ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN

----------o0o----------

BÁO CÁO ĐỒ ÁN
MÔN KHAI THÁC DỮ LIỆU VÀ ỨNG DỤNG

ĐỀ TÀI:
PHÂN LOẠI KHẢ NĂNG TỐT NGHIỆP CỦA SINH VIÊN UIT

GVHD: Nguyễn Thị Anh Thư
Nhóm 9:

1. Nguyễn Văn Thành Đạt 20520436
2. Nguyễn Hoàng Gia 20520478
3. Nguyễn Thái Huy 20520547
4. Lê Ngọc Mỹ Trang 20520817
5. Nguyễn Thế Vinh 20520862

Thành phố Hồ Chí Minh, tháng 3 năm 2023

MỤC LỤC

I. Tổng quan

1. Mô tả bài toán

- Tên đề tài: Phân loại khả năng tốt nghiệp của sinh viên U.I.T.
- Thời gian thực hiện: từ tháng 5 năm 2023
- Input: dữ liệu trường UIT về thông tin sinh viên, kết quả học tập, kết quả drl, chứng chỉ anh văn, xlhv, …
- Output: gồm 2 nhãn
    - Nhãn 0: sinh viên không còn khả năng tốt nghiệp: du học, bỏ học, xử lý học vụ…
    - Nhãn 1 sinh viên có khả năng tốt nghiệp

2. Giới thiệu bài toán

Hiện nay, nhu cầu về nhân lực lao động có trình độ cao của xã hội ngày càng tăng, rất nhiều người chọn con đường học đại học. Một số người có những định hướng và kế hoạch rõ ràng cho con đường đó của họ. Tuy nhiên, vẫn có những trường hợp cảm thấy mông lung, không biết khả năng mình đến đâu để phấn đấu tốt nghiệp để có cho mình tấm bằng đại học. Nhiều trường hợp trở nên ỷ lại vào quá khứ, khiến thời gian của họ bị lãng phí, rồi dần dần vụt mất cơ hội có cho mình tấm bằng đại học. Giả sử có một công cụ nào đó có thể giúp theo dõi và biết được khả năng tốt nghiệp của bản thân qua từng giai đoạn, trở thành hồi chuông cảnh tỉnh cũng như là động lực để sinh viên phấn đấu. Không chỉ vậy, nhà trường và các giảng viên cũng có thể sử dụng công cụ dự đoán này để tham khảo, điều chỉnh, cân đối các hoạt động giảng dạy và phong trào vui chơi rèn luyện.

Từ giả thuyết đó, nhóm chúng em đã nảy ra đề tài “Phân loại khả năng tốt nghiệp của sinh viên U.I.T” để hiện thực hóa công cụ trên. Bài toán dựa trên ý tưởng sử dụng các thông tin và đặc trưng liên quan của sinh viên UIT đến từ bộ dữ liệu có sẵn để xây dựng một mô hình dự đoán ước lượng xác suất tốt nghiệp thành công của sinh viên. Việc phân loại khả năng tốt nghiệp cũng có thể giúp phát hiện sớm các sinh viên có nguy cơ bị rớt hoặc bỏ học, từ đó đưa ra các biện pháp can thiệp và hỗ trợ kịp thời. Đề tài giúp đưa ra thông tin quan trọng về khả năng tốt nghiệp của

sinh viên, hỗ trợ quyết định của các nhà trường, các cơ quan quản lý giáo dục, và cả sinh viên trong việc lựa chọn các biện pháp hỗ trợ học tập và nghề nghiệp.

3. Thách thức

Hiện tại đã có nhiều nghiên cứu về đề tài này như "Phân loại khả năng tốt nghiệp của học sinh Philippines" [9] hoặc là "Mô hình nghiên cứu phân loại thành tích tốt nghiệp của sinh viên của trường đại học Telkom" [10], trong nước nói chung chưa có ứng dụng chính thức nào được công khai tính tới thời điểm hiện tại, tuy nhiên có một vài dự án nghiên cứu nhỏ liên quan đến việc đoán kết quả học tập hoặc tỷ lệ có việc làm của sinh viên. Tuy nhiên do đặc điểm, tính chất khác nhau của từng môi trường giảng dạy nên dữ liệu của mỗi nơi mỗi khác, không thể áp dụng để đánh giá xác suất tốt nghiệp cho sinh viên UIT được.

Dữ liệu thô sau khi được thu thập thường cần được xử lý và làm sạch để phù hợp với mô hình hoặc phân tích trước khi có thể sử dụng. Bên cạnh đó, việc áp dụng các phương pháp phân tích dữ liệu phù hợp để trích xuất thông tin, khám phá mẫu thú vị là rất cần thiết trước khi đưa ra kết luận hợp lý. Ngoài ra một thách thức khác là sự hạn chế về thời gian, nguồn lực và công nghệ. Điều này có thể ảnh hưởng đến quy mô và phạm vi nghiên cứu, khả năng tiếp cận dữ liệu, cũng như khả năng triển khai các giải pháp và công nghệ mới.

4. Đối tượng, phạm vi và mục tiêu đề tài

Đề tài được thực hiện trong phạm vi trường Đại học Công nghệ Thông tin (UIT), đối tượng hướng tới là sinh viên của trường UIT.

Mỗi quốc gia, mỗi trường học đều có đặc thù riêng. Vì vậy, mục tiêu của đề tài là thực hiện xây dựng mô hình phân loại khả năng tốt nghiệp với dữ liệu của trường U.I.T, việc này sẽ giúp tăng tính cá nhân hóa cho nhà trường. Ngoài ra có thể dự đoán tỉ lệ sinh viên không đủ điều kiện tốt nghiệp. Từ đó, phân tích các yếu tố ảnh hưởng đến xác suất tốt nghiệp của sinh viên U.I.T sẽ giúp trường có những biện pháp can thiệp kịp thời và phát triển các phương pháp triển khai nhằm nâng cao tỉ lệ tốt nghiệp của trường.

II. Mô hình bài toán

1. Các bước tiền xử lý dữ liệu

1.1. Thu thập dữ liệu:

- Dữ liệu thô được thu thập dựa trên nguồn nội bộ, cụ thể là các thông tin về lý lịch và quá trình, kết quả học tập của sinh viên trường đại học Công nghệ thông tin từ năm 2006 đến 2022 (thông tin sinh viên giữa các bảng có thể khác nhau về khoảng năm học của sinh viên).

Bộ dữ liệu gồm 15 bảng: '10.diemrl.xlsx', '06.giayxacnhan.xlsx', 'uit_hocphi_miengiam.xlsx', '05.ThiSinh.xlsx', '03.sinhvien_chungchi.xlsx', '08.XLHV.xlsx', '01.sinhvien.xlsx', 'sinhvien_dtb_toankhoa.xlsx', 'sinhvien_dtb_hocky.xlsx', 'diemrl.xlsx', 'diem_Thu.xlsx', '12.baoluu.xlsx', '04.xeploaiav.xlsx', '02.diem.xlsx', '14.totnghiep.xlsx’	Tiến hành sàn chọn các bảng dữ liệu có liên quan đến vấn đề cần giải quyết: '04.xeploaiav.xlsx', '14.totnghiep.xlsx', '01.sinhvien.xlsx', 'sinhvien_dtb_toankhoa.xlsx', 'sinhvien_dtb_hocky.xlsx', 'diemrl.xlsx', '03.sinhvien_chungchi.xlsx', '08.XLHV.xlsx’, '05.ThiSinh.xlsx', 'uit_hocphi_miengiam.xlsx'

- Trước khi thực hiện làm sạch dữ liệu, ta có một số phân tích về dữ liệu thô như sau:
    - Dữ liệu thô chứa nhiều hàng (cá nhân sinh viên) trùng lặp, trống ở một số thông tin. Đặc biệt dữ liệu có các cột (thông tin của sinh viên) không mang bất cứ ý nghĩa gì và có thể là lỗi trong quá trình nhập thông tin.
    - Các bảng có cột trạng thái nhưng không có ghi chú kèm theo định nghĩa phân lớp cho các trạng thái này.
    - Dữ liệu gặp nhiều lỗi về sự thiếu nhất quán ảnh hưởng đến kết quả cuối cùng khi thực hiện thống kê. Các lỗi này bao gồm: thông tin cung cấp không khớp với đầu vào mong muốn (chẳng hạn khi được yêu cầu

nhập tỉnh/thành phố, sinh viên cung cấp quận 12 hoặc cung cấp cả thông tin về huyện và thành phố cách nhau bởi dấu phẩy “,” dẫn đến khi đọc file sẽ bị nhầm thành hai thông tin khác nhau), dữ liệu cùng một nghĩa nhưng khi diễn giải lại khác nhau (chẳng hạn các cột có chứa giá trị NULL nhưng có thể ghi theo các cách khác nhau là “NULL”, “NULL)”, “NULL);”), các thông tin có thể không thỏa điều kiện yêu cầu. Các chữ cùng nghĩa nhưng có khi được viết hoa chữ đầu hoặc không (ví dụ: TB Khá và TB khá)

- Thông tin dùng để phân loại trong dữ liệu thô được cung cấp ở dạng số tự nhiên hoặc dưới dạng mã (mã ngành)
- Định dạng ngày và giờ không thống nhất với nhau (ví dụ: 4/18/2017, 2017-10-26)
- Ngoài ra có các lỗi về chính tả, cú pháp và các khoảng trắng không cần thiết. Chẳng hạn ở tên các cột thường hay có khoảng trắng ở phía trước một vài tên.

- Xử lý khoảng trắng và các lỗi cú pháp
  - Loại bỏ phần khoảng trắng trước tên trường của các bảng: sử dụng hàm replace thay thế các khoảng trắng trong tên trường bằng rỗng, tránh gây khó khăn khi gọi tên.
  - Sửa lỗi sai do nhầm lẫn về dấu phẩy ngăn cách các cột với dấu phẩy trong chuỗi khi đưa dữ liệu vào tạo thành các cột thừa, tạo thành các giá trị sai lầm ở các cột sau.
- Chuẩn hóa cấu trúc và định dạng dữ liệu.
  - Các cột về thời gian đôi khi thông tin ghi nhận bị lỗi và ghi thành 0000-00-00 00:00:00. Chuyển toàn bộ thời gian về theo định dạng yyyy-mm-dd hh:mm:ss hoặc yyyy-mm-dd nếu chỉ có ngày.
  - Chỉnh lại kiểu dữ liệu ứng với từng cột sau khi sửa xong
  - Xóa dữ liệu trùng lặp và bất thường: Nhằm xóa các dữ liệu trùng lặp, ta sử dụng hàm drop_duplicates() từ thư viện pandas.

- Điều chỉnh sự không nhất quán.
    - Sử dụng hàm sub() của thư viện re để loại bỏ các dấu câu và ký tự đặc biệt cho các cột (lưu ý chọn cột để loại bỏ, tránh các cột mà ký tự đặc biệt là một phần dữ liệu, ví dụ: MMT&TT, IT002.G21.HTCL, mã số sinh viên (đã mã hóa),...).
    - Ở bảng sinhvien, tính từ cột diachi_tinhtp trở về sau, có thể nối toàn bộ các chuỗi trong các cột lại tạo thành chuỗi duy nhất và chọn (hoặc nội suy) từ cột đó thành thông tin tỉnh thành.
    - Ở bảng sinhvien_chungchi, loại bỏ tên cột url_1 và loaixn_2, chuyển tên cột từ listening lên 2 cột thay thế cho hai cột vừa xóa
    - Ở bảng XLHV, hai cột cuối (ngayqd và Column1) là thông tin về ngày quyết định xử lý học vụ, tuy nhiên đã bị tách ra thành hai cột, một có thông tin và một không.
- Tìm kiếm các thông tin có chức năng như nhau nhưng viết theo các cách khác nhau:
    - Chuyển các cách viết các thông tin được để trống thành NULL.
    - Chuyển các cột có chứa giá trị NULL nhưng có thể ghi theo các cách khác nhau là “NULL”, “NULL)”, “NULL);”) về thành NULL

1.2. Chọn lọc các đặc trưng quan trọng:

- Bảng ‘01.sinhvien’: Chọn các thuộc tính, trong đó:
    - 'mssv': để liên kết với các bảng khác nhằm trích xuất dữ liệu
    - 'namsinh', 'gioitinh', 'noisinh', 'lopsh', 'khoa', 'hedt', 'khoahoc', 'chuyennganh2’ là các thuộc tính cần xét
    - 'tinhtrang': căn cứ để gán nhãn cho sinh viên là đã tốt nghiệp hay không (3: đã tốt nghiệp, khác 3 là chưa hoặc không tốt nghiệp: cần dựa vào dữ liệu từ bảng xử lý học vụ)
- Bảng ‘03.sinhvien_chungchi’: Chọn các thuộc tính, trong đó:
    - 'mssv': để liên kết với các bảng khác nhằm trích xuất dữ liệu
    - 'loaixn', 'tongdiem' là các thuộc tính cần xét

