INTRODUCTION TO DATA SCIENCE

SCORE ANALYSIS

MEMBER OF GROUP

MEMBER

1. Nguyễn Tấn Phát	- 20127588	40%
2. Lê Ngọc Tường	- 20127383	35%
3. Huỳnh Lợi Chuẩn	- 19127344	25%

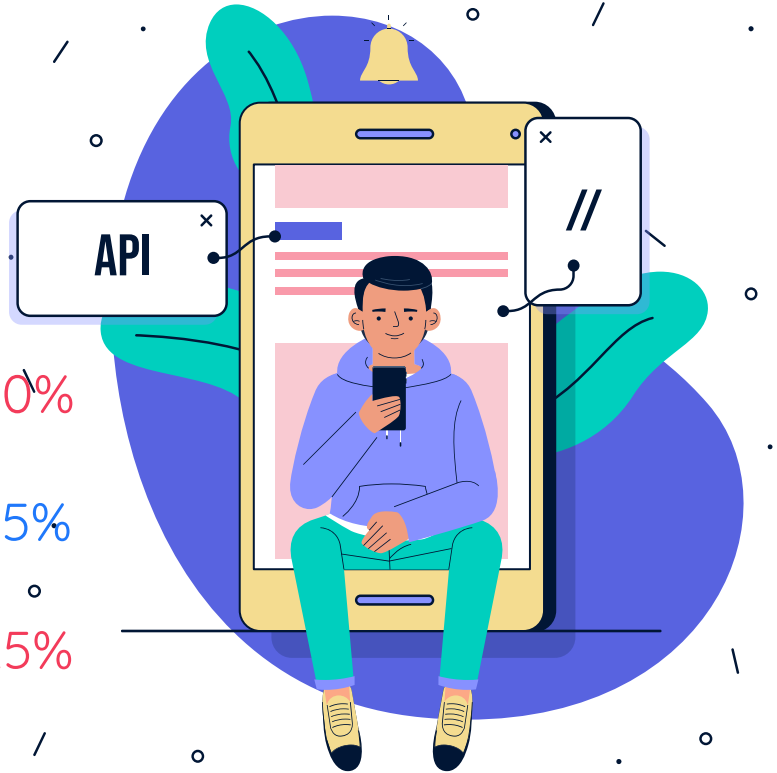


TABLE OF CONTENTS

01. COLLECTION

Lấy dữ liệu từ web về dưới dạng csv

02. CLEANING

Tiền xử lí dữ liệu, xóa thông tin thừa,...

03. EXPLORATORY DATA ANALYSIS

Thăm dò, phân tích, đúc trích dữ liệu

04. MODEL

Xây dựng các mô hình machine learning

01

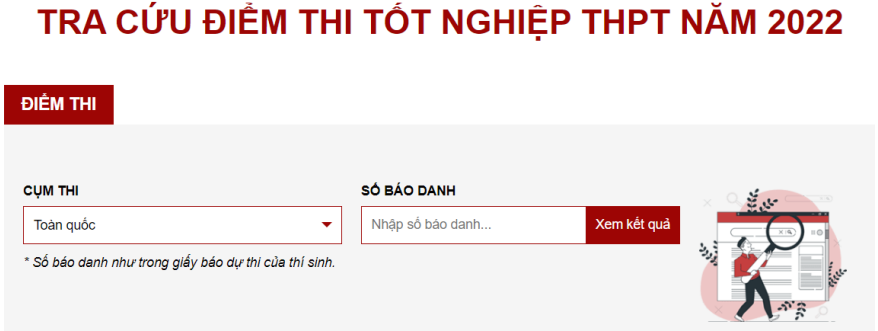
DATA SCIENCE

COLLECTION DATA

CHỌN WEBSITE

Giáo dục thời đại – Ngọc Tường

Thanh niên – Tấn Phát



Nhanh nhất

Lấy 1 lần
100 thí sinh

Tiền phong – Lợi Chuẩn

CÁCH LẤY DỮ LIỆU

1. Sử dụng subprocess để sử dụng các lệnh trong cmd

```
import subprocess
import time
```

2. Dùng lệnh ‘curl’ để lấy API của website

```
response = subprocess.check_output('curl "https://tienphong.vn/api/diemthi/get/result?
value = json.loads(response.decode('utf-8'))
```

3. Xử lí string, đưa về dạng csv

```
error.close()
file = open('csv.csv', 'w', encoding='utf-8')
for lstScore in allValue:
    for i in range(0, 10):
        file.write(lstScore[i] + ',')
    file.write(lstScore[10] + '\n')
```

02

CLEANING DATA

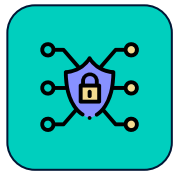
Phân loại đối tượng

K2022 thi THPTQG

6/9 môn có điểm
~850k thí sinh

GDTX 2022 thi THPTQG

5/9 môn có điểm và không có điểm TA
~10k thí sinh

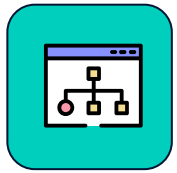


Khóa trước thi lại ĐH

Từ 3 -> 5 môn có điểm và nếu là 5 môn thì có điểm TA
~125k thí sinh

Khóa trước thi lại TN

Từ 1 -> 2 môn có điểm
~1k thí sinh



Đặc điểm các đối tượng

K2022 thi THPTQG

Thuộc 1 trong 2 trường hợp:

- T, V, A và Khối KHTN
- T, V, A và Khối KHXH

GDTX 2022 thi THPTQG

Thuộc 1 trong 2 trường hợp:

- T, V và Khối KHTN
- T, V và Khối KHXH

Khóa trước thi lại ĐH

Vì thi lại để xét ĐH nên sẽ có từ 3 đến 5 môn, TH 6 môn ta xếp vào đt 1

Khóa trước thi lại TN

Các điểm năm trước bảo lưu và năm nay thi để xét TN, TH 3 đến 5 môn ta xếp vào đt 3

Khám phá - Tiền xử lí dữ liệu

Hàng, cột, miss

Raw:

995.441, 12

Xóa Hàng

Xóa các dòng mà thí sinh không có bất kì điểm nào (6 dòng)

Thêm cột

Thêm cột phân loại đt:

- THPT_KHTN,THPT_KHXH
- GDTX_KHTN, GDTX_KHXH
- DH_RETEST
- TN_RETEST

Ghi File

Ghi dữ liệu đã xử lí vào file new_data.csv

Khám phá - Tiền xử lí dữ liệu

	col	% miss
cum	995441	0.000000
sbd	995441	0.000000
toan	982726	1.277323
ngu_van	981407	1.409827
ngoai_ngu	870609	12.540372
vat_li	325523	67.298614
hoa_hoc	327367	67.113370
sinh_hoc	322198	67.632637
lich_su	659662	33.731683
dia_ly	657421	33.956809
gdcd	554343	44.311818

Về % miss:

- Các cột cum, sbd là đầy đủ
- Toán, Văn miss ~1%
- Ngoại ngữ miss ~12% do các thí sinh tự do xét tuyển không cần môn này
- Vật lí, hóa học , sinh học miss ~2/3 do 2/3 các thí sinh thi KHXH
- Lịch sử, địa lý, GDCD miss ~1/3 do 1/3 các thí sinh thi KHTN

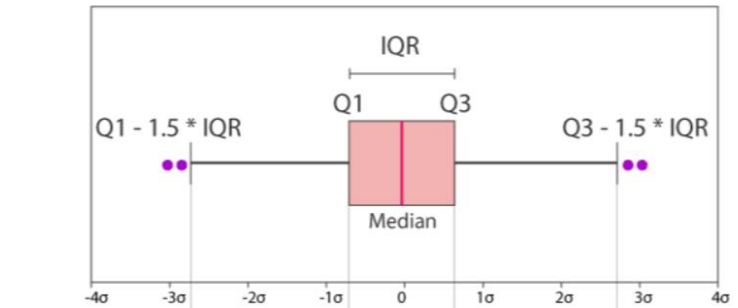
03

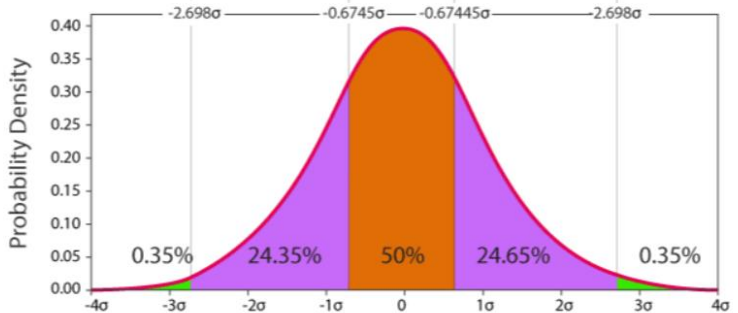
DATA SCIENCE
EXPLORATORY DATA ANALYSIS

Do khá nhiều câu hỏi(README.md), nên tụi em vừa phân tích và vừa trả lời ở từng file một

PHÂN BỔ DỮ LIỆU

Biểu diễn dữ liệu ở dạng Boxplot - từ đó quan sát và nhận xét sự phân bố dữ liệu





Boxplot on a normal distribution

© Byjus.com

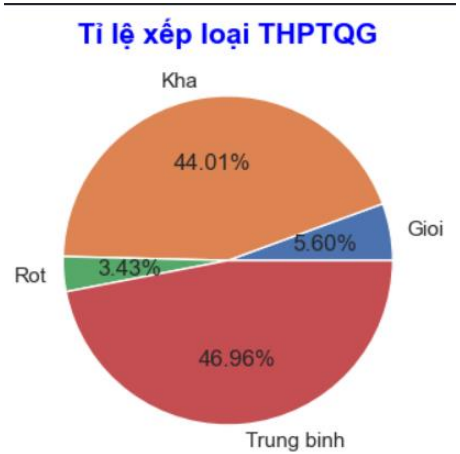
2. Tính điểm trung bình tốt nghiệp DT1 & DT2 - kiểm tra đậu/rớt - tốt nghiệp loại gì?

