosting Machine. — IMS 1999 Reitz Lecture.
2. Nonita Sharma, XGBoost. The Extreme Gradient Boosting for Mining Applications. - 2018 GRIN Verlag
3. xgboost documentation. Режим доступа: https://xgboost.readthedocs.io/en/latest/ Дата обращения: 28.11.2020
4. seaborn documentation. Режим доступа: https://seaborn.pydata.org/docs/ Дата обращения: 28.11.2020
5. pandas documentatio.n Режим доступа: https://pandas.pydata.org/docs/ Дата обращения: 28.11.2020
6. Введение в pandas: анализ данных на Python. Режим доступа: https://khashtamov.com/ru/pandas-introduction/ Дата обращения: 28.11.2020