    - 'trangthai': căn cứ để xem xét bằng của sinh viên có đạt yêu cầu hay không
- Bảng ‘08.XLHV’: Chọn các thuộc tính, trong đó:
    - 'mssv': để liên kết với các bảng khác nhằm trích xuất dữ liệu
    - ‘lydo’: xem xét để lọc ra các sinh viên bị buộc thôi học
- Bảng ‘05.ThiSinh’: Chọn các thuộc tính, trong đó:
    - 'mssv': để liên kết với các bảng khác nhằm trích xuất dữ liệu
    - 'dien_tt' là các thuộc tính cần xét
- Bảng ‘sinhvien_dtb_toankhoa’: Chọn các thuộc tính, trong đó:
    - 'mssv': để liên kết với các bảng khác nhằm trích xuất dữ liệu
    - 'dtb_toankhoa', 'dtb_tichluy', 'sotc_tichluy' là các thuộc tính cần xét
- Bảng ‘sinhvien_dtb_hocky’: Chọn các thuộc tính, trong đó:
    - 'mssv': để liên kết với các bảng khác nhằm trích xuất dữ liệu
    - 'hocky', 'namhoc', 'dtbhk', 'sotchk' là các thuộc tính cần xét
- Bảng ‘drl’: Chọn các thuộc tính, trong đó:
    - 'mssv': để liên kết với các bảng khác nhằm trích xuất dữ liệu
    - 'hocky', 'namhoc', 'drl' là các thuộc tính cần xét
- Bảng ‘04.xeploaiav’: Chọn các thuộc tính, trong đó:
    - ‘mssv': để liên kết với các bảng khác nhằm trích xuất dữ liệu
    - 'total', 'mamh' là các thuộc tính cần xét
- Bảng ‘14.totnghiep’: Chọn 'mssv' để liên kết với các bảng khác nhằm trích xuất dữ liệu
- Bảng ‘uit_hocphi_miengiam.xlsx’: Chọn 'mssv' để liên kết với các bảng khác nhằm trích xuất dữ liệu

1.3. Tiến hành gán nhãn:

- Sau khi đã lọc ra các bảng và các cột ở từng bảng, ta tiến hành gán nhãn tập dữ liệu bằng cách
    - Lọc ra các sinh viên của bảng ‘01.sinhvien’ có trong bảng ‘14.totnghiep’ và gán nhãn là 1 cho thấy sinh viên này đã tốt nghiệp

- Lọc ra các sinh viên của bảng ‘01.sinhvien’ có trong bảng ‘08.XLHV’ với lý do bị cho thôi học và gán nhãn là 0 cho thấy sinh viên này không tốt nghiệp

- Sau bước này, ta sẽ có 2 dataframe gồm 1845 sinh viên đã tốt nghiệp và 340 sinh viên không tốt nghiệp. Tiếp tục tiến hành các bước xử lý:
  - Ghép 2 kết quả trên thành 1 dataframe, đặt tên cột nhãn là ‘label’
  - Tiến hành xóa các phần tử trùng lặp, do mssv là duy nhất nên xóa trùng theo ‘mssv’
- Kết quả đạt được: bảng sv_fly gồm 2183 dòng và 11 cột ('mssv', 'namsinh', 'gioitinh', 'noisinh', 'lopsh', 'khoa', 'hedt', 'khoahoc', 'chuyennganh2', 'tinhtrang', 'label')

1.4. Chuẩn bị dữ liệu:

a. Bảng ‘05.ThiSinh’

- Dựa vào ‘mssv’, ta kết hợp dữ liệu từ các cột của bảng ‘05.ThiSinh’ với dữ liệu bảng sv_fly đã gán nhãn ở bước trên
- Xử lý giá trị NULL của cột ‘dien_tt’:
  - Duyệt qua danh sách ‘dien_tt’, trả về giá trị thuộc ‘dien_tt’ khác NULL xuất hiện nhiều nhất
  - Thay thế những vị trí mà giá trị ‘dien_tt’ bị NULL thành giá trị xuất hiện nhiều nhất vừa tìm được ở trên
- Phân cấp dữ liệu cột ‘dien_tt’:
  - Trước tiên cần xác định sẽ phân cấp dữ liệu trên vào các khoảng như: xuất sắc (A), giỏi (B), khá (C)
  - Do có nhiều ‘dien_tt’ khác nhau trong cột, cần đưa chúng về một hệ quy chiếu để có thể dễ dàng so sánh, gồm có:
    - 'THPT': xét dựa trên kết quả quá trình học THPT→B
    - 'CUTUYEN': cử tuyển là việc tuyển sinh qua phương thức xét tuyển vào đại học, cao đẳng, trung cấp đối với người học được quy định như sau: Người dân tộc thiểu số rất ít người, Người

dân tộc thiểu số ở vùng có điều kiện kinh tế - xã hội đặc biệt khó khăn chưa có hoặc có rất ít đội ngũ cán bộ, công chức, viên chức là người dân tộc thiểu số→C

- 'TT-Bộ', 'ƯT-Bộ': thí sinh tham dự các cuộc thi về khoa học, kỹ thuật, ngoại ngữ … và đạt giải→A
- 'ƯT-ĐHQG': ưu tiên xét tuyển theo quy định của ĐHQG→B
- 'ĐGNL': xét tuyển dựa trên điểm ĐGNL→B

- Dựa vào các nhãn đã quy ước ở trên để tiến hành thay thế các giá trị trong cột 'dien_tt'

b. Bảng '04.xeploaiav'

- Dựa vào 'mssv', ta kết hợp dữ liệu từ các cột của bảng '04.xeploaiav' với dữ liệu của dataframe ở bước trên
- Thay thế các giá trị NULL trong cột 'total' bằng 0
- Định dạng lại kiểu dữ liệu trong cột 'mamh' cho thống nhất do có nhiều giá trị ở dạng chuỗi có thêm dấu cách ở đầu và cuối, việc ta cần làm là loại bỏ dấu cách.
- Cập nhật lại 'mamh' cho các sinh viên có 'total' bằng 0 thành 'AVSC1', tức là sinh viên không tham gia thi anh văn đầu vào, hoặc điểm thi anh văn đầu vào bằng 0 và loại bỏ cột 'total' ra khỏi bảng

c. Bảng 'sinhvien_dtb_hocky'

- Group 'dtbhk' theo 2 cột 'mssv' và 'namhoc'
- Chia theo từng kỳ học tăng dần bắt đầu từ 1
  - Drop các giá trị NULL, gom hết các giá trị khác NULL của các cột điểm theo 'namhoc' thành 1 mảng số
  - Chia các mảng điểm ta vừa gom lại được thành các cột theo độ dài của mảng điểm dài nhất trong các mảng, với mỗi cột được đánh số tăng dần bắt đầu từ hk1, các mảng nào không đủ giá trị để điền vào các cột mới được tạo thì thay thành None

- Tuy ta được 9 học kỳ, nhưng vì cột hk9 chỉ có 2 giá trị khác None và chương trình đào tạo của trường phổ biến là 4 năm nên ta xóa cột này

- Thay các giá trị None thành giá trị -1
- Dựa vào ‘mssv’, ta kết hợp dữ liệu từ các cột của bảng ‘sinhvien_dtb_hocky’ đã được xử lý ở trên với dữ liệu ở bước trước

d. Bảng ‘drl’

- Thay thế drl nhỏ hơn 0 thành 0 và lớn hơn 100 thành 100
- Group ‘drl’ theo 2 cột ‘mssv’ và ‘namhoc’
- Chia theo từng kỳ học tăng dần bắt đầu từ 1
    - Drop các giá trị NULL, gom hết các giá trị khác NULL của các cột điểm theo ‘namhoc’ thành 1 mảng số
    - Chia các mảng điểm ta vừa gom lại được thành các cột theo độ dài của mảng điểm dài nhất trong các mảng, với mỗi cột được đánh số tăng dần bắt đầu từ hk1, các mảng nào không đủ giá trị để điền vào các cột mới được tạo thì thay thành None
- Dựa vào ‘mssv’, ta kết hợp dữ liệu từ các cột của bảng ‘drl’ đã được xử lý ở trên với dữ liệu ở bước trước
- Điền các giá trị None:
    - Tính drl trung bình của mỗi sinh viên bằng danh sách điểm sẵn có khác None
    - Đối với các sinh viên có nhãn bằng 0, thay thế None bằng giá trị 0 đề phòng trường hợp người đó không học ở hk này
    - Đối với các sinh viên có nhãn bằng 1, thay thế None bằng giá trị drl trung bình của chính họ đã tìm được ở trên
    - Loại bỏ cột drl trung bình

e. Bảng ‘03.sinhvien_chungchi’

- Dựa vào ‘mssv’, ta kết hợp dữ liệu từ các cột của bảng ‘03.sinhvien_chungchi’ với dữ liệu ở bước trên

- Cột ‘trangthai_av’ chứa giá trị thể hiện rằng sinh viên đã có bằng ngoại ngữ đủ chuẩn đầu ra chưa
- Cập nhật giá trị của cột ‘trangthai_av’:
    - Những sinh viên nào đã tốt nghiệp mà chưa được cập nhật chuẩn đầu ra ngoại ngữ (NULL) thì ta gán giá trị cho cột ‘trangthai_av’ bằng 1, tức là đã có bằng ngoại ngữ đạt chuẩn đầu ra
    - Những sinh viên nào không tốt nghiệp và chưa có thông tin liên quan đến chuẩn đầu ra ngoại ngữ (NULL) thì ta gán giá trị cho cột ‘trangthai_av’ bằng 0, tức là chưa có bằng ngoại ngữ đạt chuẩn đầu ra
- Cập nhật giá trị của cột ‘loaixn’:
    - Duyệt qua danh sách ‘loaixn’, trả về giá trị thuộc ‘loaixn’ khác NULL xuất hiện nhiều nhất
    - Những sinh viên nào có giá trị cột ‘trangthai_av’ bằng 1, thay thế những nơi mà giá trị ‘loaixn’ bị NULL thành giá trị xuất hiện nhiều nhất vừa tìm được ở trên
    - Những sinh viên nào có giá trị cột ‘trangthai_av’ bằng 0, thay thế những nơi mà giá trị ‘loaixn’ bị NULL thành giá trị ‘Khong co’, tức là sinh viên này chưa có chứng chỉ ngoại ngữ nào
- Cập nhật giá trị của cột ‘tongdiem_av’:
    - Đối với mỗi loại tín chỉ thuộc cột ‘loaixn’, lưu lại số điểm cao nhất tương ứng ở cột ‘tongdiem_av’
    - Đối với một sinh viên có ‘trangthai_av’ bằng 1 nhưng giá trị ‘tongdiem_av’ bằng 0 hoặc NULL, thay thế giá trị đó bằng điểm số cao nhất dựa trên ‘loaixn’ của người đó và danh sách điểm số cao nhất ứng với mỗi ‘loaixn’ vừa tìm được ở trên
- Tiến hành cập nhật lại kiểu dữ liệu cho trường ‘trangthai_av’ để thống nhất giá trị

f. Bảng ‘uit_hocphi_miengiam’

- Đếm số lần 1 sinh viên được miễn giảm học phí trong suốt quá trình học

- Dựa vào ‘mssv’, ta kết hợp dữ liệu từ các cột của bảng ‘uit_hocphi_miengiam’ với dữ liệu ở bước trên
- Thay thế các giá trị NULL thành 0, tức là người đó không được miễn giảm học phí lần nào

g. Bảng ‘XLHV_8’

- Đếm số lần 1 sinh viên được bị xử lý học vụ trong suốt quá trình học
- Dựa vào ‘mssv’, ta kết hợp dữ liệu từ các cột của bảng ‘XLHV_8’ với dữ liệu ở bước trên
- Thay thế các giá trị NULL thành 0, tức là người đó không bị xử lý học vụ lần nào

h. Bảng ‘sinhvien_dtb_toankhoa’

- Dựa vào giá trị của cột ‘mssv’, ta kết hợp dữ liệu từ các cột của bảng ‘sinhvien_dtb_toankhoa’ với dữ liệu ở bước trên
- Chuyển cột ‘label’ về cuối bảng
- Xóa cột ‘tinhtrang’

2. Các thuộc tính được sử dụng

STT	Tên thuộc tính	Ý nghĩa	Phạm vi	Kiểu dữ liệu
1	namsinh	Năm sinh	1985, 2001	int64
2	gioitinh	Giới tính	0, 1	int64
3→ 10	mssv, noisinh, lopsh, khoa, hedt, chuyennganh2, mamh, loaixn	Mã số sinh viên, Nơi sinh, Lớp sinh hoạt, Khoa, Hệ đào tạo, Chuyên ngành, Xếp lớp anh văn, Loại bằng ngoại ngữ	_	object
11	khoahoc	Khóa học	8, 14	int64
12	dien_tt	Diện trúng tuyển	A, B, C	object
13	diem_hk1	Điểm trung bình học kỳ 1	0.0, 9.7	float64
14	diem_hk2	Điểm trung bình học kỳ 2	-1.0, 9.36	float64

15	diem_hk3	Điểm trung bình học kỳ 3	1.0, 9.66	float64
16	diem_hk4	Điểm trung bình học kỳ 4	-1.0, 9.9	float64
17	diem_hk5	Điểm trung bình học kỳ 5	-1.0, 10.0	float64
18	diem_hk6	Điểm trung bình học kỳ 6	-1.0, 9.3	float64
19	diem_hk7	Điểm trung bình học kỳ 7	-1.0, 8.7	float64
20	diem_hk8	Điểm trung bình học kỳ 8	-1.0, 6.7	float64
21→27	drl_hk1, drl_hk2, drl_hk3, drl_hk4, drl_hk5, drl_hk6, drl_hk7	Điểm rèn luyện học kỳ 1 đến 7	10, 100	int64
28	dtb_toankhoa	Điểm trung bình toàn khóa	0.0, 9.22	float64
29	dtb_tichluy	Điểm trung bình tích lũy	0.0, 9.22	float64
30	sotc_tichluy	Số tín chỉ tích lũy	0, 177	int64
31	tongdiem_av	Tổng điểm thi anh văn	0.0, 990.0	float64
32	trangthai_av	Trạng thái anh văn	0, 1	int64
33	sl_giam	Số lần được miễn giảm học phí	0, 6	int64
34	xlhv	Số lần bị xử lý học vụ	0, 6	int64
35	tinhtrang	Tình trạng học vụ của sinh viên	3, 9	in64

Bảng 1 - Các thuộc tính được sử dụng

3. Phương pháp đề xuất

3.1. Tên phương pháp

Sử dụng các thuật toán phân lớp máy học để tiến hành phân loại khả năng tốt nghiệp của sinh viên dựa trên thông tin của học kỳ học kỳ có sẵn và tăng cường dữ liệu theo chiều ngang từ dữ liệu các học kỳ mới

3.2. Các đặc trưng chính của phương pháp

- Học có giám sát: Hầu hết các thuật toán phân lớp là thuộc loại học có giám sát, tức là chúng được huấn luyện trên một tập dữ liệu có nhãn, mỗi mẫu dữ liệu trong tập huấn luyện có một nhãn tương ứng.