- Điểm tốt nghiệp của THPTQG DT1:

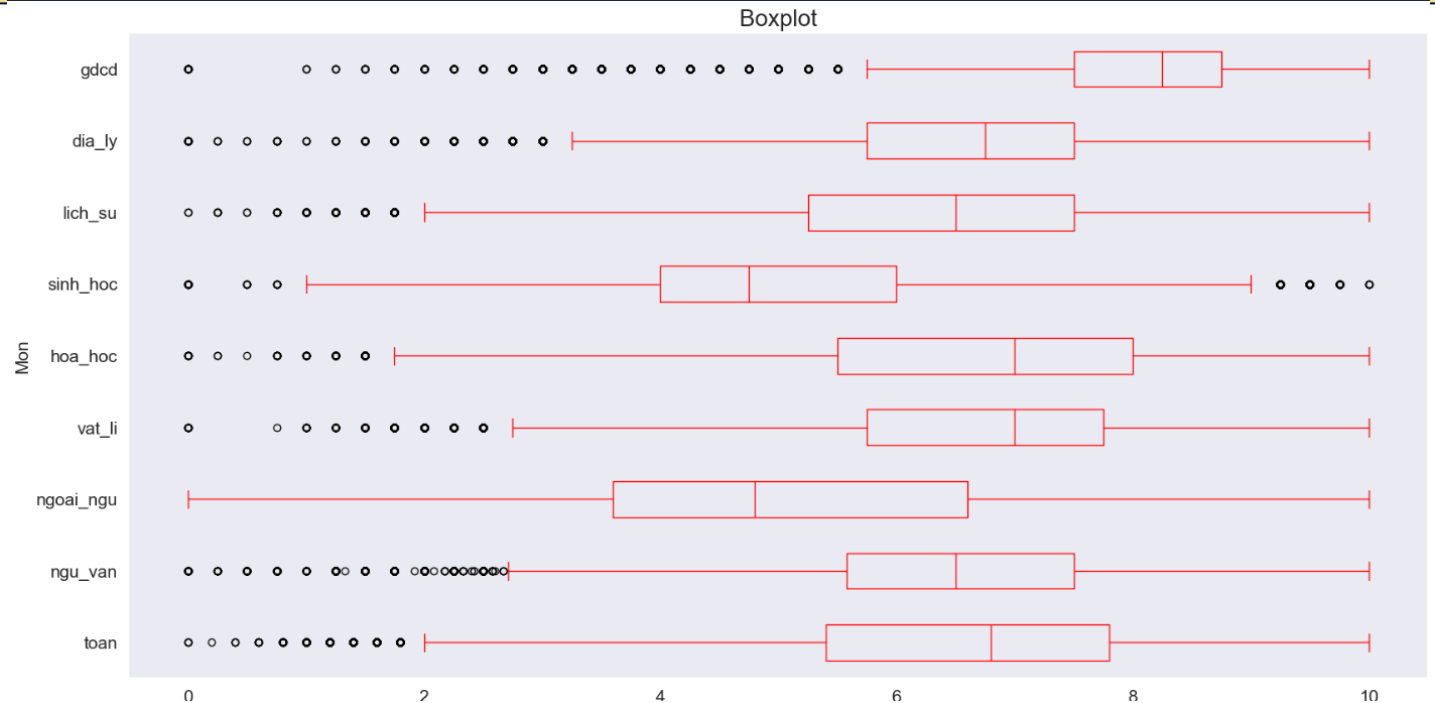
$$DXTN = \frac{Toan + NguVan + Ngoaingữ + average(KHTN/KHXH)}{6}$$

- Điểm tốt nghiệp của GNTX DT2:

$$DXTN = \frac{Toan + NguVan + average(KHTN/KHXH)}{5}$$



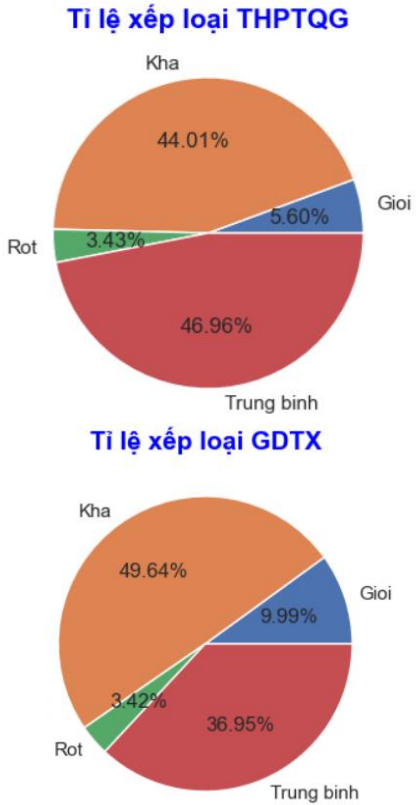
PHÂN BỔ DỮ LIỆU

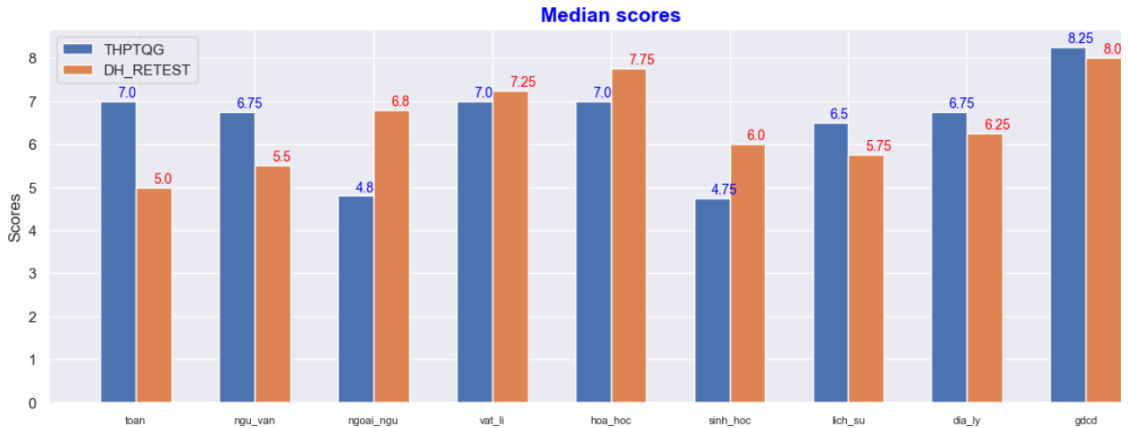


Tổng quan: Có sự phân bố điểm không đồng đều giữa các môn, sẽ chia làm 3 mức độ: dễ, trung bình, khó