- Tập huấn luyện: Một thuật toán phân lớp cần được huấn luyện trên một tập dữ liệu được gọi là tập huấn luyện bao gồm các mẫu dữ liệu đã được gán nhãn, từ đó học cách phân lớp các mẫu mới.
- Tính toán đặc trưng: Trước khi áp dụng thuật toán phân lớp, thường cần thực hiện quá trình trích xuất đặc trưng từ dữ liệu đầu vào. Đặc trưng này có thể là các thuộc tính số (ví dụ: chiều cao, trọng lượng) hoặc các đặc trưng rời rạc (ví dụ: màu sắc, từ ngữ).
- Mô hình phân lớp: Thuật toán phân lớp xây dựng một mô hình dự đoán nhãn lớp của các mẫu dữ liệu không được gán nhãn. Mô hình này có thể có nhiều dạng khác nhau, chẳng hạn như Decision Tree, Support Vector Machine (SVM), Neural Network….
- Tiêu chí học: Mỗi thuật toán phân lớp có một tiêu chí học riêng để tìm ra mô hình tốt nhất.
- Phương pháp tối ưu: Để tìm mô hình tốt nhất, các thuật toán phân lớp thường sử dụng các phương pháp tối ưu hóa như gradient descent, quy hoạch tuyến tính, hoặc các phương pháp tiến hóa.
- Đánh giá hiệu suất: Sau khi huấn luyện, thuật toán phân lớp cần được đánh giá hiệu suất của nó trên các dữ liệu kiểm tra. Các độ đo thông thường bao gồm Accuracy, Precision, Recall, độ F1 và Confusion matrix.
- Overfitting và underfitting: Một vấn đề phổ biến trong thuật toán phân lớp là overfitting (quá khớp) và underfitting (quá khớp không đủ). Overfitting xảy ra khi mô hình quá phức tạp và "nhớ" các mẫu huấn luyện, trong khi underfitting xảy ra khi mô hình quá đơn giản và không thể học được các mẫu huấn luyện một cách chính xác.

3.3. Độ đo

a. Confusion matrix

- Tổng quan về confusion matrix: Confusion Matrix (ma trận nhầm lẫn hay ma trận lỗi) là một bố cục bảng cụ thể cho phép hình dung hiệu suất của một thuật toán, được sử dụng rộng rãi cho các mô hình phân loại.
- Thông thường, confusion matrix sẽ có dạng như sau:

	Positive	Negative
Positive	True Positive (TP)	False Positive (FP)
Negative	False Negative (FN)	True Negative (TN)

Bảng 2 - Confusion matrix

- Trong đó:
    - True Positive: Số nhãn dự đoán "có" là đúng
    - False Positive: Số nhãn dự đoán "có" là sai
    - False Negative: Số nhãn dự đoán "không" là sai
    - True Negative: Số nhãn dự đoán "không" là đúng

b. Accuracy

Ý tưởng: Accuracy là độ đo cho phép tính số nhãn được dự đoán đúng trên toàn bộ tập dữ liệu

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN}$$

c. Confusion matrix

Ý tưởng: Về cơ bản, độ đo Precision cho phép người đánh giá hiểu được tỉ lệ các trường hợp được dự đoán là "có" *chính xác* bao nhiêu. Như vậy nếu Precision càng thấp, chứng tỏ mô hình có thiên hướng gán sai nhãn cho các trường hợp 0 trong thực tế và ngược lại.

$$Precision = \frac{TP}{TP + FP}$$

d. Recall

Ý tưởng: Recall hỗ trợ việc đánh giá xem mô hình có *bỏ sót* các trường hợp "có" nào không. Tức là nêu Recall càng thấp, nhiều nhãn "có" trong thực tế đã bị gán nhầm và ngược lại.

$$Recall = \frac{TP}{TP + FN}$$

e. F1-score

Ý tưởng: Để đánh giá mô hình phân lớp một cách khách quan nhất, trong trường hợp lý tưởng, ta cần kết hợp kết quả đánh giá của cả Precision và Recall. Như vậy, F1 score là tổng hòa của hai độ đo kể trên.

$$F1\ score = \frac{2 * Precision * Recall}{Precision + Recall}$$

3.4. Thuật toán máy học

Nhóm đề xuất thử nghiệm bộ dữ với các mô hình phân loại từ đơn giản cho đến phức tạp, để có thể đánh giá độ hiệu quả của từng mô hình đối với tập dữ liệu.

a. Naive Bayes [4]

- Ý tưởng: Thuật toán phân lớp Naïve Bayes dựa trên công thức Bayes, được sử dụng để tính xác suất của một lớp dựa trên các đặc trưng của dữ liệu. Công thức Bayes là một công thức toán học đơn giản để tính xác suất của một biến ngẫu nhiên dựa trên các thông tin liên quan đến nó. Ý tưởng của thuật toán Naïve Bayes là tính toán xác suất của mỗi lớp và chọn lớp có xác suất cao nhất cho đầu vào.
- Ưu điểm:
    - Đơn giản và nhanh chóng
    - Hiệu suất tốt với các bộ dữ liệu lớn
    - Hiệu quả trong việc phân loại văn bản
    - Không yêu cầu nhiều dữ liệu huấn luyện
    - Khả năng xử lý dữ liệu nhiễu tốt
- Nhược điểm:
    - Giả định giá trị các thuộc tính là độc lập với nhau
    - Có thể xảy ra hiện tượng quá khớp (overfitting
    - Không thể xử lý các giá trị bị thiếu (missing values
    - Có thể bị ảnh hưởng bởi các thuộc tính không liên quan

b. Logistic Regression [1]

- Ý tưởng: Logistic Regression là một thuật toán phân lớp dựa trên mô hình hồi quy logistic. Ý tưởng của thuật toán là tìm một đường cong S-shaped (hàm sigmoid) để phân loại dữ liệu thành hai lớp 0 và 1.
- Ưu điểm:
    - Đơn giản, dễ hiểu bởi bất kỳ ai có kiến thức cơ bản về toán học và thống kê.
    - Tính toán nhanh chóng và dễ dàng.

- Được sử dụng cho cả bài toán phân loại nhị phân và phân loại đa lớp.
- Không yêu cầu giả định về phân phối của dữ liệu đầu vào, điều này làm cho nó có thể áp dụng cho nhiều loại dữ liệu khác nhau.

- Nhược điểm
  - Dữ liệu chứa nhiễu hoạc mất cân bằng có thể ảnh hưởng đến độ chính xác của mô hình.
  - Chỉ giới hạn trong khả năng xử lý các biến độc lập không tuyến tính.

c. Decision Tree [6]

- Ý tưởng: Decision Tree được xây dựng bằng cách tách các mẫu dữ liệu thành các nhóm con dựa trên các thuộc tính của chúng. Mỗi nút trên cây quyết định đại diện cho một thuộc tính của dữ liệu, và các nhánh đi từ nút này đại diện cho các giá trị ngưỡng có thể có của thuộc tính đó. Tại mỗi nút lá trên cây quyết định, các mẫu dữ liệu được phân loại vào một lớp nhất định.
- Ưu điểm:
  - Dễ hiểu và giải thích cho những người không có kinh nghiệm về Machine Learning. Kết quả của thuật toán được biểu diễn dưới dạng cây quyết định, giúp dễ dàng hình dung và phân tích.
  - Xử lý những dữ liệu có tính phức tạp, tức là dữ liệu có nhiều đặc trưng và phân bố không đồng đều, không cần phải chuẩn hóa dữ liệu trước khi đưa vào huấn luyện.
  - Tương đối nhanh và có thể xử lý được những tập dữ liệu lớn.
- Nhược điểm
  - Dễ bị overfitting khi số lượng các quyết định quá lớn, dẫn đến việc mô hình chỉ hoạt động tốt trên tập huấn luyện nhưng không thể tổng quát hóa được trên tập kiểm tra.
  - Dẫn đến kết quả khác nhau khi dữ liệu thay đổi nhỏ.
  - Dễ bị ảnh hưởng bởi nhiễu trong dữ liệu, đặc biệt là khi số lượng các đặc trưng quá lớn.

d. K Nearest Neighbors (KNN)[5]

- Ý tưởng: Thuật toán phân lớp KNN (K-Nearest Neighbors) dựa trên cách tính khoảng cách giữa các điểm dữ liệu và dựa trên những điểm gần nhất của dữ liệu đó để dự đoán lớp của nó. Ý tưởng của thuật toán KNN là dựa trên giả định rằng những điểm gần nhau trong không gian đặc trưng cũng có xu hướng thuộc cùng một lớp.
- Ưu điểm:
    - Dễ dàng triển khai và áp dụng trong thực tế.
    - Không yêu cầu giả định về phân phối của dữ liệu.
    - Có khả năng xử lý được các loại dữ liệu phức tạp, bao gồm cả dữ liệu không cân bằng và dữ liệu có nhiễu.
    - Độ chính xác cao khi số lượng điểm dữ liệu lớn.
- Nhược điểm:
    - Phụ thuộc vào số lượng điểm dữ liệu gần nhất (K) được chọn, nếu chọn K sai có thể dẫn đến kết quả phân loại không chính xác.
    - Cần phải tính toán khoảng cách giữa các điểm dữ liệu, đặc biệt khi số lượng điểm dữ liệu lớn thì việc tính toán này sẽ tốn nhiều thời gian tính toán.
    - Không phù hợp với dữ liệu có nhiều chiều vì độ chính xác sẽ giảm đi rất nhanh với số chiều của dữ liệu.

e. Support Vector Machine (SVM) [3]

- Ý tưởng:
    - Support Vector Machine (SVM) là thuật toán học máy có giám sát, được sử dụng chủ yếu cho việc phân loại. Trong thuật toán này, chúng ta vẽ đồ thị dữ liệu là các điểm trong n chiều với giá trị của mỗi chiều sẽ là một phần liên kết. Sau đó thực hiện tìm một siêu phẳng (hyper-plane) phân chia các lớp.

- Hyper-plane chỉ hiểu đơn giản là 1 đường thẳng, đường cong, mặt phẳng,... tùy vào số chiều chúng ta đang xét có thể phân chia các lớp ra thành các phần riêng biệt.
- Ý tưởng cơ bản của SVM là tìm một siêu phẳng (hyperplane) để phân tách các điểm dữ liệu, siêu phẳng này sẽ chia không gian thành các miền khác nhau và mỗi miền sẽ chứa một loại dữ liệu

- Ưu điểm:
  - Xử lý dữ liệu thuộc không gian có số chiều cao, thích hợp cho bài toán phân tích văn bản với lượng lớn từ.
  - Chỉ lưu trữ những điểm giúp đưa ra quyết định phân biệt các lớp giúp tiết kiệm được bộ nhớ.
  - Áp dụng tốt cho cả dữ liệu tuyến tính và phi tuyến.
- Nhược điểm
  - Không hiệu quả khi các lớp bị chồng chéo lên nhau và không có ranh giới rõ ràng.
  - Khó giải thích được xác suất thuộc về một lớp của một đối tượng vì chỉ sử dụng siêu phẳng ngăn cách các điểm dữ liệu.

f. Random Forest [2]

- Ý tưởng: Ý tưởng cơ bản của thuật toán Random Forest là tập hợp các cây quyết định ngẫu nhiên và kết hợp dữ đoán từ các cây này để đưa ra dự đoán cuối cùng, Sự phức tạp của thuật toán này là trong quá thì xây dựng mô hình, thuật toán sẽ thực hiện hai phương pháp là bootstrap sampling (BS) và random feature selection (RFS). BS sẽ tạo ra các tập dữ liệu con từ tập huấn luyện bằng cách lấy mẫu với việc thay thế. RFS được sử dụng để giới hạn số lượng đặc trưng ngẫu nhiên để xem xét ở mỗi nút chia nhánh.
- Ưu điểm:
  - Tính ổn định và khả năng giảm overfitting nhờ việc sử dụng nhiều cây quyết định ngẫu nhiên giúp thuật toán không phải quá phụ thuộc cụ thể cây quyết định nào.

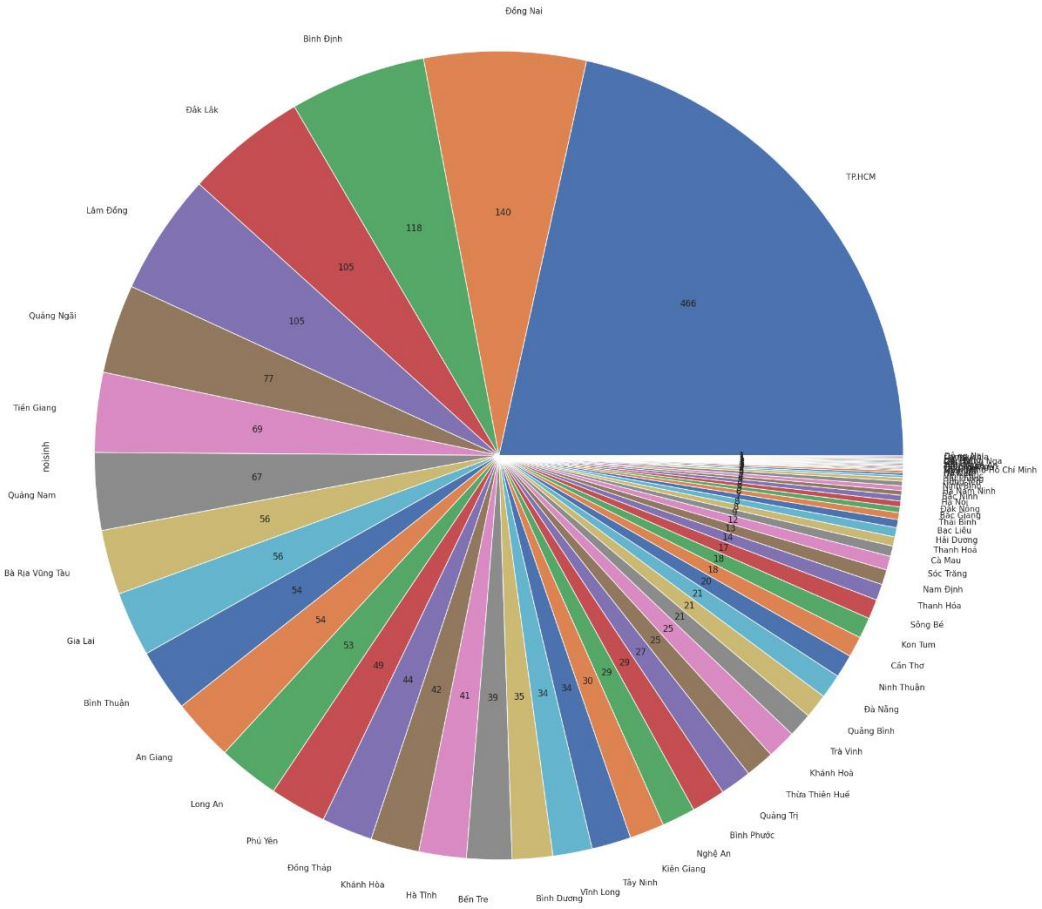
- Khả năng xử lý các tập dữ liệu lớn mà không cần quá nhiều tài nguyên tính toán, nhờ các cây quyết định có thể xây dựng song song và xử lý phân tán.
- Khả năng xử lý cả dữ liệu rời rạc và liên tục mà không yêu cầu sự chuẩn hóa dữ liệu đặc trưng
- Thể hiện độ quan trọng của đặc trưng: trong quá trình xây dựng mô hình thuật toán sẽ đánh giá độ quan trọng của đặc trưng, qua đó có thể tạo ra các tập dữ liệu con trong quá trình bootstrap sampling tốt hơn

- Nhược điểm:
  - Yêu cầu phải cài đặt nhiều tham số khác nhau như số lượng cây quyết định, số lượng đặc trưng cho mỗi lần tách cây. Các tham số này ảnh hưởng rất lớn đến kết quả.
  - Thời gian huấn luyện và dự đoán lớn so với các mô hình đơn giản
  - Đây là một phương pháp phức tạp nên là nó rất khó hiểu và khó để diễn giải các kết quả đầu ra và quá trình quyết định của mô hình
  - Do thời gian huấn luyện lớn nên là nó sẽ không hoạt động hiệu quả với dữ liệu lớn.

III. Tiến hành thực nghiệm

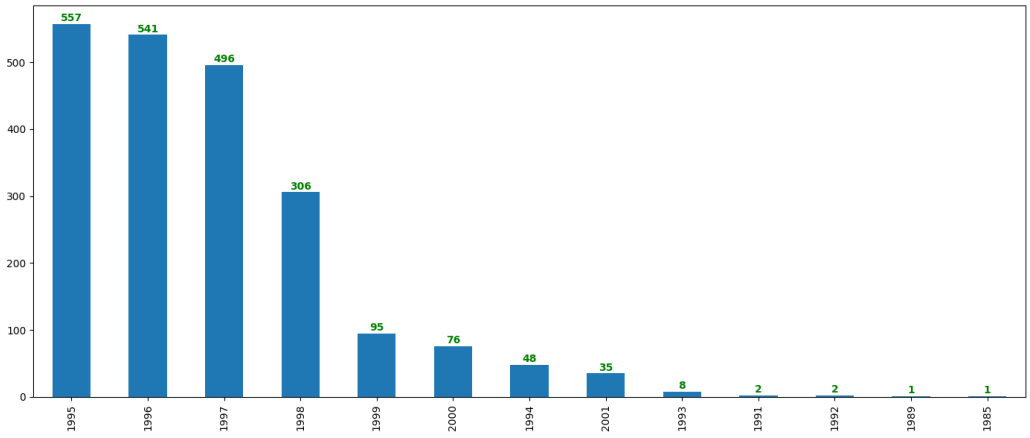
1. Dataset

- Tổng quan:
    - Dataset gồm có 2168 mẫu và có tổng cộng 2 lớp
    - Có 34 cột, trong đó 33 cột chứa đặc trưng và 1 cột chứa nhãn
    - Thông tin và phạm vi giá trị của các cột được đề cập đến ở phần II.2 bài này
- Phân tích dataset:
    - Cột ‘noisinh’



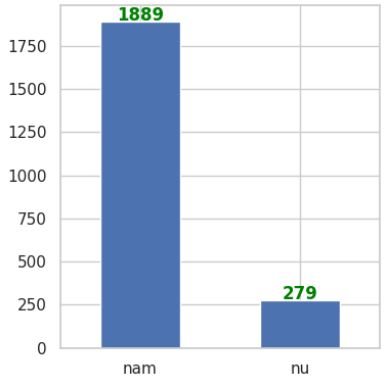
Hình 1 – Biểu đồ phân bố dữ liệu cột ‘noisinh’

- Cột ‘namsinh’

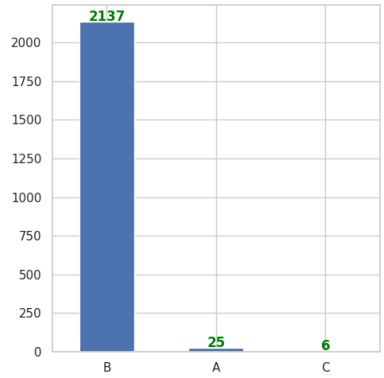


Hình 2 – Biểu đồ phân bố dữ liệu cột ‘namsinh’

- Cột ‘gioitinh’, ‘dien_tt’

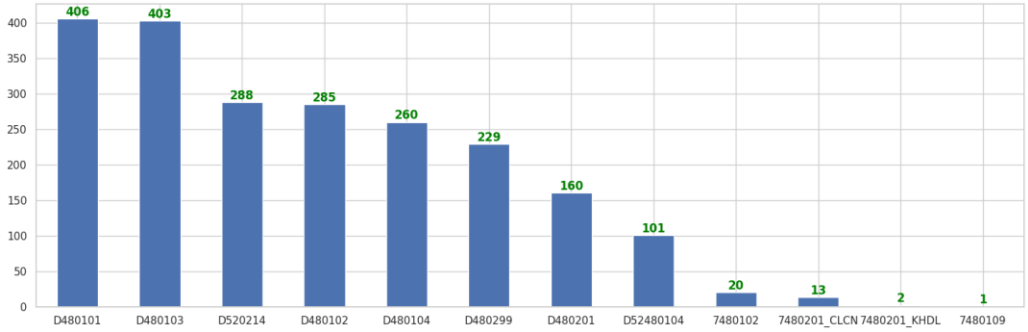


Hình 3 – Biểu đồ phân bố dữ liệu cột ‘gioitinh’



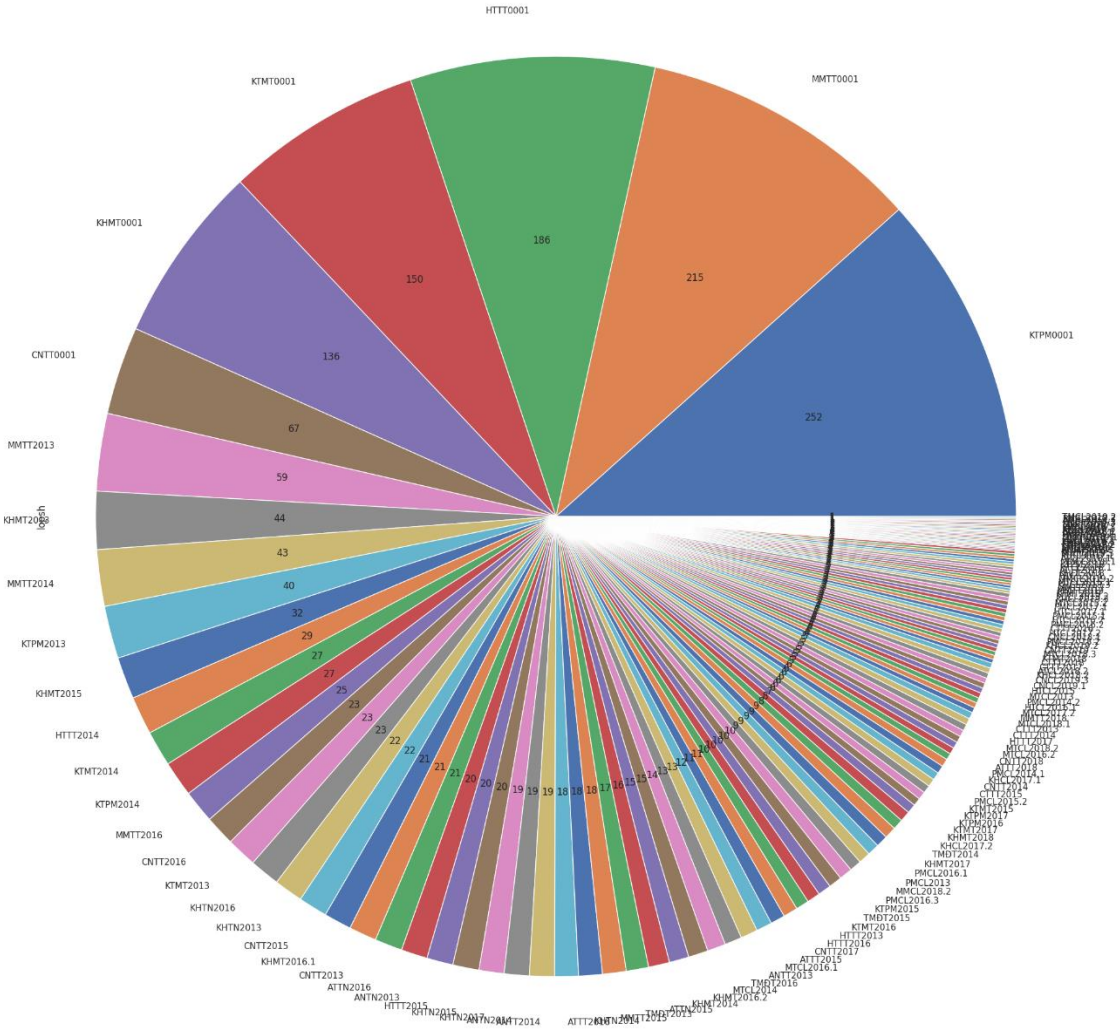
Hình 4 – Biểu đồ phân bố dữ liệu cột ‘dien_tt’

- Cột ‘chuyennganh2’



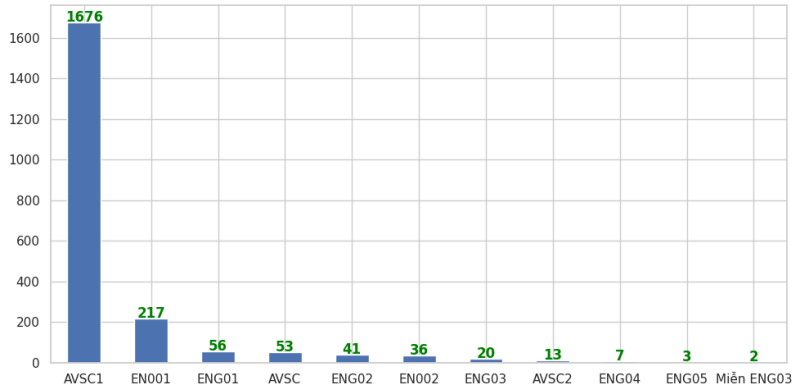
Hình 5 – Biểu đồ phân bố dữ liệu cột ‘chuyennganh2’

- Cột ‘lopsh’



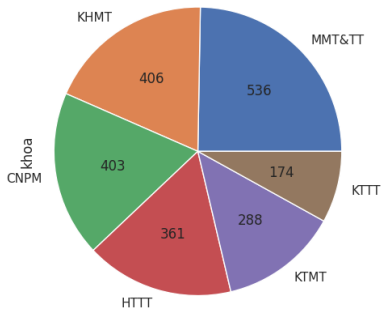
Hình 6 – Biểu đồ phân bố dữ liệu cột ‘lopsh’

- Cột ‘mamh’

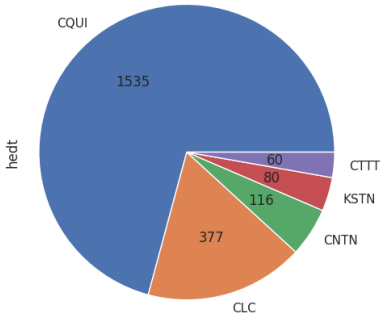


Hình 7 – Biểu đồ phân bố dữ liệu cột ‘mamh’

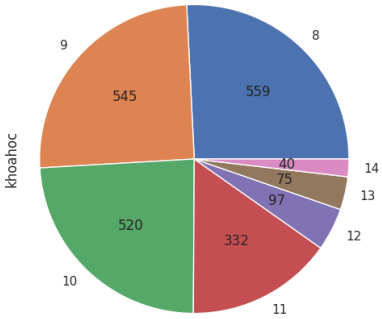
- Cột ‘khoa’, ‘hedt’, ‘khoahoc’



Hình 8 – Biểu đồ phân bố dữ liệu cột ‘khoa’