- Khó - Ngoại ngữ _ Sinh học
- Dễ - GDCD
- Trung bình - Các môn còn lại

PHÂN BỔ DỮ LIỆU

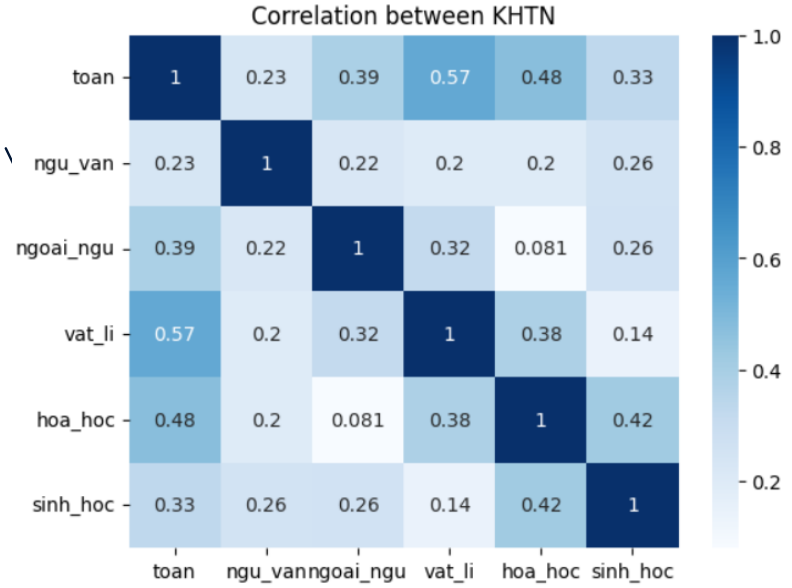


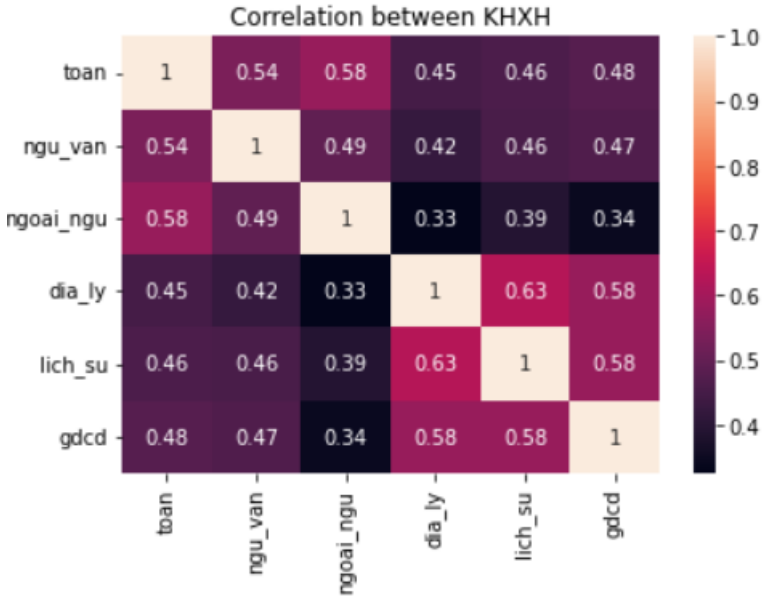


Nhận xét

- Đối với Toán và Ngữ văn: đa số học sinh thi THPTQG đều cao hơn Thi_lại_DH 1 - 2 điểm. => Hai môn này không được các thí sinh DT3 coi trọng. Hoặc là do sự ảnh hưởng của các thi sinh thi khối không có môn Toán (Văn)
- Đối với Ngoại ngữ và Sinh học: thì ngược lại so với trên, Thi_lại_DH cao hơn hẳn so với THPTQG => Một phần vì đây là 2 môn khó nhất trong tất cả các môn và cũng một phần đây là các được DT3 chú trọng.

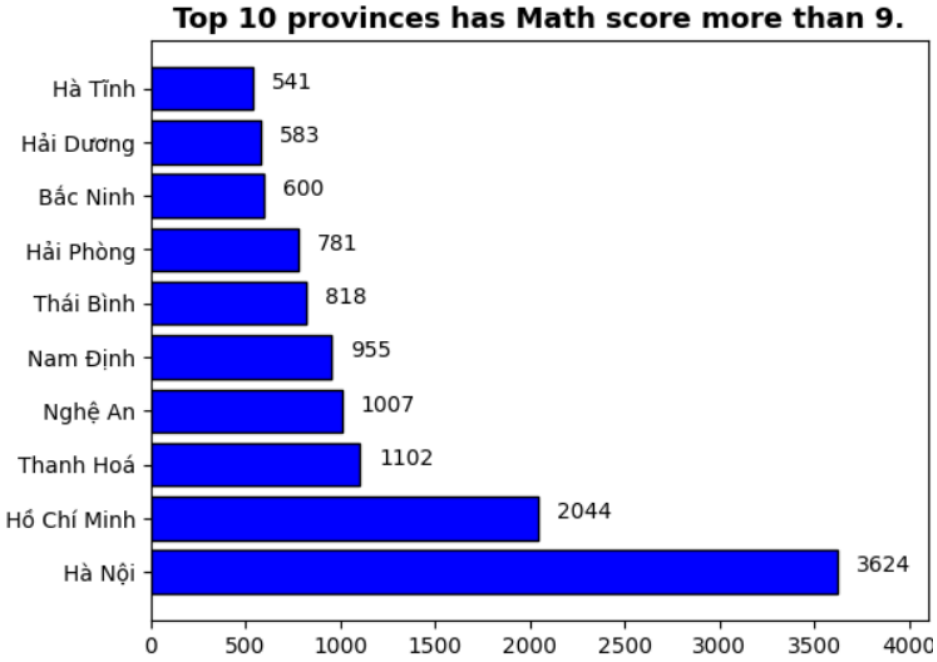
ĐỘ TƯƠNG QUAN





- Toán, Lí, Hoá có độ tương quan rất cao từ 0.38 đến 0.57 nên ta thấy một học sinh giỏi Toán thì sẽ giỏi Lý, Hoá. Ngược lại nếu môn Lí hoặc Hoá giỏi thì các môn còn lại sẽ giỏi
- Toán, Ngữ văn có độ tương quan 0.54 và là 2 môn bắt buộc trong xét tốt nghiệp thpt nên hầu hết mọi học sinh đều chú tâm, nên ta thấy nếu xét điểm Toán tăng dần thì điểm Ngữ văn sẽ tăng dần và ngược lại.

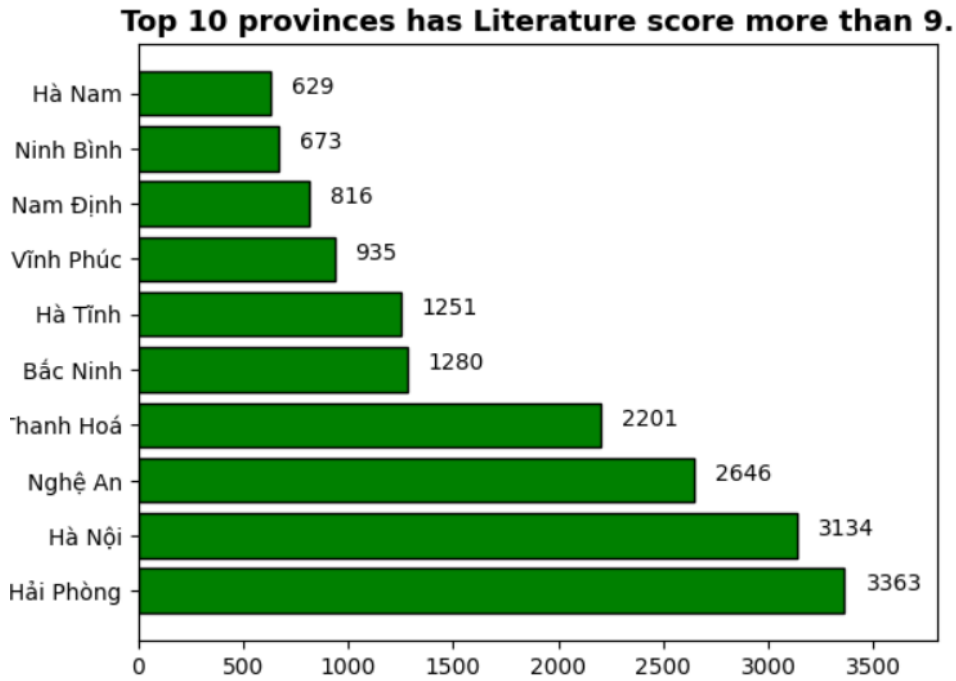
TOP 10



- TP.Hồ Chí Minh và Hà Nội có nhiều điểm Toán >= 9 nhất và cách xa các tỉnh/tp khác vì đây là hai TP lớn có nhiều trường nổi tiếng và học sinh giỏi đồng thời cũng vì là 2 TP đông dân nhất nên số thí sinh chiếm phần lớn.

- Các tỉnh khác như: Nghệ An, Thanh Hóa, Nam Định, Hải Phòng... có nền giáo dục tốt, có nhiều nhân tài và qua các năm đều có tỉ lệ điểm cao hơn.