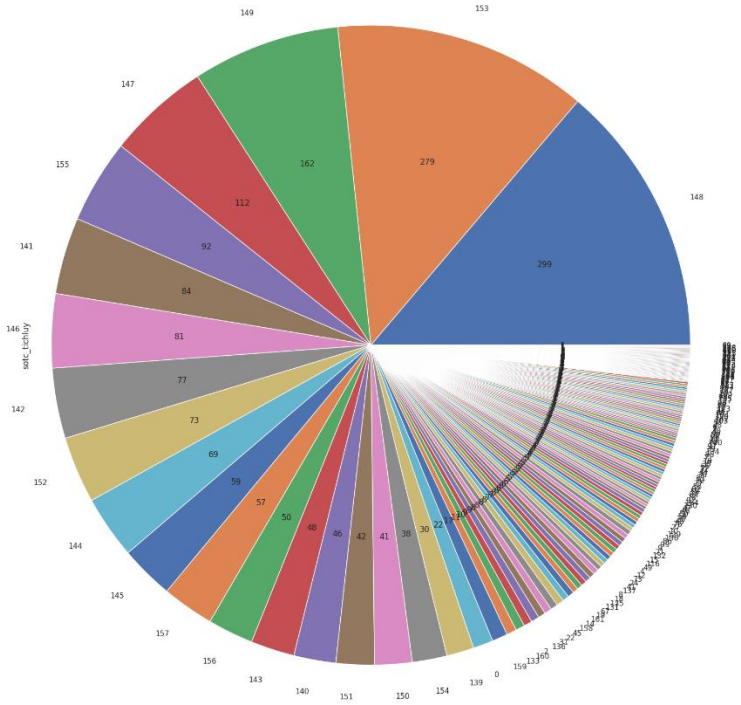


Hình 9 – Biểu đồ phân bố dữ liệu cột ‘hedt’



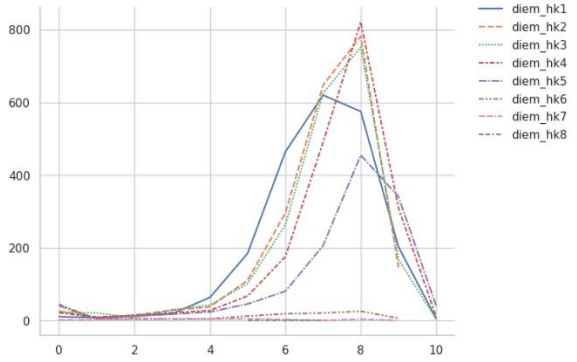
Hình 10 – Biểu đồ phân bố dữ liệu cột ‘khoahoc’

- Cột ‘Sotc_tichluy’

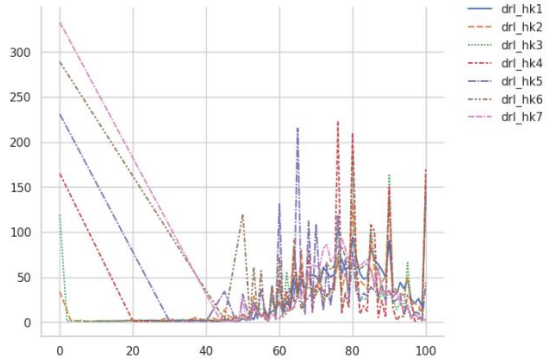


Hình 11 – Biểu đồ phân bố dữ liệu cột ‘Sotc_tichluy’

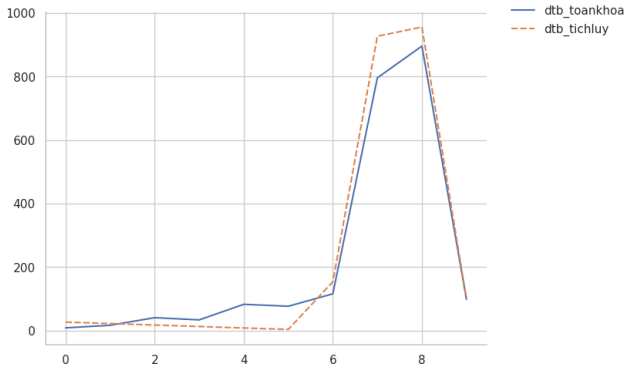
- Cột ‘dtb_toankhoa’, ‘dtb_tichluy’, ‘drl’, ‘dtb_hk’



Hình 12 – Biểu đồ phân bố dữ liệu các cột ‘dtb_hk’

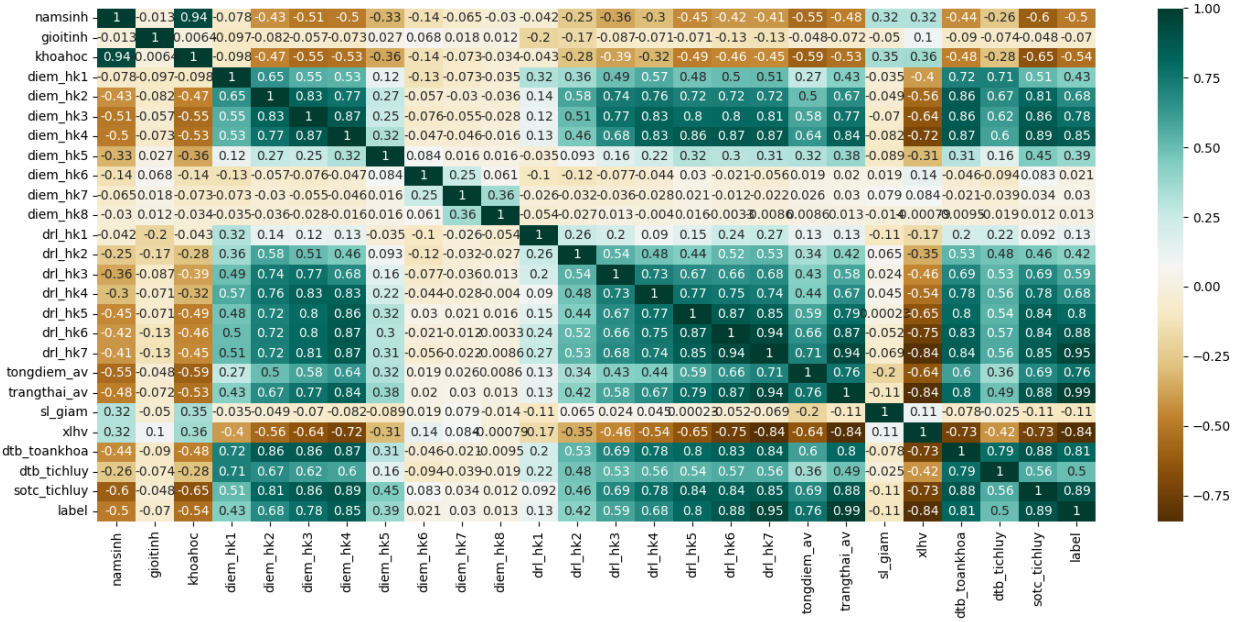


Hình 13 – Biểu đồ phân bố dữ liệu các cột ‘drl’



Hình 14 – Biểu đồ phân bố dữ liệu cột ‘dtb_toankhoa’ và ‘dtb_tichluy’

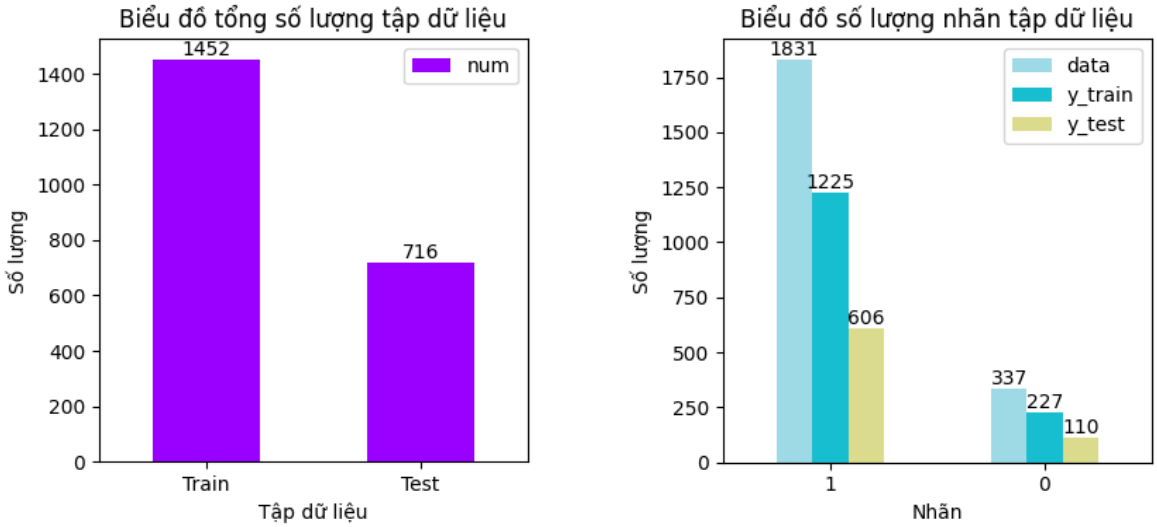
- Sau khi có bộ dữ liệu đã được tiền xử lý, ta thực hiện encoder để biến đổi dữ liệu thuộc dạng object về dạng số
- Scale dữ liệu về dạng chuẩn
- Trực quan hóa độ tương quan của dữ liệu



Hình 15 - Correlation matrix

2. Phương pháp đánh giá

Với bộ dữ liệu đã chuẩn bị, ta thực hiện chia thành tập huấn luyện (tập train) và tập kiểm tra (tập test) theo tỉ lệ 7:3 ta được như hình bên dưới, cũng như tỉ lệ các nhãn dữ liệu theo cách chia này theo hình bên cạnh



Hình 16 - Biểu đồ chi tiết dữ liệu

Sau khi chạy xong các mô hình, kiểm tra đánh giá dựa trên các độ đo Accuracy, Precision, Recall, F1-score và thống kê kết quả

3. Phương pháp thực nghiệm

Đối với tập train và tập test đã chia, nhóm tạo ra thành 10 nhóm khác nhau. Mỗi nhóm sẽ đại diện cho từng giai đoạn học của sinh viên (theo học kỳ):

- Nhóm 1_sinh viên mới vào trường: gồm có 10 cột: 'namsinh', 'gioitinh', 'noisinh', 'lopsh', 'khoa', 'hedt', 'khoahoc', 'chuyennganh2', 'dien_tt', 'mamh'.
- Nhóm 2_sinh viên có kq hk1: gồm có 10 cột của nhóm 1 và: 'diem_hk1', 'drl_hk1': tổng 12 cột.
- Nhóm 3_sinh viên có kq hk2: gồm có 12 cột của nhóm 2 và: 'diem_hk2', 'drl_hk2': tổng 14 cột
- Nhóm 4_sinh viên có kq hk3: gồm có 14 cột của nhóm 3 và: 'diem_hk3', 'drl_hk3': tổng 16 cột
- Nhóm 5_sinh viên có kq hk4: gồm có 16 cột của nhóm 4 và 'diem_hk4', 'drl_hk4': tổng 18 cột
- Nhóm 6_sinh viên có kq hk5: gồm có 18 cột của nhóm 5 và: 'diem_hk5', 'drl_hk5': tổng 20 cột
- Nhóm 7_sinh viên có kq hk6: gồm có 20 cột của nhóm 6 và: 'diem_hk6', 'drl_hk6': tổng 22 cột
- Nhóm 8_sinh viên có kq hk7: gồm có 22 cột của nhóm 7 và: 'diem_hk7', 'drl_hk7': tổng 24 cột
- Nhóm 9_sinh viên có kq hk8: gồm có 24 cột của nhóm 8 và: 'diem_hk8': tổng 25 cột
- Nhóm 10_sinh viên có kq hk8: gồm có 25 cột của nhóm 8 và 1 số thông tin khác:'loaixn', 'tongdiem_av', 'trangthai_av', 'sl_giam', 'xlhv', 'dtb_toankhoa', 'dtb_tichluy', 'sotc_tichluy': tổng 33 cột.

Cuối cùng với mỗi nhóm bộ dữ liệu đã được chia, nhóm lần lượt sử dụng để huấn luyện trên các mô hình gồm Logistic Regression, Naive Bayes, Decision Tree, Support Vector Machine, K Nearest Neighbors, Random Forest và kiểm tra đánh giá dựa trên các độ đo Accuracy, Precision, Recall, F1 score.

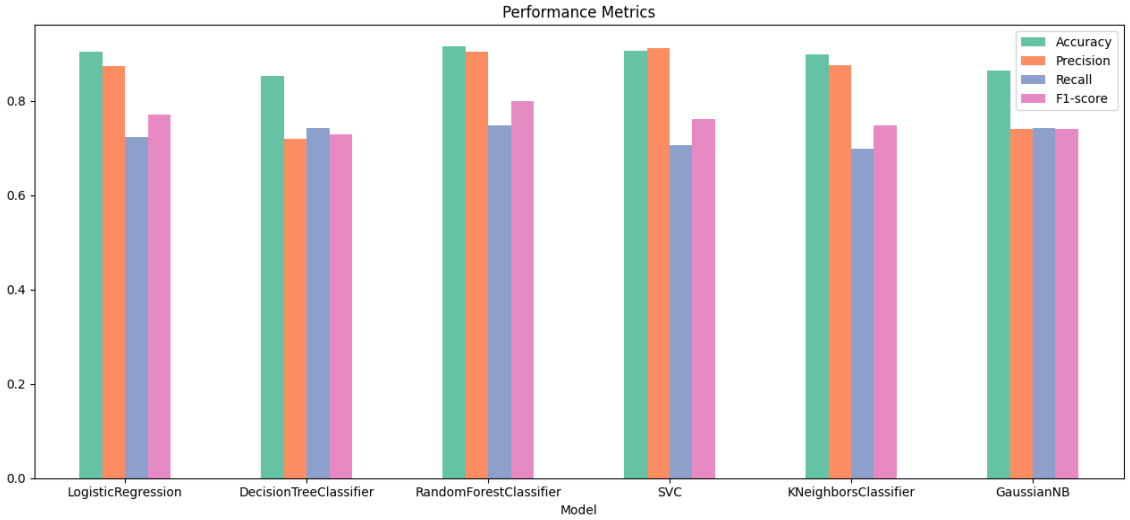
4. Kết quả thực nghiệm

4.1. Kết quả theo từng nhóm

a. Nhóm 1

Model	Accuracy	Precision	Recall	F1-score
Logistic Regression	0.90223464	0.87093496	0.7190219	0.76638389
Decision Tree	0.84916201	0.71243127	0.72487249	0.7183074
Random Forest	0.90921788	0.89091478	0.73430843	0.78416832
Support Vector Machine	0.90642458	0.92141174	0.70661566	0.76299239
K Nearest Neighbors	0.8952514	0.87119269	0.68885389	0.73757642
Naive Bayes	0.86452514	0.73960986	0.74138914	0.74049344

Bảng 3 - Kết quả nhóm 1



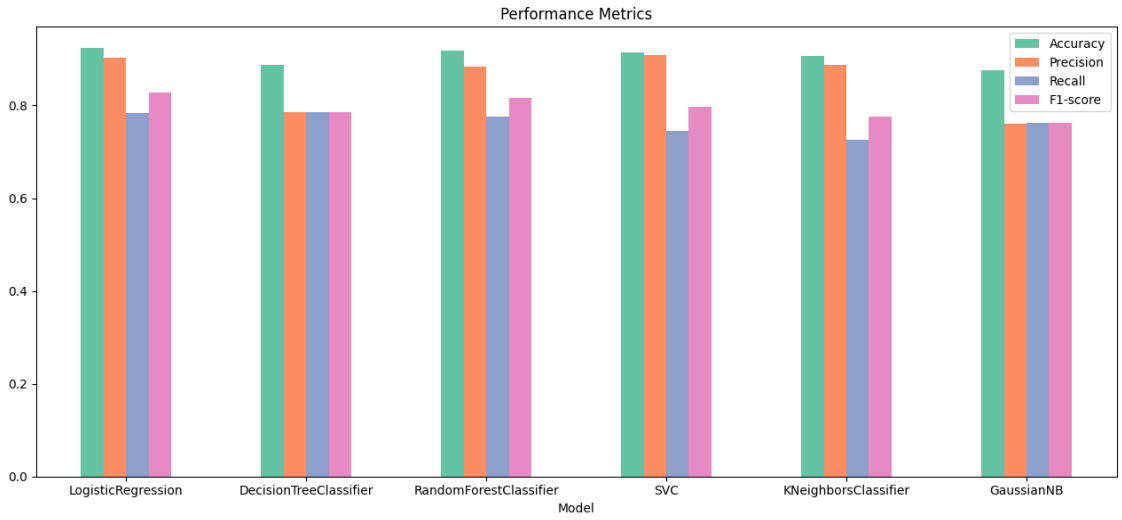
Hình 17 - Biểu đồ kết quả nhóm 1

b. Nhóm 2

Model	Accuracy	Precision	Recall	F1-score
Logistic Regression	0.92458101	0.90873016	0.78430843	0.83004835
Decision Tree	0.88826816	0.78632849	0.77773777	0.78192008
Random Forest	0.91061453	0.8664646	0.76117612	0.80044593

Support Vector Machine	0.91759777	0.92424477	0.74669967	0.8020867
K Nearest Neighbors	0.90363128	0.87299462	0.72356736	0.77088638
Naive Bayes	0.87569832	0.76093366	0.76287129	0.76189604

Bảng 4 - Kết quả nhóm 2

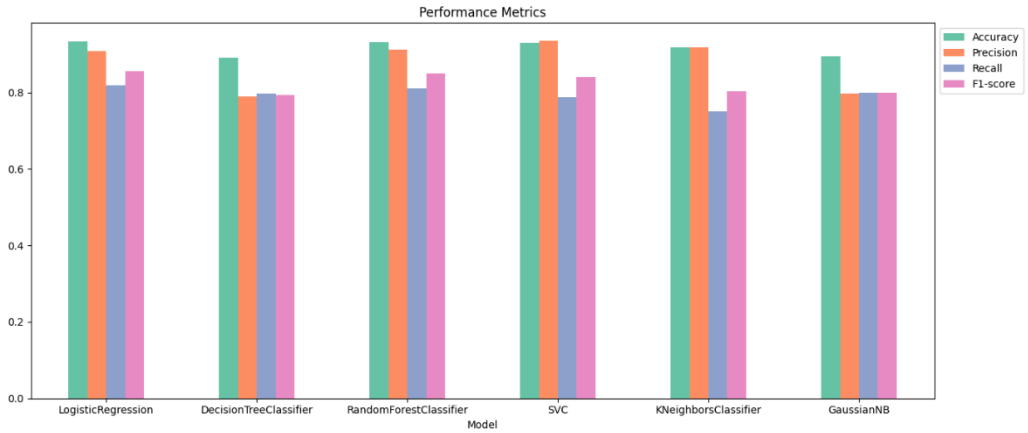


Hình 18 - Biểu đồ kết quả nhóm 2

c. Nhóm 3

Model	Accuracy	Precision	Recall	F1-score
Logistic Regression	0.93156425	0.93156425	0.81447645	0.85198561
Decision Tree	0.88687151	0.88687151	0.78435344	0.78329865
Random Forest	0.9301676	0.9301676	0.80993099	0.84829223
Support Vector Machine	0.93156425	0.93156425	0.78843384	0.84206001
K Nearest Neighbors	0.91620112	0.91620112	0.74215422	0.79769439
Naive Bayes	0.8952514	0.798250	0.80046505	0.7993506

Bảng 5 - Kết quả nhóm 3

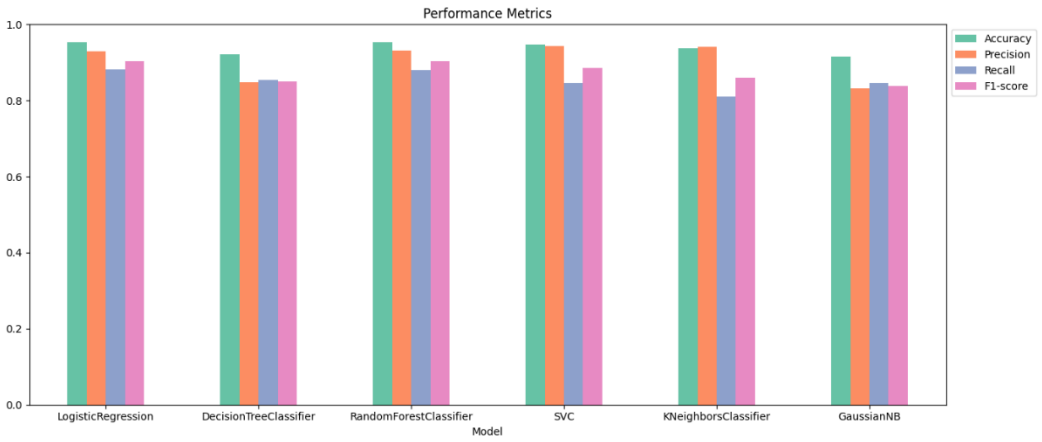


Hình 19 - Biểu đồ kết quả nhóm 3

d. Nhóm 4

Model	Accuracy	Precision	Recall	F1-score
Logistic Regression	0.952514	0.928562	0.882658	0.903609
Decision Tree	0.913408	0.832308	0.837234	0.834741
Random Forest	0.955307	0.941924	0.880588	0.907784
Support Vector Machine	0.945531	0.942364	0.841329	0.882193
K Nearest Neighbors	0.935754	0.945290	0.802070	0.853852
Naive Bayes	0.914804	0.833227	0.845500	0.839175

Bảng 6 - Kết quả nhóm 4

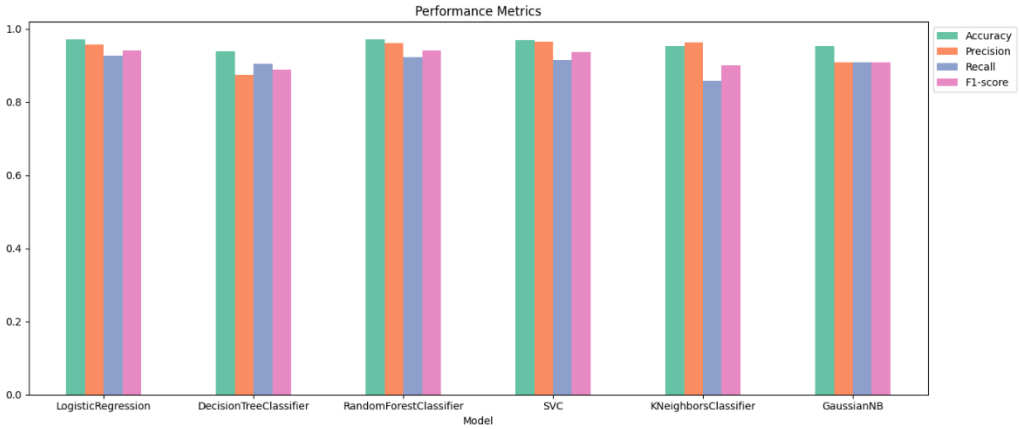


Hình 20 - Biểu đồ kết quả nhóm 4

e. Nhóm 5

Model	Accuracy	Precision	Recall	F1-score
Logistic Regression	0.970670	0.958102	0.926868	0.941637
Decision Tree	0.948324	0.899538	0.904995	0.889025
Random Forest	0.969274	0.964651	0.914881	0.937630
Support Vector Machine	0.969274	0.964651	0.914881	0.937630
K Nearest Neighbors	0.951117	0.961439	0.848350	0.893329
Naive Bayes	0.952514	0.908701	0.908701	0.908701

Bảng 7 - Kết quả nhóm 5

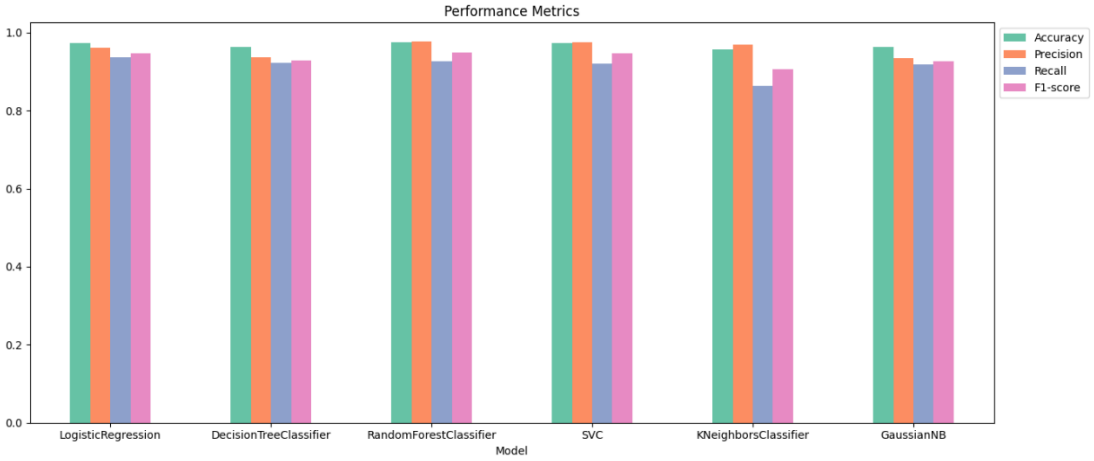


Hình 21 - Biểu đồ kết quả nhóm 5

f. Nhóm 6

Model	Accuracy	Precision	Recall	F1-score
Logistic Regression	0.973464	0.960270	0.935959	0.947606
Decision Tree	0.959497	0.920862	0.923987	0.922416
Random Forest	0.976257	0.981854	0.926448	0.951609
Support Vector Machine	0.973464	0.975786	0.921077	0.945916
K Nearest Neighbors	0.953911	0.968553	0.853720	0.899424
Naive Bayes	0.962291	0.934526	0.918197	0.926116

Bảng 8 - Kết quả nhóm 6

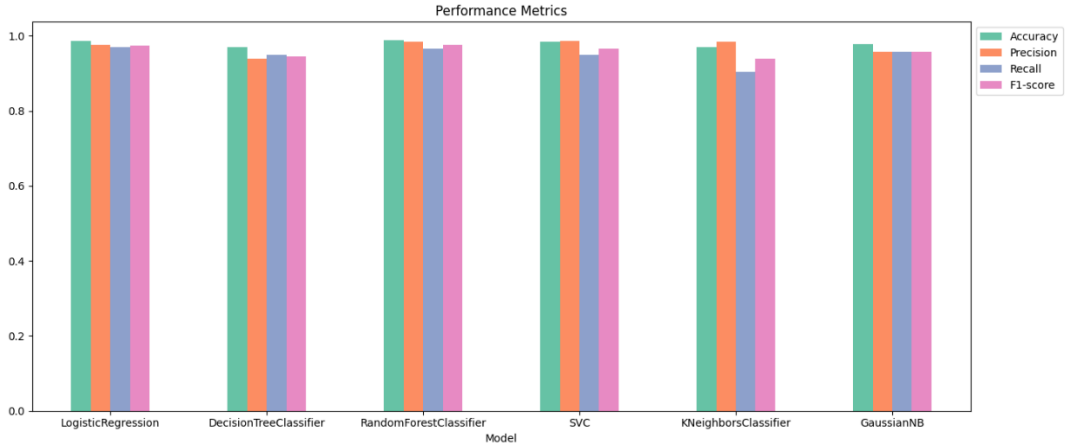


Hình 22 - Biểu đồ kết quả nhóm 6

g. Nhóm 7

Model	Accuracy	Precision	Recall	F1-score
Logistic Regression	0.998603	0.999176	0.995455	0.997305
Decision Tree	0.997207	0.994629	0.994629	0.994629
Random Forest	0.998603	0.999176	0.995455	0.997305
Support Vector Machine	0.998603	0.999176	0.995455	0.997305
K Nearest Neighbors	0.990223	0.994290	0.968182	0.980697
Naive Bayes	0.998603	0.999176	0.995455	0.997305