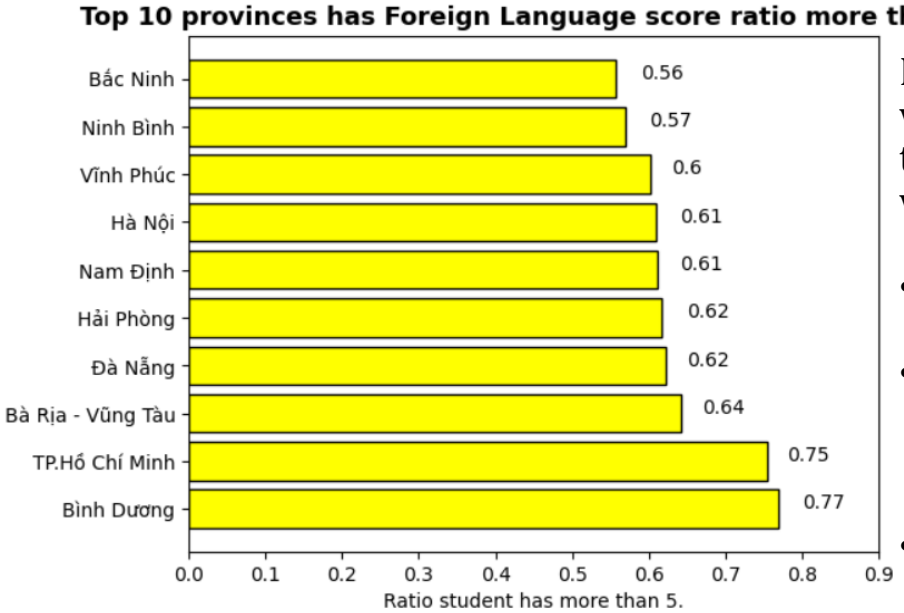
TOP 10



- Khác với Toán, Hải Phòng có số thí sinh trong top này cao nhất, và năm 2022 cũng là lần đầu tiên Hải Phòng có điểm môn văn cao nhất.

- Các tỉnh/tp Hà Nội, Nghệ An, Thanh Hóa, Nam Định, Bắc Ninh,... tiếp tục thuộc top này.

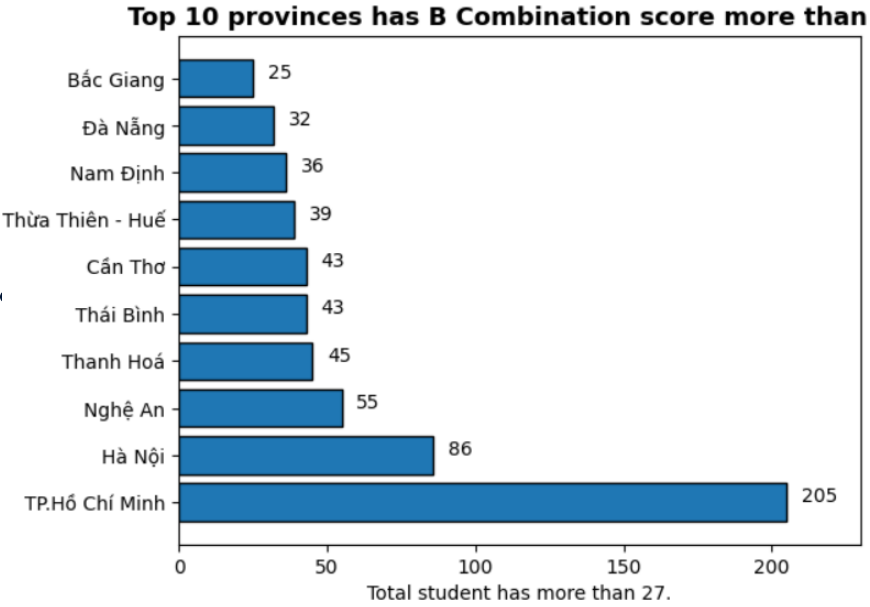
TOP 10



Người ta thường nói: Trình độ ngoại ngữ của một vùng sẽ đánh giá mức độ phát triển của vùng đó. Giờ ta sẽ kiểm chứng Điểm Ngoại_ngữ >= 5 tương ứng với trình độ tiếng anh mức độ cơ bản

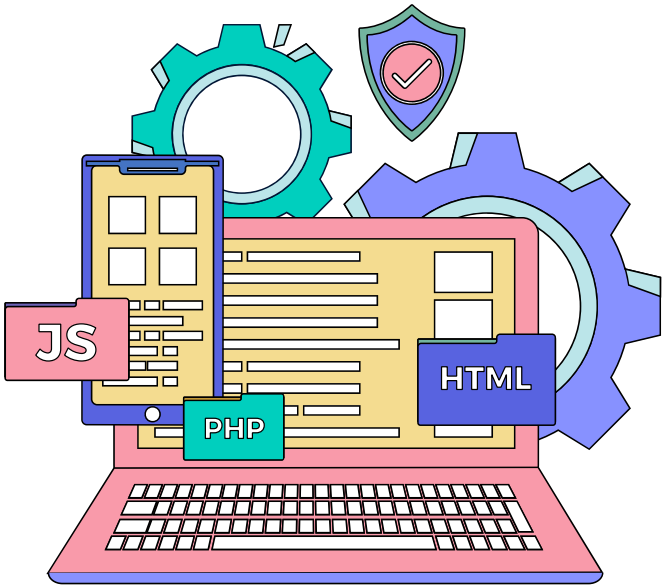
- Đứng top đầu là Bình Dương và TPHCM => Đây là nơi được đầu tư môn tiếng anh nhất
- Tiếp đến là Vùng Tàu, Hải Phòng, Đà Nẵng, Hà Nội, Nam Định,... => Các tỉnh thành này đều rất phát triển so với mặt bằng chung. Nhất là về kinh tế, du lịch, cơ sở vật chất - hạ tầng.
- Nghệ An, Hải Dương, Thanh Hóa tuy lọt Top10 tỉnh/thành có số thí sinh trên 9 nhiều nhất nhưng lại không lọt top này. => Chứng tỏ có nhiều người tài giỏi nhưng mặt bằng chung lại không được đầu tư phát triển.

TOP 10



Sở dĩ xét trên 27 vì đa số các ngành y đều phải đạt 27 trở lên mới đậu

- TPHCM là thành phố có số lượng điểm khối B >= 27 nhiều nhất - top1. => Đây là nơi có nhiều học sinh mong muốn được theo ngành y nhất và cũng một phần do TPHCM đông dân - nhiều trường y (bệnh viện) - Môi trường làm việc tốt.
- Hà Nội cũng tương tự với top2 nhưng lại không nhiều bằng TPHCM.
- Tiếp đến là Nghệ An, Thanh Hóa, Huế, Đà Nẵng, Cần Thơ => Các thí sinh của các tỉnh/thành này đều có học lực rất tốt.



MODEL

04

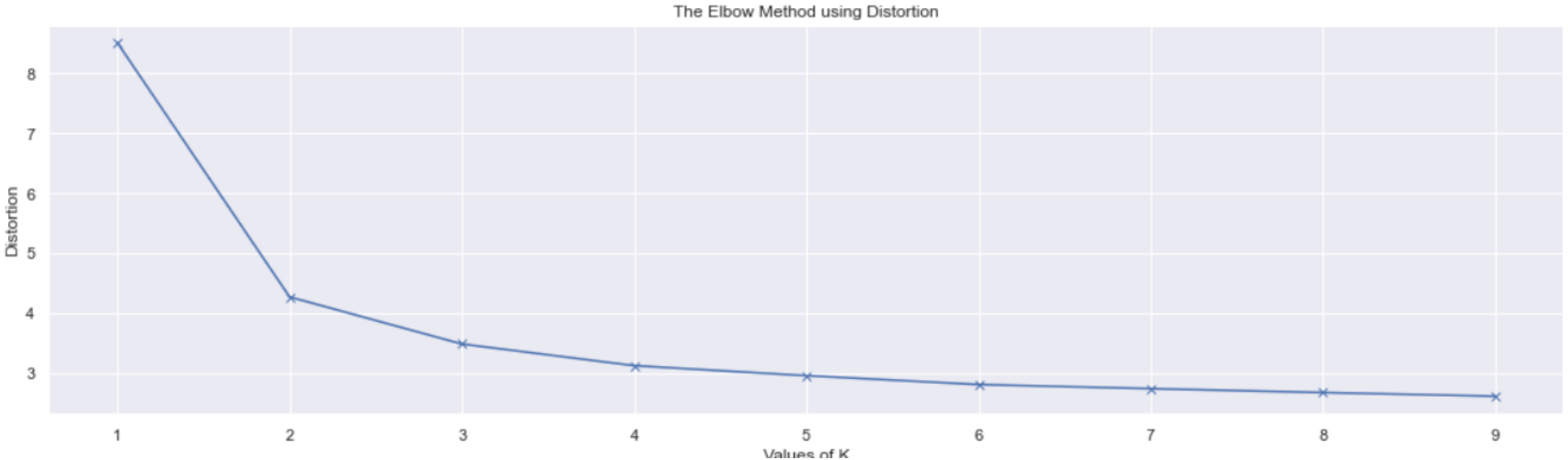
ML

GÁN NHÃN – KHỐI THI

	cum	sbd	toan	ngu_van	ngoai_ngu	vat_li	hoa_hoc	sinh_hoc	lich_su	dia_ly	gdcd	phan_loai	tong_diem_3_mon	khoi_thi
0	1	1000001	3.6	5.00	4.0	NaN	NaN	NaN	2.75	6.0	8.75	THPT_KHXH	13.75	C00
1	1	1000002	8.4	6.75	7.6	NaN	NaN	NaN	8.50	7.5	8.25	THPT_KHXH	24.40	A07
2	1	1000003	5.8	7.50	5.0	NaN	NaN	NaN	7.25	5.5	8.75	THPT_KHXH	20.25	C00
3	1	1000004	7.4	7.50	8.6	NaN	NaN	NaN	7.50	6.5	7.50	THPT_KHXH	23.50	D01
4	1	1000005	7.2	8.50	9.0	NaN	NaN	NaN	8.00	8.5	8.25	THPT_KHXH	25.00	C00

- Dựa vào tổng điểm 3 môn cao nhất của thí sinh, từ đó dự đoán thi sinh đó thi khối nào? Có tổng điểm là bao nhiêu?
- Quá trình tính toán sử dụng hàm so sánh khá nhiều, nên thời gian chạy khá lâu
- Cuối cùng, tạo file **data_khoi_thi.csv** và lưu.