Bảng 9 - Kết quả nhóm 7

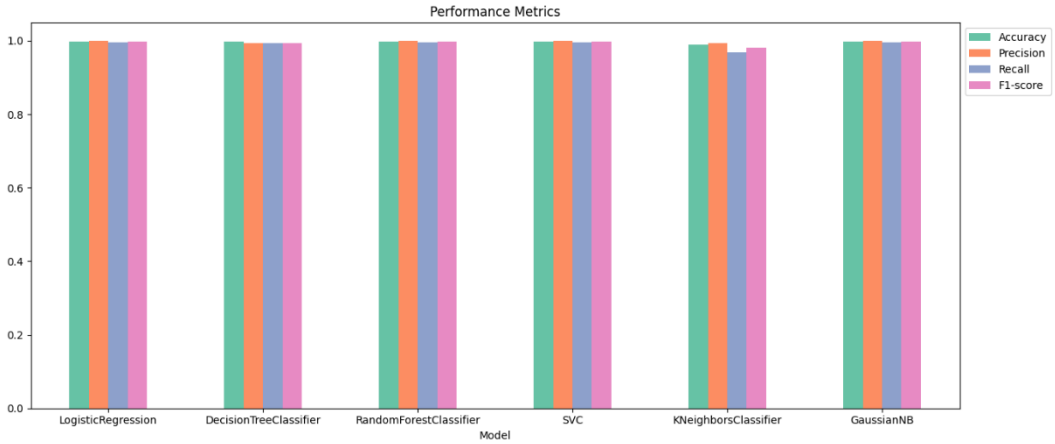


Hình 23 - Biểu đồ kết quả nhóm 7

h. Nhóm 8

Model	Accuracy	Precision	Recall	F1-score
Logistic Regression	0.998603	0.999176	0.995455	0.997305
Decision Tree	0.997207	0.994629	0.994629	0.994629
Random Forest	0.998603	0.999176	0.995455	0.997305
Support Vector Machine	0.998603	0.999176	0.995455	0.997305
K Nearest Neighbors	0.990223	0.994290	0.968182	0.980697
Naive Bayes	0.998603	0.999176	0.995455	0.997305

Bảng 10 - Kết quả nhóm 8

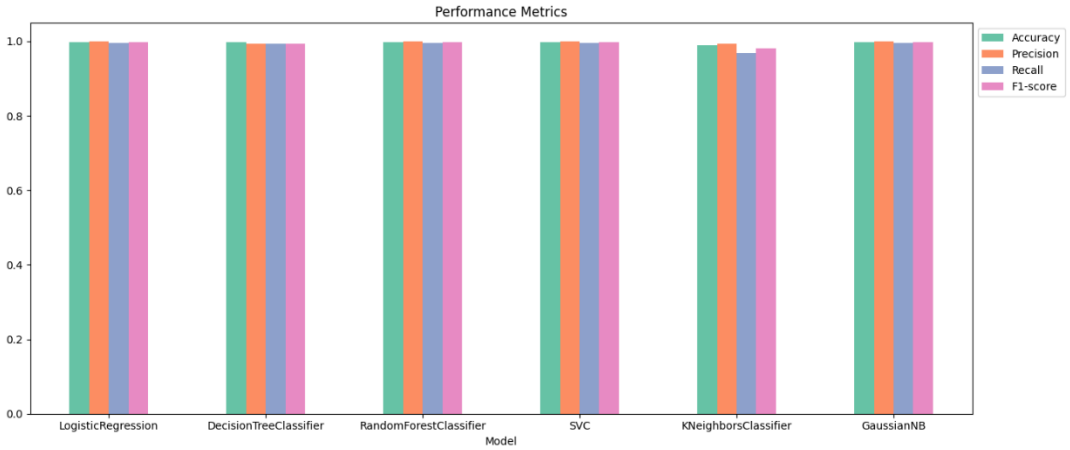


Hình 24 - Biểu đồ kết quả nhóm 8

i. Nhóm 9

Model	Accuracy	Precision	Recall	F1-score
Logistic Regression	0.998603	0.999176	0.995455	0.997305
Decision Tree	0.997207	0.994629	0.994629	0.994629
Random Forest	0.998603	0.999176	0.995455	0.997305
Support Vector Machine	0.998603	0.999176	0.995455	0.997305
K Nearest Neighbors	0.990223	0.994290	0.968182	0.980697
Naive Bayes	0.998603	0.999176	0.995455	0.997305

Bảng 11 - Kết quả nhóm 9

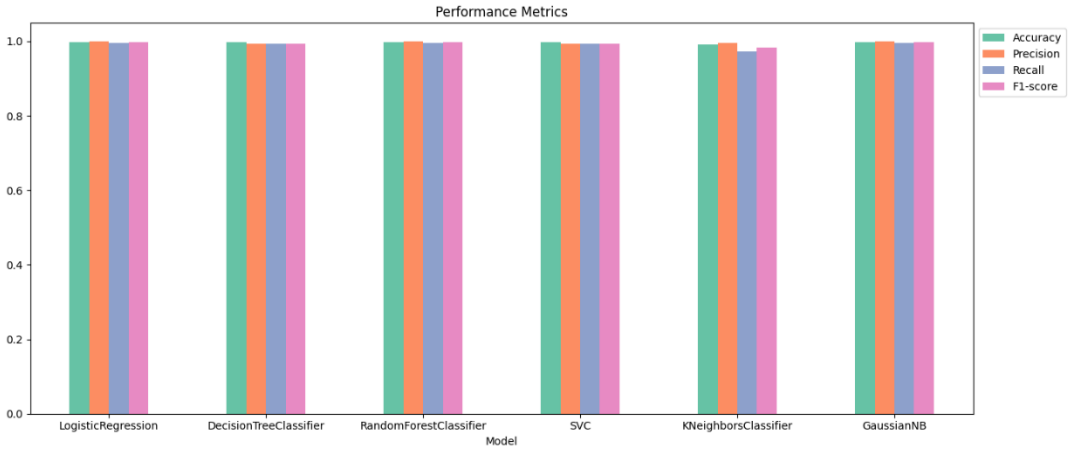


Hình 25 - Biểu đồ kết quả nhóm 9

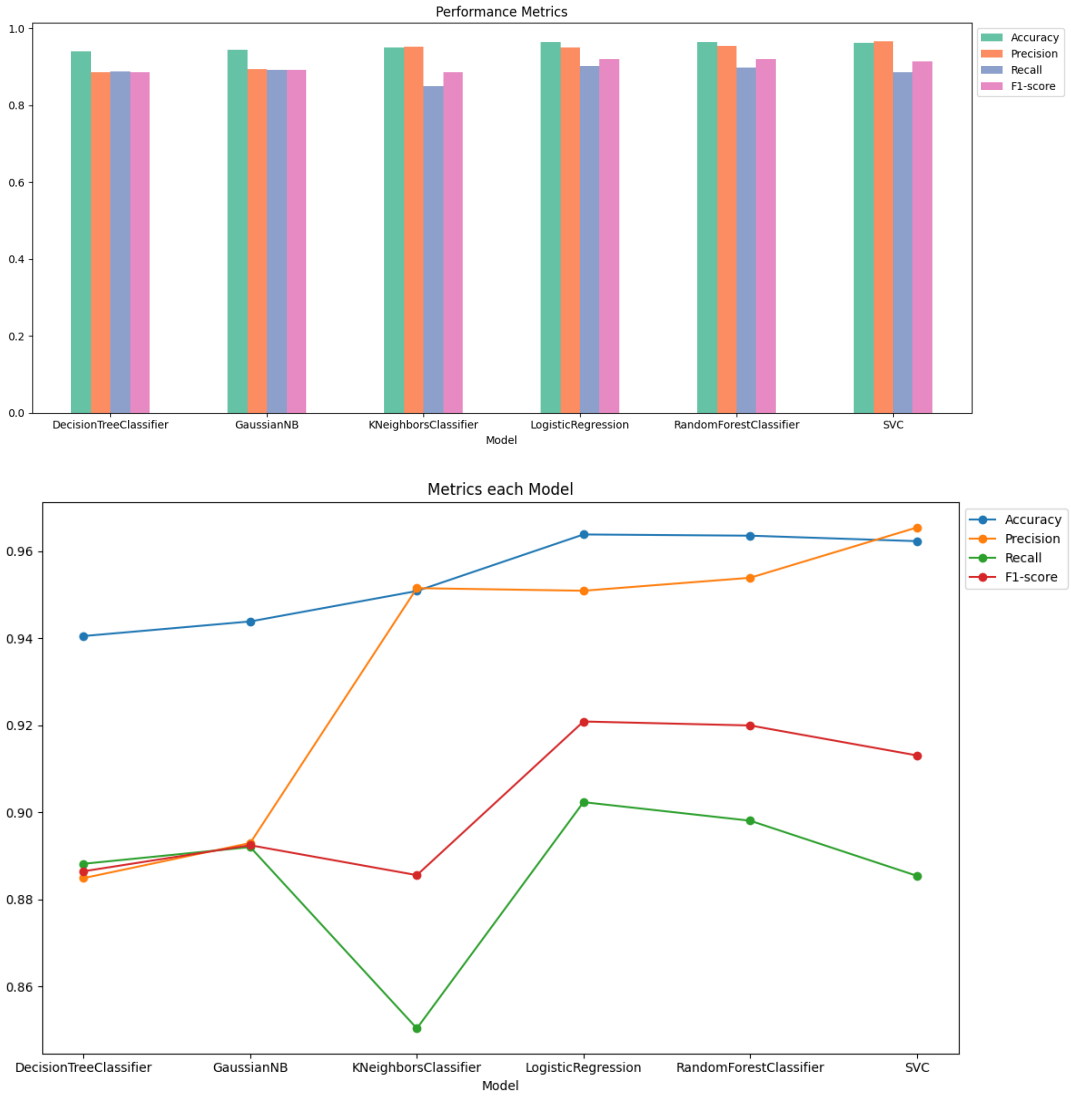
j. Nhóm 10

Model	Accuracy	Precision	Recall	F1-score
Logistic Regression	1.000000	1.000000	1.000000	1.000000
Decision Tree	0.997207	0.994629	0.994629	0.994629
Random Forest	1.000000	1.000000	1.000000	1.000000
Support Vector Machine	1.000000	1.000000	1.000000	1.000000
K Nearest Neighbors	1.000000	1.000000	1.000000	1.000000
Naive Bayes	0.998603	0.999176	0.995455	0.997305

Bảng 12 - Kết quả nhóm 10



Hình 26 - Biểu đồ kết quả nhóm 10



Hình 27 - Biểu đồ kết quả trung bình theo các mô hình

4.2. Kết quả trung bình theo từng nhóm

Group	Accuracy	Precision	Recall	F1-score
1	0.887803	0.834416	0.719177	0.751654
2	0.903399	0.853283	0.759393	0.791214
3	0.915270	0.876544	0.789969	0.820447
4	0.936220	0.903946	0.848230	0.870226
5	0.960196	0.942847	0.902698	0.919990

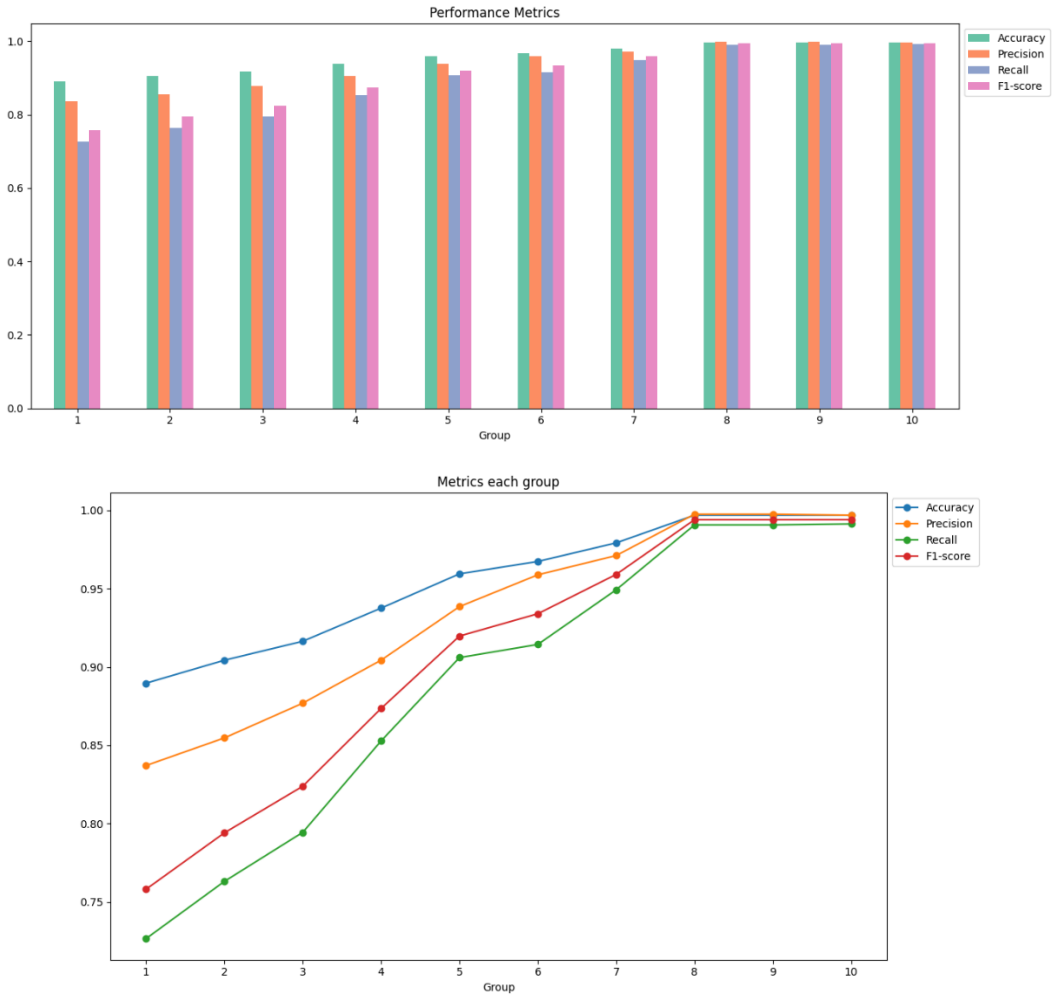
6	0.966480	0.956975	0.913231	0.932181
7	0.978818	0.970516	0.948422	0.958332
8	0.996974	0.997604	0.990772	0.994091
9	0.996974	0.997604	0.990772	0.994091
10	0.999302	0.998968	0.998347	0.998656

Bảng 13 - Kết quả trung bình 10 nhóm

IV. Nhận xét và hướng phát triển

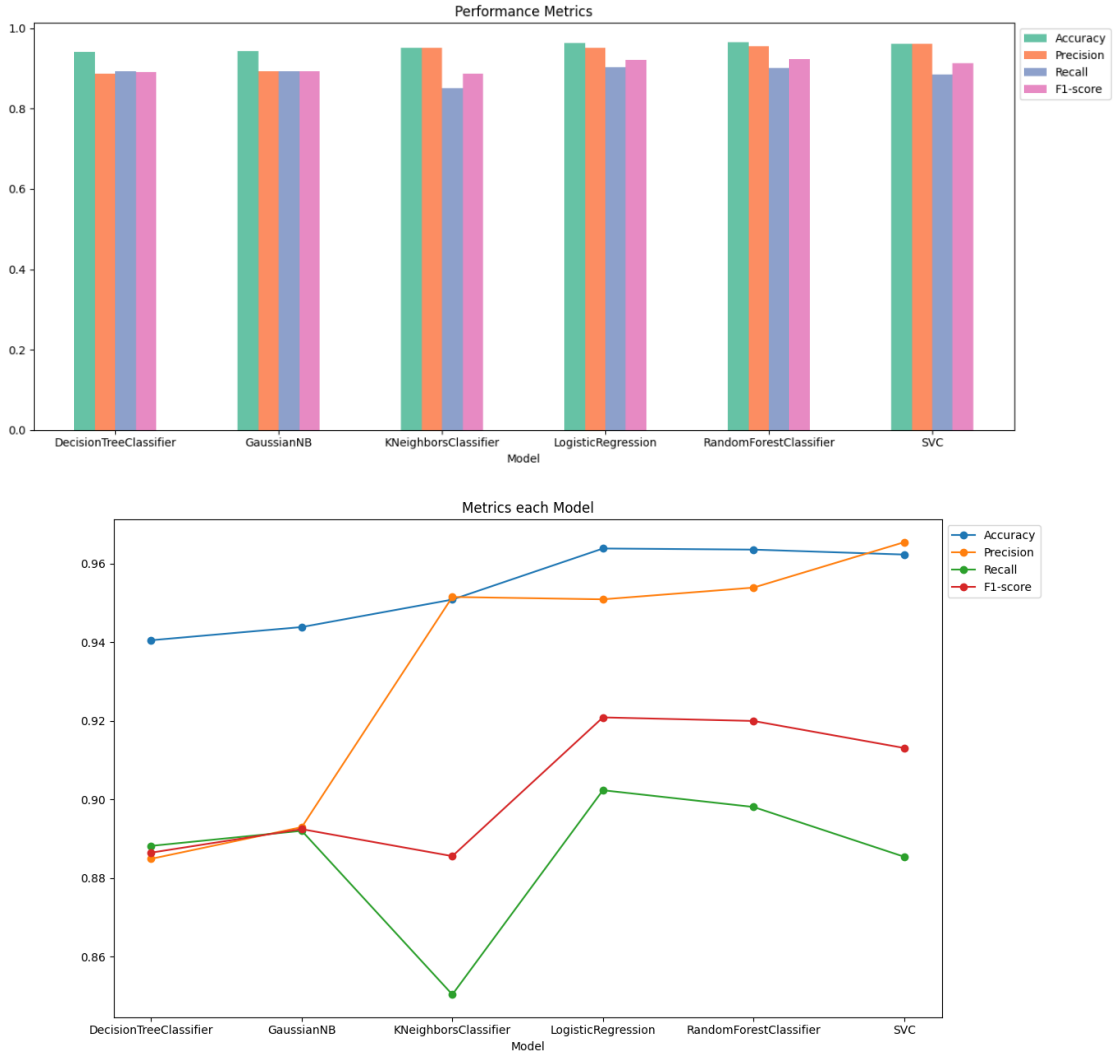
1. Nhận xét

Thông qua các đánh giá cho từng mô hình trên mỗi bộ dữ liệu khác nhau, ta có các nhận xét như sau:



Hình 28 – Biểu đồ kết quả trung bình của 10 nhóm data

- Đánh giá về bộ dữ liệu:
    - Với mỗi bộ dữ liệu, khi càng được bổ sung các thuộc tính thì tỉ lệ chính xác trong dự đoán càng được cải thiện rõ rệt. Điều này có thể được giải thích là vì hiển nhiên quá trình học (điểm số của từng học kỳ được thêm vào cho từng bộ dữ liệu) phản ánh rõ ràng khả năng tốt nghiệp của một sinh viên. Với mỗi thuộc tính về điểm học kỳ được thêm vào, mô hình sẽ có thêm các cơ sở đáng tin cậy hơn để đánh giá về mức độ tốt nghiệp của sinh viên.
    - Ta có thể thấy trong các thang đo, recall cho ra giá trị thấp hơn so với các độ đo còn lại. Để có thể đánh giá khách quan trên một bộ dữ liệu mất cân bằng nghiêm trọng như vậy, recall là độ đo cho được cái nhìn khách quan trong tình huống này khi có khả năng đo được các nhãn đã được dự đoán nhầm trên đối tượng. Khi sử dụng cách thức đánh giá macro, chúng ta có thể xem xét độ chính xác của mô hình trên cả nhãn "tốt nghiệp" và "không tốt nghiệp". Như vậy ở nhãn "không tốt nghiệp", phần lớn các sinh viên đã bị gán nhầm thành "tốt nghiệp" (mô hình khi huấn luyện trên dữ liệu mất cân bằng sẽ có xu hướng dự đoán vào nhãn chiếm số lượng lớn) dẫn đến recall không cao. Hiện tượng này cũng giải thích được tỉ lệ cao hơn hẳn của độ đo accuracy khi chỉ quan tâm đến khả năng dự đoán đúng của mô hình trên toàn bộ các điểm dữ liệu.



Hình 29 – Biểu đồ kết quả trung bình của 6 mô hình

- Mặc dù tất cả mô hình đều cho ra kết quả tốt trên các bộ dữ liệu, tuy nhiên chúng ta vẫn có thể nhận thấy các khác biệt giữa các mô hình này với nhau thông qua kết quả đánh giá, cụ thể như sau:
    - Sở dĩ Random Forest cho ra kết quả tốt hơn là bởi mô hình này thực hiện việc đánh giá và đưa ra quyết định trên các mô hình con khác nhau. Đây chính là lợi thế của Random Forest khiến cho nó được sử dụng rộng rãi trên các dữ liệu dạng bảng.
    - Đối với dữ liệu dạng bảng và các thuộc tính tương đối tuyến tính so với nhãn dữ liệu, việc mô hình Logistic Regression cho ra kết quả ấn tượng là không đáng ngạc nhiên. Với bản chất hồi quy tuyến tính, mô

hình này đã có thể dự đoán hầu như chính xác và đạt được F1 score cao hơn so với các mô hình khác trừ Random Forest.

- Bên cạnh đó mô hình Support Vector Machine (SVM hoặc SVC) cũng có kết quả thuộc nhóm cao trên F1 score chính bởi khả năng phân lớp khá hiệu quả thông qua việc tạo ra siêu mặt phẳng phân các điểm dữ liệu thành hai lớp trên bộ dữ liệu tuyến tính.
- Decision Tree và Naive Bayes tuy có kết quả đánh giá cao nhưng không rõ nét như các mô hình đã được đề cập ở trên. Điều này xảy ra là bởi hai mô hình này đều giả sử các thuộc tính độc lập với nhau (không tương quan với nhau). Thế nhưng như ma trận tương quan đã chỉ ra, các thuộc tính đều có mối liên hệ với nhau từ nhẹ đến mạnh. Chính sự tương quan này đã phá vỡ đi giả sử mà các mô hình này đã đặt ra.
- Cuối cùng, đối với mô hình K Nearest Neighbors (KNN), ta nhận thấy precision tăng cao trong khi đó recall giảm nhiều hơn so với trung bình của các mô hình khác. Với bản chất dự đoán nhãn dựa trên các các nhãn gần điểm dữ liệu cần dự đoán có số lượng nhiều hơn, trong một bộ dữ liệu mất cân bằng lệch nhiều vào nhãn "tốt nghiệp", có khả năng hầu hết các nhãn sẽ đều được dự đoán là "tốt nghiệp" khiến cho accuracy và precision cao hơn (các nhãn dự đoán "tốt nghiệp" đúng với tỉ lệ cao do tỉ lệ nhãn "tốt nghiệp" trong bộ dữ liệu cao hơn và "không tốt nghiệp" đúng với tỉ lệ cao do các trường hợp "không tốt nghiệp" khi được dự đoán bằng KNN sẽ cần các điểm dữ liệu xung quanh hầu hết đều cùng giống nhãn dẫn đến dự đoán khả năng đúng cao hơn) và recall thấp hơn (nhiều nhãn "không tốt nghiệp" bị gán thành nhãn tốt nghiệp).

2. Hướng phát triển

Chờ đợi cho thời gian phát triển của trường lâu hơn từ đó sẽ có nhiều dữ liệu hơn ngoài ra có thể thu thập nhiều dữ liệu hơn về một sinh viên, qua đó làm tăng độ

hiệu quả và chất lượng của bộ dữ liệu. Ngoài ra tạo ra một chuẩn để người nhập dữ liệu có thể thao tác có hệ thống.

Kết hợp đa dạng hơn các mô hình dự đoán cũng như các kỹ thuật khai phá dữ liệu để tìm ra phương pháp giải quyết bài toán này phù hợp nhất với chi phí phải chăng. Chọn và xác định nâng cao các đặc trưng, điều này giúp cho mô hình dễ dàng dự đoán được kết quả chính xác hơn.

Thực hiện việc đánh giá và cải thiện mô hình nhiều lần, cập nhập mô hình và các xu hướng dữ liệu mới theo thời gian. Làm giàu thêm dữ liệu qua việc tự gán nhãn cho các sinh viên đang học

V. Bảng phân công công việc

MSSV	Tên thành viên	Nội dung tìm hiểu	Đánh giá
20520436	Nguyễn Văn Thành Đạt	Các mô hình máy học	100%
20520478	Nguyễn Hoàng Gia	Phân tích dữ liệu	100%
20520547	Nguyễn Thái Huy	Các phương pháp đánh giá mô hình	100%
20520817	Lê Ngọc Mỹ Trang	Các phương pháp tiền xử lý dữ liệu	100%
20520862	Nguyễn Thế Vinh	Chia dữ liệu và chạy mô hình	100%

Bảng 14 – Phân công công việc trong nhóm

VI. Tài liệu tham khảo

[1] Wright, R. E. (1995). Logistic regression. In L. G. Grimm & P. R. Yarnold (Eds.), Reading and understanding multivariate statistics (pp. 217–244). American Psychological Association.

[2] Breiman, Leo. "Random forests." Machine learning 45 (2001): 5-32.

[3] Hearst MA, Dumais ST, Osuna E, Platt J, Scholkopf B. Support vector machines. IEEE Intelligent Systems and their applications. 1998 Jul;13(4):18-28.

[4] Rish I. An empirical study of the naive Bayes classifier. InIJCAI 2001 workshop on empirical methods in artificial intelligence 2001 Aug 4 (Vol. 3, No. 22, pp. 41-46).

[5] Peterson LE. K-nearest neighbor. Scholarpedia. 2009 Feb 21;4(2):1883.

[6] Myles AJ, Feudale RN, Liu Y, Woody NA, Brown SD. An introduction to decision tree modeling. Journal of Chemometrics: A Journal of the Chemometrics Society. 2004 Jun;18(6):275-85.

[7] Novaković JD, Veljović A, Ilić SS, Papić Ž, Tomović M. Evaluation of classification models in machine learning. Theory and Applications of Mathematics & Computer Science. 2017 Apr 1;7(1):39.

[8] Alasadi SA, Bhaya WS. Review of data preprocessing techniques in data mining. Journal of Engineering and Applied Sciences. 2017 Sep;12(16):4102-7.

[9] Ali H, Salleh MN, Saedudin R, Hussain K, Mushtaq MF. Imbalance class problems in data mining: A review. Indonesian Journal of Electrical Engineering and Computer Science. 2019 Jun 1;14(3):1560-71.

[10] Lagman AC, Calleja JQ, Fernando CG, Gonzales JG, Legaspi JB, Ortega JH, Ramos RF, Solomo MV, Santos RC. Embedding naïve Bayes algorithm data model in predicting student graduation. InProceedings of the 3rd international conference on telecommunications and communication engineering 2019 Nov 9 (pp. 51-56).

[11] Lagman AC, Alfonso LP, Goh ML, Lalata JA, Magcuyao JP, Vicente HN. Classification algorithm accuracy improvement for student graduation prediction using ensemble model. International Journal of Information and Education Technology. 2020 Oct;10(10):723-7.