KMEANS



- Sử dụng Kmeans để phân các thí sinh thành k cụm (theo điểm) từ đó có thể biết các nhóm thí sinh có chung sở thích, sở trường
- Sử dụng elbow để chọn số cụm k một cách phù hợp, ở đây số cụm được chọn là 4

LINEAR REGRESSION

```
GridSearchCV(cv=10,
             estimator=Pipeline(steps=[('standardscaler', StandardScaler()),
                                       ('polynomialfeatures',
                                        PolynomialFeatures()),
                                       ('linearregression',
                                        LinearRegression())]),
             param_grid={'linearregression__fit_intercept': [True, False],
                         'linearregression__positive': [True, False],
                         'polynomialfeatures__degree': [3, 4, 5]},
             scoring='neg_mean_squared_error')
```

```
RMSE: 0.7145047770164877
MSE: 0.5105170763793808
MAE: 0.5416058503581866
Model score: 1.0
```

```
Pipeline(steps=[('standardscaler', StandardScaler()),
                ('polynomialfeatures', PolynomialFeatures(degree=5)),
                ('linearregression', LinearRegression(normalize=False))])
```

- Sử dụng Linear Regression để dự đoán điểm môn toán qua điểm môn lý và hóa (các môn này có độ tương quan cao)
- Sử dụng GridSearch để tìm các parameter tốt nhất
- Cuối cùng, test cho ra MSE và MAE khá tốt (0.5)

LINEAR REGRESSION

```
GridSearchCV(cv=10,
             estimator=Pipeline(steps=[('standardscaler', StandardScaler()),
                                       ('polynomialfeatures',
                                        PolynomialFeatures()),
                                       ('linearregression',
                                        LinearRegression())]),
             param_grid={'linearregression__fit_intercept': [True, False],
                         'linearregression__positive': [True, False],
                         'polynomialfeatures__degree': [3, 4, 5]},
             scoring='neg_mean_squared_error')
```

```
RMSE: 1.1818903177371458
MSE: 1.3968647231608113
MAE: 0.948635548933555
Model score: 1.0
```

- Tương tự, sử dụng Linear Regression để dự đoán điểm môn ngữ văn qua điểm môn lịch sử, địa lý, gdcd (các môn này có độ tương quan cao)
- Sử dụng GridSearch để tìm các parameter tốt nhất
- Cuối cùng, test cho ra MSE và MAE (1.39 và .94) là khá ổn, tuy vậy vẫn cao hơn mô hình dự đoán điểm toán

LOGISTIC REGRESSION

- Dựa vào sự phân bố loại tốt nghiệp ở file 1_Distribution.ipynb có thể thấy:
  - Tỉ lệ học sinh rớt tốt nghiệp rất ít (3%)
  - Nên không đủ để đánh giá, dự đoán xem thí sinh đậu hay rớt
- Với file 2_Correlation:
  - Môn toán (KHTN):
    - Tương quan mạnh với vật lí _ hóa học
    - Tương quan vừa với ngoại ngữ _ sinh học
  - Môn ngữ văn (KHXH):
    - Tương quan mạnh với ngoại ngữ _ GDCD
    - Tương quan mạnh với lịch sử _Địa lý

Vậy từ điểm môn toán - ngữ văn (Hai môn thi đầu tiên) ta sẽ dự đoán thí sinh tốt nghiệp THPT loại "khá - giỏi" hay là không??*

LOGISTIC REGRESSION

- Tiền xử lí tạo cột xếp loại
- Lấy ra các hàng giỏi, khá
- Vì giỏi ít hơn 8 lần so với khá nên ta thực hiện over sampling
- Để tìm tham số tốt nhất, ta sử dụng **Optuna** trên pipeline (gồm chuẩn hóa StandardScaler và LogisticRegression) dựa trên cross_val_score (có cv là RepeatedStratifiedKFold với k = 10 và lặp lại 3 lần, đánh giá bằng trung bình của f1-score)
- Và sau đó là các bước fit và predict

LOGISTIC REGRESSION

```
Accuracy: 0.7918179266959708
Best hyperparameters: {'penalty': 'l2', 'tol': 1.7571855365552842e-06, 'max_iter': 200.0, 'fit_intercept': 0}
```

	precision	recall	f1-score	support
0	0.81	0.74	0.77	38085
1	0.76	0.83	0.80	38475
accuracy			0.78	76560
macro avg	0.79	0.78	0.78	76560
weighted avg	0.79	0.78	0.78	76560

- Nhìn chung các chỉ số và f1-score đều ổn
- Nếu không có over sampling thì khá sẽ có f1-score cao và giỏi sẽ có f1-score thấp

DỰ ĐOÁN ĐIỂM CHUẨN

- Chúng ta cần phải biết điểm chuẩn của ngành đó vào năm ngoái (2021)
- Khối thi ngành đó năm 2021 2022 (thường không đổi)
- Tỉ lệ thi giữa các khối

```
khoiA00 = ['toan','vat_li','hoa_hoc']
khoiA01 = ['toan', 'vat_li', 'ngoai_ngu']
khoiA02 = ['toan', 'vat_li', 'sinh_hoc']
khoiA07 = ['toan', 'lich_su', 'dia_ly']
khoiB00 = ['toan', 'sinh_hoc', 'hoa_hoc']
khoiB03 = ['toan', 'sinh_hoc', 'ngu_van']
khoiB08 = ['toan', 'sinh_hoc', 'ngoai_ngu']
khoiC00 = ['ngu_van', 'lich_su', 'dia_ly']
khoiC01 = ['ngu_van', 'toan', 'vat_li']
khoiC02 = ['ngu_van', 'toan', 'hoa_hoc']
khoiD01 = ['ngu_van', 'toan', 'ngoai_ngu']
khoiD07 = ['toan', 'hoa_hoc', 'ngoai_ngu']
khoiD08 = ['toan', 'sinh_hoc', 'ngoai_ngu']
```

```
C00    379394
A07    190098
D01    123663
A00     96795
C01     51762
C02     51064
A01     45924
B00     20346
D07     19563
B03      8101
A02      2863
B08      2422
Name: khoi_thi, dtype: int64
```

DỰ ĐOÁN ĐIỂM CHUẨN

- Đầu tiên lấy dữ liệu điểm của năm 2021 (đã được phân tích) xem phân bố điểm của từng môn là như thế nào
- Tiếp đến, tính tỉ lệ học sinh trên điểm 8, trên điểm 9 của năm 2022 và so với năm 2021:
- Nếu tỉ lệ dưới 1 thì tức là đề năm 2022 khó hơn, ngược lại nếu tỉ lệ trên 1 thì đề năm 2021 khó hơn

diem	ngu_van	vat_li	hoa_hoc	sinh_hoc	lich_su	dia_ly	gdcd
0	26	5	10	25	4	94	24
0.25	10	2	1	1	7	1	0
0.5	45	1	1	1	24	3	1
0.75	68	3	20	11	125	9	2
1	23	14	26	37	380	11	2
1.25	563	37	90	109	957	26	4
1.5	820	76	151	176	2351	42	12
1.75	984	162	334	364	4417	72	10
2	1667	277	612	620	8118	142	19
2.25	1879	494	984	1293	12773	250	45
2.5	2995	819	1519	1924	17921	390	56

```
point_count2022 = \
{\
'toan'      : dict(data_2022['toan'].value_counts()),
'ngoai_ngu' : dict(data_2022['ngoai_ngu'].value_counts()),
'ngu_van'   : dict(data_2022['ngu_van'].value_counts()),
'vat_li'    : dict(data_2022['vat_li'].value_counts()),
'hoa_hoc'   : dict(data_2022['hoa_hoc'].value_counts()),
'sinh_hoc'  : dict(data_2022['sinh_hoc'].value_counts()),
'lich_su'   : dict(data_2022['lich_su'].value_counts()),
'dia_ly'    : dict(data_2022['dia_ly'].value_counts()),
'gdcd'      : dict(data_2022['gdcd'].value_counts()),
}
```

DỰ ĐOÁN ĐIỂM CHUẨN

- Điểm dự đoán = Điểm 2021 * (tỉ lệ điểm trên 8 2022/2021)
- Mô hình chỉ dự đoán theo công thức (không có train) nên một số ngành vẫn có sai số lớn (2đ). Ngoài ra, điểm chuẩn còn bị ảnh hưởng bởi rất nhiều yếu tố như xu hướng, sở thích, tỉ lệ xét tuyển theo hình thức THPTQG (vì còn có hình thức khác như DGNL, tuyển thẳng),...
- Một vài giá trị dự đoán

```
# Ngành: Nhóm ngành CNTT - trường ĐH Khoa học tự nhiên - DHQG TP.HCM --> 27.2
diem2021 = 27.4
khoi_xet_tuyen = ['A00', 'A01', 'B08', 'D07']
diem2022 = predict_score(diem2021, khoi_count, khoi_xet_tuyen, rate_8_subject, rate_9_subject)
print(diem2022)
```

✓ 0.4s

```
27.174712448444073
```

```
# Ngành: Hóa học - trường ĐH Khoa học tự nhiên - DHQG TP.HCM -->23.75
diem2021 = 24.5
khoi_xet_tuyen = ['A00', 'B00']
diem2022 = predict_score(diem2021, khoi_count, khoi_xet_tuyen, rate_8_subject, rate_9_subject)
print(diem2022)
```

✓ 0.4s

```
25.808695545268623
```

```
# Ngành: Công nghệ sinh học - trường ĐH Khoa học tự nhiên - DHQG TP.HCM -->23.75
diem2021 = 25.5
khoi_xet_tuyen = ['A02', 'B00']
diem2022 = predict_score(diem2021, khoi_count, khoi_xet_tuyen, rate_8_subject, rate_9_subject)
print(diem2022)
```

✓ 0.3s

```
# Ngành: Kiểm toán - Đại học Kinh tế quốc dân --> 28.15
diem2021 = 28.1
khoi_xet_tuyen = ['A00', 'A01', 'D07']
diem2022 = predict_score(diem2021, khoi_count, khoi_xet_tuyen, rate_8_subject, rate_9_subject)
print(diem2022)
```

✓ 0.2s

```
28.001717409695782
```

DATA SCIENCE

REFLECTION

1. Nguyễn Tấn Phát - 20127588

DIFFICULTIES

- Dữ liệu không mang nhiều thông tin (ý nghĩa)
- Do có đến 1tr dòng nên mỗi lần tính toán (so sánh) thì thời gian rất lâu
- Khi crawl quá nhiều thì web sẽ bị lag, phải delay một vài giây

LEARNED

- Các cách để crawl dữ liệu và xử lí khi web bị lag.
- Một số cách xử lí hoặc các hàm hay trong pandas
- Như thế nào là Data Science, các bước làm như nào, thứ tự ra sao
- Cách dự đoán mà không phụ thuộc vào mô hình tuyến tính

IF I HAD MORE TIME

- Lấy thêm dữ liệu các năm trước để dự đoán điểm chuẩn chính xác hơn.
- Lấy dữ liệu thi thử của một vài trường. Kiểm tra xem điểm thi thật và thi thử có lệch nhau nhiều không? (gian lận không)

2. Lê Ngọc Tường - 20127383

DIFFICULTIES

- Lượng dữ liệu lớn (1 triệu đối tượng) cũng gây trở ngại trong việc khám phá và tiền xử lí dữ liệu, để phân loại ra đối tượng này.
- Khó khăn về quản lí thời gian phải cân bằng với deadline những môn khác.

LEARNED

- Hiểu rõ hơn khi được thực hiện một quy trình khoa học dữ liệu
- Nâng cao kỹ năng sử dụng thư viện python và các kỹ thuật trong data science, đặc biệt là sklearn dùng để tìm mô hình và optuna dùng để tối ưu tham số

IF I HAD MORE TIME

- Phân tích thêm nhiều khía cạnh khác của phổ điểm các môn khác nhau
- Tối ưu các mô hình linear để cho ra kết quả tốt hơn nhưng vẫn tránh overfitting

3. Huỳnh Lợi Chuẩn - 19127344

DIFFICULTIES

- Chưa hiểu biết về các hàm của pandas
- Do có 1tr dòng và code chưa tối ưu dẫn đến tính toán mất nhiều thời gian.

LEARNED

- Tiếp xúc, học nhiều hơn về python
- Một số cách xử lí hoặc các hàm trong pandas
- Như thế nào là Data Science, các bước làm như nào, thứ tự ra sao
- Nhận xét các biểu đồ: Boxplot, Heatmap,...

IF I HAD MORE TIME

- Tối ưu code khiến việc xử lý nhanh hơn
- Tìm hiểu và ứng dụng thêm các biểu đồ vào đồ án

DATA SCIENCE

FILE DETAILS

FOLDER: DATA – FILE: GET_DATA_TO_CSV.PY

- Do số lượng thí sinh rất nhiều, nên phải chia ra nhiều lần để lấy (theo các cụm)
- Sau khi lấy toàn bộ rồi thì gộp tất cả vào file **data.csv**
- Quá trình lấy được thực thi ở file **get_data_to_csv.py**:

cum	sbd	toan	ngu_van	ngoai_ngu	vat_li	hoa_hoc	sinh_hoc	lich_su	dia_ly	gdcd
1	1000001	3.6	5	4				2.75	6	8.75
1	1000002	8.4	6.75	7.6				8.5	7.5	8.25
1	1000003	5.8	7.5	5				7.25	5.5	8.75
1	1000004	7.4	7.5	8.6				7.5	6.5	7.5
1	1000005	7.2	8.5	9				8	8.5	8.25
1	1000006	6.8	8.5	9.4				7	7.5	9.25
1	1000007	7.2	6	5				8	7.5	9
1	1000008	8	8	8				5.25	7	7
1	1000009	2.6	5.25					5.25	6.75	
1	1000010	8.8	8	7				6.5	6.5	6.75
1	1000011	6.4	7.25	7.6	3.5	8	3			
1	1000012	6.4	7.5					4.75	5	8

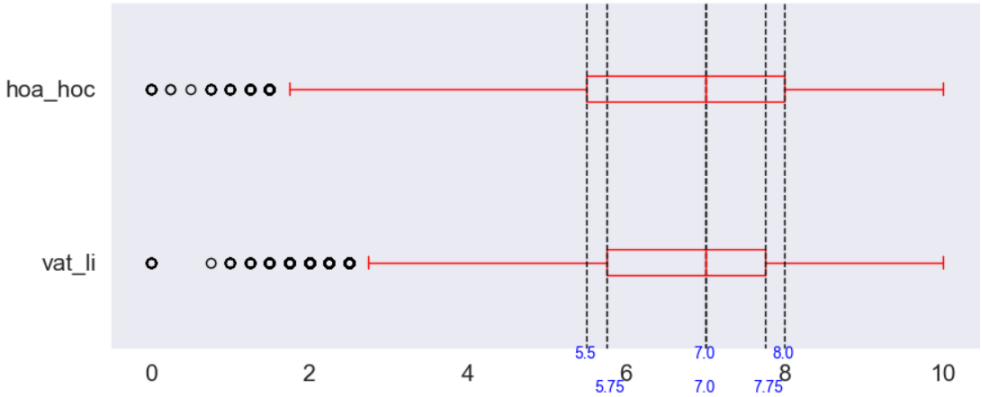
0_PREPROCESSING.IPYNB

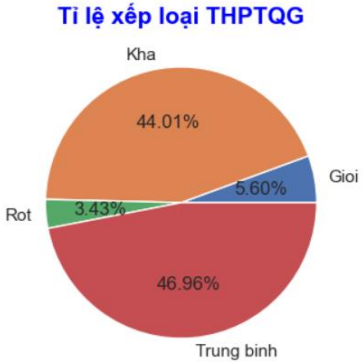
- Tiền xử lí dữ liệu, làm sạch dữ liệu
- Phân lớp các đối tượng cho dữ liệu (THPT_KHXH, THPT_KHTN, GDTX_KHXH, GDTX_KHTN, DH_RETEST, TH_RETEST)

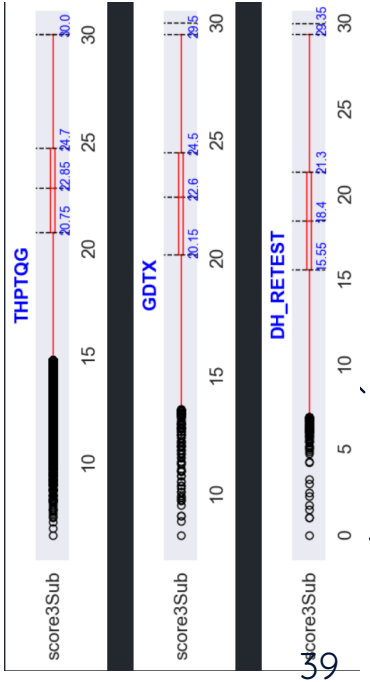
	cum	sbd	toan	ngu_van	ngoai_ngu	vat_li	hoa_hoc	sinh_hoc	lich_su	dia_ly	gdcd	phan_loai
0	1	1000001	3.6	5.00	4.0	NaN	NaN	NaN	2.75	6.00	8.75	THPT_KHXH
1	1	1000002	8.4	6.75	7.6	NaN	NaN	NaN	8.50	7.50	8.25	THPT_KHXH
2	1	1000003	5.8	7.50	5.0	NaN	NaN	NaN	7.25	5.50	8.75	THPT_KHXH
3	1	1000004	7.4	7.50	8.6	NaN	NaN	NaN	7.50	6.50	7.50	THPT_KHXH
4	1	1000005	7.2	8.50	9.0	NaN	NaN	NaN	8.00	8.50	8.25	THPT_KHXH
...	...	...	...	...	...	...	...	...	...	...	...	...
995436	64	64006584	8.4	6.75	4.6	NaN	NaN	NaN	6.50	6.75	9.00	THPT_KHXH
995437	64	64006585	5.6	6.50	2.8	NaN	NaN	NaN	6.25	6.75	8.50	THPT_KHXH
995438	64	64006586	5.8	6.00	6.6	NaN	NaN	NaN	7.25	8.00	8.00	THPT_KHXH
995439	64	64006587	7.6	6.75	7.0	NaN	NaN	NaN	8.75	7.25	9.75	THPT_KHXH
995440	64	64006588	6.6	4.50	3.2	NaN	NaN	NaN	3.00	6.00	7.50	THPT_KHXH

1_DISTRIBUTION.IPYNB

- Phân bố dữ liệu, biểu diễn dưới dạng box_plot xem mức độ khó dễ của từng môn thi
- Tính điểm trung bình tốt nghiệp, rồi phân lớp (giỏi, khá, trung bình, rớt) ở đây do có tính thêm điểm năm học lớp 12 nhưng không có data nên tạm tính trên 4.5 là đậu
- So sánh giữa các đối tượng với nhau

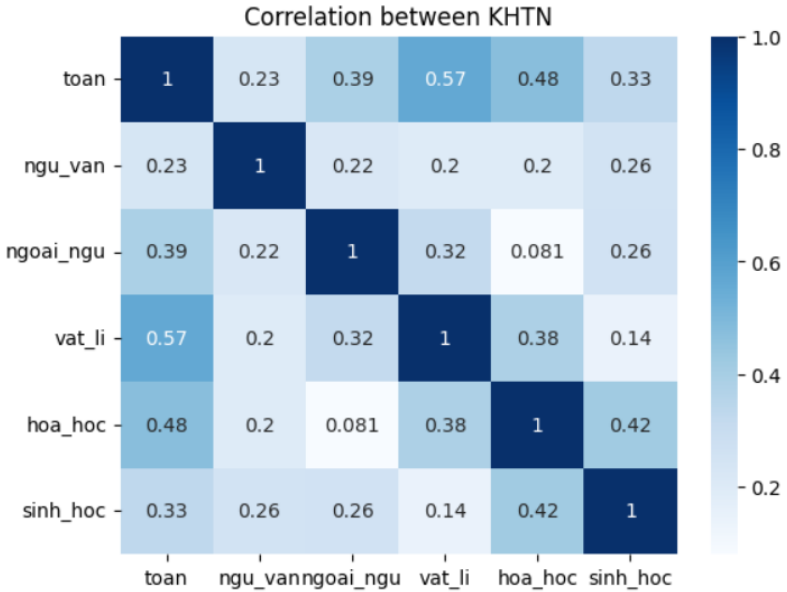


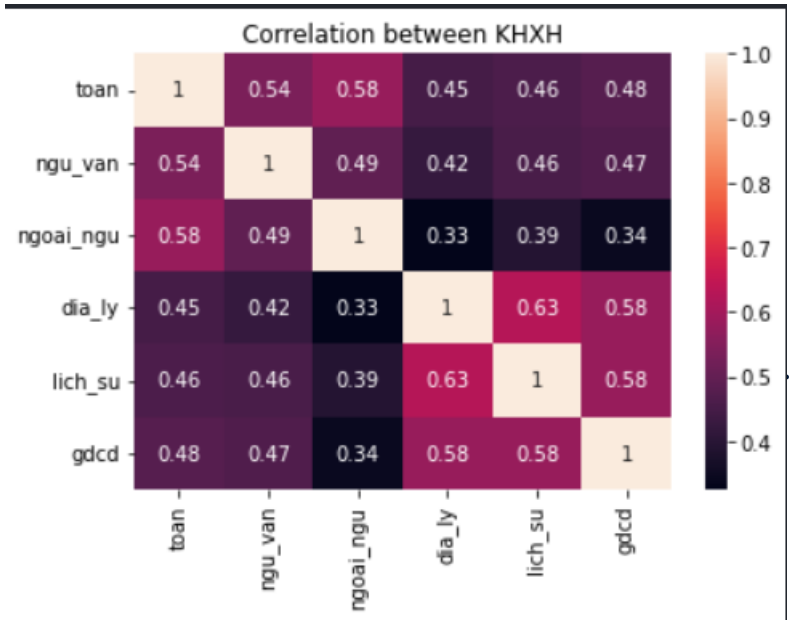




2_CORRELATION.IPYNB

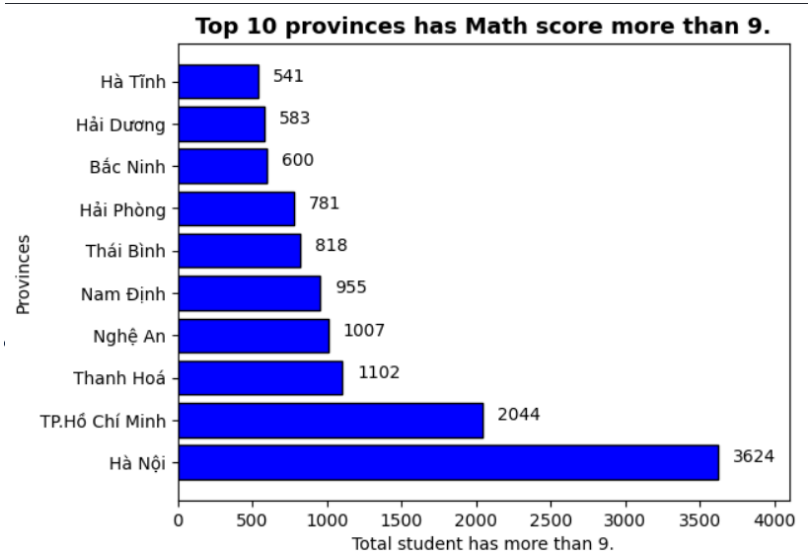
- Tìm độ tương quan giữa các môn với nhau
- Cơ bản chia ra 2 khối KHTN và KHXH

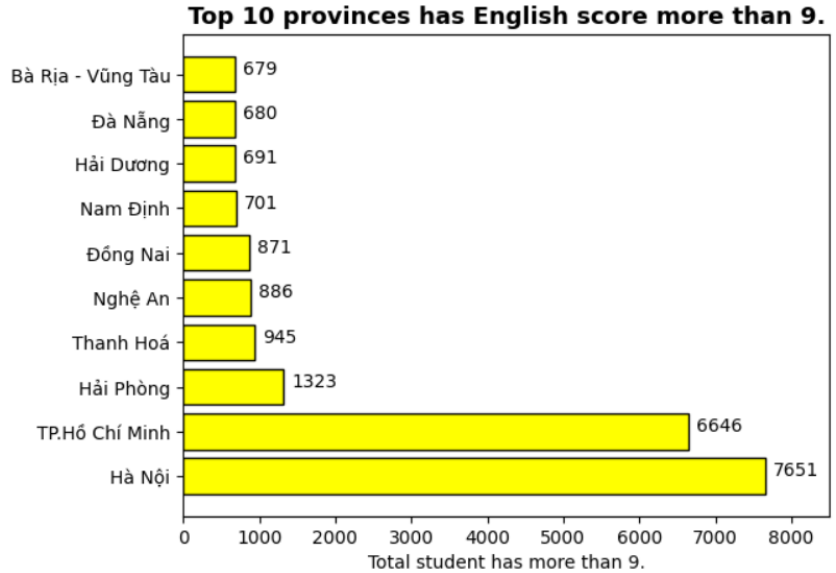




3_TOP10-STATISTICS.IPYNB

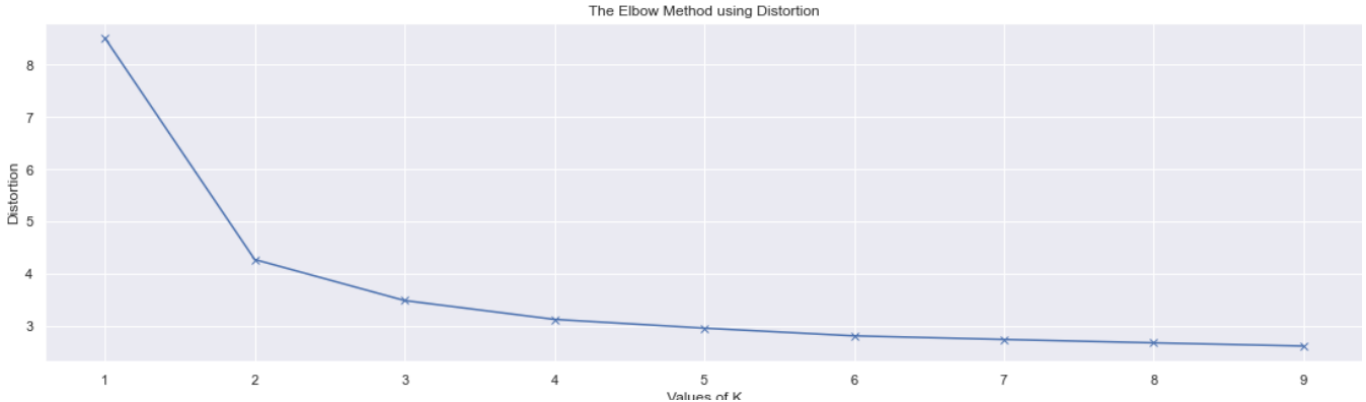
- Liệt kê 10 tỉnh/thành có thành tích tốt nhất
- Nhận xét
- Giải thích





4.1_PREDICT-KMEANS.IPYNB

- Gán nhãn khối thi cho dữ liệu
- Phân cum theo kmeans



	cum	sbd	toan	ngu_van	ngoai_ngu	vat_li	hoa_hoc	sinh_hoc	lich_su	dia_ly	gdcd	phan_loai	tong_diem_3_mon	khoi_thi
0	1	1000001	3.6	5.00	4.0	NaN	NaN	NaN	2.75	6.0	8.75	THPT_KHXH	13.75	C00
1	1	1000002	8.4	6.75	7.6	NaN	NaN	NaN	8.50	7.5	8.25	THPT_KHXH	24.40	A07
2	1	1000003	5.8	7.50	5.0	NaN	NaN	NaN	7.25	5.5	8.75	THPT_KHXH	20.25	C00
3	1	1000004	7.4	7.50	8.6	NaN	NaN	NaN	7.50	6.5	7.50	THPT_KHXH	23.50	D01
4	1	1000005	7.2	8.50	9.0	NaN	NaN	NaN	8.00	8.5	8.25	THPT_KHXH	25.00	C00

4.2_PREDICT-REGRESSION.IPYNB

- Các mô hình tuyến tính
- Sử dụng Grid Search và Optuna để đánh giá mô hình

	vat_li	hoa_hoc	Output	Predict
446204	7.75	8.00	7.4	8.098424
967293	8.50	5.25	8.4	8.340112
778919	7.00	6.75	7.6	7.639045
659572	6.25	6.50	7.0	7.364588
717937	8.75	7.50	8.6	8.420190
...	...	...	...	...
116898	5.75	4.00	4.2	6.885830
916656	4.00	2.75	6.0	5.736614
132790	7.50	7.25	7.2	7.880957
487097	7.00	8.25	8.6	8.011858
171712	5.75	5.25	6.8	6.999843

	lich_su	dia_ly	gdcd	Output	Predict
186176	4.50	4.25	5.25	3.25	4.810653
222803	6.50	5.25	8.75	4.50	6.708959
616110	7.25	8.00	8.00	6.75	6.940266
46556	6.75	7.75	8.00	7.50	6.792566
487931	5.00	5.00	8.25	4.50	6.235406
...	...	...	...	...	...
617138	4.00	5.00	6.25	5.50	5.321021
864566	7.25	8.50	8.50	8.00	7.214333
747887	5.50	6.50	8.50	6.25	6.586211
954464	6.00	6.50	8.00	5.00	6.524992
322258	7.50	6.75	9.25	7.50	7.209209

	toan	ngu_van	Output	Predict
297814	7.4	6.25	0	0
442526	8.4	8.75	1	1
363585	7.6	6.75	0	0
501353	7.0	7.50	1	0
181275	7.0	9.00	0	1
...	...	...	...	...
48920	8.6	6.25	0	0
417611	8.2	8.25	0	1
384062	7.8	7.25	0	0
225309	8.4	8.50	1	1
433475	8.2	7.75	1	1

4.3_PREDICT-STANDARDPOINT.IPYNB

- Dự đoán điểm chuẩn theo ngành năm 2022
- Dựa vào điểm chuẩn 2021 và phân bố điểm năm 2022 so với năm 2021 (xem số lượng điểm trên 8, 9 năm 2022 nhiều hay ít hơn năm 2021).

```python
def predict_score(diem2021, khoi_count, khoi_xet_tuyen, rate8, rate9):
    diem2022 = 0
    rate_count = {}
    sum_count = 0
    for khoi in khoi_xet_tuyen:
        sum_count += khoi_count[khoi]
        subjects = khoi_thi[khoi]
        rate_khoi = 0
        for sub in subjects:
            rate_khoi += rate8[sub] * 0.97 + rate9[sub] * 0.03
        rate_count[khoi] = rate_khoi / len(subjects)
    for khoi in khoi_xet_tuyen:
        diem2022 += rate_count[khoi] * khoi_count[khoi] / sum_count

    diem2022 = diem2022 * diem2021
    return diem2022
```

```python
# Ngành: Nhóm ngành CNTT - trường ĐH Khoa học tự nhiên - DHQG TP.HCM --> 27.2
diem2021 = 27.4
khoi_xet_tuyen = ['A00', 'A01', 'B08', 'D07']
diem2022 = predict_score(diem2021, khoi_count, khoi_xet_tuyen, rate_8_subject, ra
print(diem2022)
```

✓ 0.4s

```
27.174712448444073
```

```python
# Ngành: Công nghệ sinh học - trường ĐH Khoa học tự nhiên - DHQG TP.HCM -->23.75
diem2021 = 25.5
khoi_xet_tuyen = ['A02', 'B00']
diem2022 = predict_score(diem2021, khoi_count, khoi_xet_tuyen, rate_8_subject, ra
print(diem2022)
```

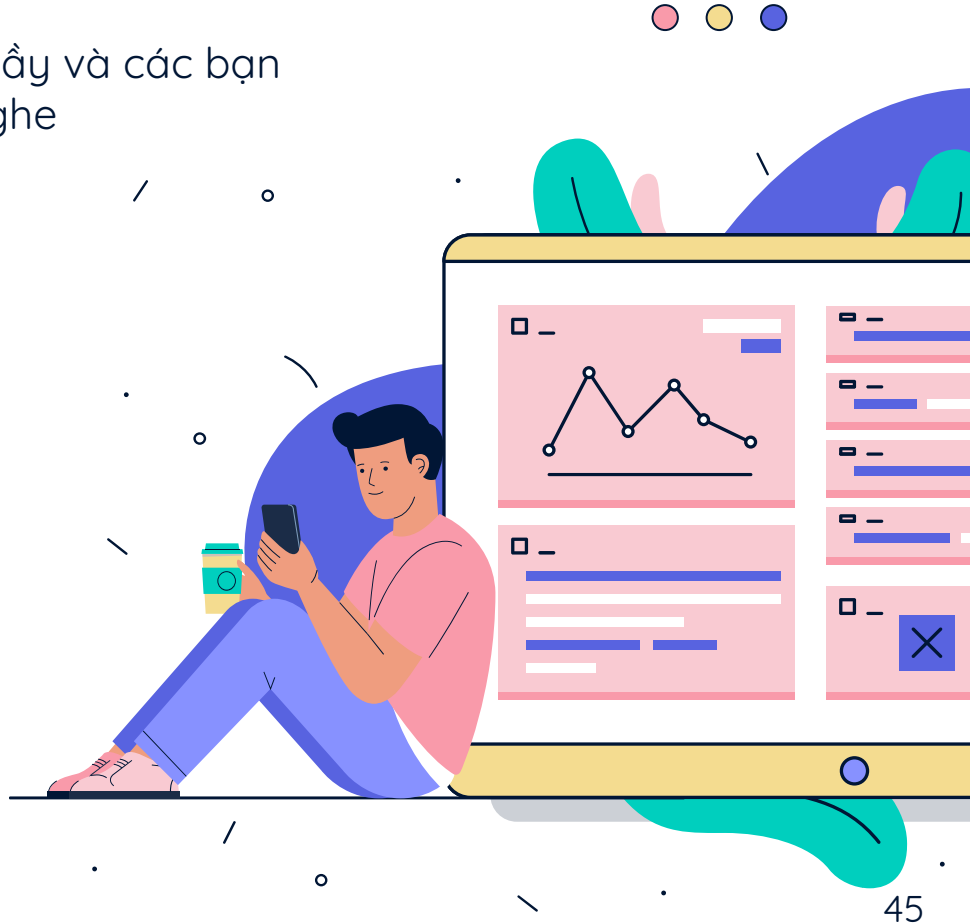
✓ 0.3s

```
22.782139949872334
```

THANKS!

Cảm ơn thầy và các bạn
đã lắng nghe

Do you have any questions?



Introduction to Data Science: Score Analysis