**Ảnh có chứa quảng trường

Mô tả được tạo tự độngĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH**

**TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN**

**KHOA HỆ THỐNG THÔNG TIN**

**BÁO CÁO ĐỒ ÁN**

**MÔN ĐIỆN TOÁN ĐÁM MÂY**

**Đề tài: SỬ DỤNG GOOGLE COLLAB ĐỂ PHÂN TÍCH VÀ DỰ ĐOÁN KHÁCH HÀNG RỜI BỎ DỊCH VỤ NGÂN HÀNG**



GVHD: ThS. Hà Lê Hoài Trung

Nhóm sinh viên thực hiện:

1. Nguyễn Cao Khoa MSSV: 19521694

2. Hồ Trọng Khang MSSV: 19521661

3. Trần Nhật Tân MSSV: 19522177

4. Huỳnh Quốc Khánh MSSV: 19521677

Tp. Hồ Chí Minh, 12/2022

**NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN**

*……., ngày……...tháng……năm 2022…*

**Người nhận xét**

*(Ký tên và ghi rõ họ tên****)***

**LỜI CẢM ƠN**

Nhóm xin chân thành cảm ơn thầy Hà Lê Hoài Trung - giảng viên môn Điện Toán Đám Mây đã nhiệt tình chỉ bảo và hướng dẫn tận tình trong suốt thời gian thực hiện đồ án môn học. Nhờ đó, chúng em đã có được nhiều kiến ​​thức bổ ích trong việc ứng dụng cũng như kỹ năng làm đồ án.

Nhóm em tìm hiểu về một thư viện mã nguồn mở cho machine learning là Tensorflow từ đó phân tích và dự đoán khả năng khách hàng rời bỏ dịch vụ dựa trên tệp dữ liệu Churn Modeling. Dựa trên những kiến ​​thức được giáo viên cung cấp trên lớp, kết hợp với việc tự nghiên cứu các công cụ và kiến ​​thức mới, nhóm cố gắng thực hiện dự án một cách tốt nhất có thể. Trong thời gian thực hiện dự án, nhóm chúng em đã sử dụng kiến ​​thức nền tảng đã tích lũy được, kết hợp học tập và nghiên cứu kiến ​​thức mới. Từ đó, nhóm của chúng em sử dụng đầy đủ các thông tin thu thập được để đưa ra báo cáo dự án tốt nhất có thể. Tuy nhiên, trong quá trình thực hiện, nhóm chúng em không tránh khỏi những thiếu sót. Vì vậy, nhóm rất mong nhận được sự góp ý của các thầy để nhóm em hoàn thiện hơn kiến ​​thức và chuẩn bị cho các đề tài khác trong tương lai.

Sau cùng, nhóm xin kính chúc thầy thật dồi dào sức khỏe, niềm tin để tiếp tục thực hiện sứ mệnh cao đẹp là truyền đạt kiến thức cho các bạn sinh viên.

Xin chân thành cảm ơn Thầy!

**DANH MỤC CÁC BẢNG, HÌNH ẢNH:**

[**Hình 1: Tổng quan về dataset** 7](#_Toc122953174)

[**Hình 2: Thành phân chính của Tensorflow** 11](https://uithcm.sharepoint.com/sites/inTon/Shared%20Documents/General/Cloude_Final.docx#_Toc122953175)

[**Hình 3: Cấu trúc cơ bản của mạng nơ ron nhân tạo ANN** 14](#_Toc122953176)

[**Hình 4: Mô hình ANN đơn giản** 15](#_Toc122953177)

[**Hình 5: Kết nối google drive** 16](#_Toc122953178)

[**Hình 6: Import các thư viện** 16](#_Toc122953179)

[**Hình 7: Đọc dữ liệu** 16](#_Toc122953180)

[**Hình 8: Xóa các cột không liên quan** 17](#_Toc122953181)

[**Hình 9: Kiểm tra dữ liệu Null** 17](#_Toc122953182)

[**Hình 10: Kiểm tra kiểu dữ liệu các cột** 17](#_Toc122953183)

[**Hình 11: Kiểm tra outliers các thuộc tính** 17](#_Toc122953184)

[**Hình 12: Biểu đồ quan hệ giữa Age và Exit** 18](#_Toc122953185)

[**Hình 13: Biểu đồ quan hệ giữa Balance và Exit** 19](#_Toc122953186)

[**Hình 14: Biểu đồ quan hệ giữa Gender và Exit** 19](#_Toc122953187)

[**Hình 15: Biểu đồ quan hệ giữa Geography và Exit** 20](#_Toc122953188)

[**Hình 16: Dùng hot coding chuyển dữ liệu cột Gender và Age thành dạng 1;0** 20](#_Toc122953189)

[**Hình 17: Sử dụng MinMaxScaler để chuyển đổi dữ liệu** 21](#_Toc122953190)

[**Hình 18: Tách dữ liệu và dùng train\_test\_split** 21](#_Toc122953191)

[**Hình 19: Dùng TensorFlow và Keras để format** 21](#_Toc122953192)

**MỤC LỤC:**

[**LỜI CẢM ƠN** 3](#_Toc123028671)

[**PHẦN 1: TỔNG QUAN ĐỀ TÀI** 7](#_Toc123028672)

[**1.** **Lý do chọn đề tài** 7](#_Toc123028673)

[**2.** **Mục tiêu** 7](#_Toc123028674)

[**3.** **Hướng tiếp cận** 7](#_Toc123028675)

[**PHẦN 2: TỔNG QUAN VỀ TẬP DỮ LIỆU** 8](#_Toc123028676)

[**1.** **Giới thiệu tập dữ liệu** 8](#_Toc123028677)

[**2.** **Mô tả tập dữ liệu** 9](#_Toc123028678)

[**PHẦN 3: GIỚI THIỆU VỀ TENSORFLOW** 10](#_Toc123028679)

[**1.** **Khái niệm** 10](#_Toc123028680)

[**2.** **TensorFlow là gì?** 10](#_Toc123028681)

[**3.** **Cách hoạt động** 11](#_Toc123028682)

[**4.** **Các thành phần chính** 12](#_Toc123028683)

[**5.** **Lợi ích từ TensorFlow** 13](#_Toc123028684)

[**6.** **Các thuật toán nổi bật được hỗ trợ bởi TensorFlow** 13](#_Toc123028685)

[**PHẦN 4: GIỚI THIỆU VỀ ARTIFICIAL NEURAL NETWORK** 14](#_Toc123028686)

[**1.** **Artificial Neural Network là gì ?** 14](#_Toc123028687)

[**2.** **Cấu trúc và thành phần của ANN** 14](#_Toc123028688)

[**3.** **Ứng dụng của ANN vào đời sống** 16](#_Toc123028689)

[**PHẦN 5: XỬ LÝ VÀ MÔ HÌNH HÒA TRỰC QUAN DỮ LIỆU** 17](#_Toc123028690)

[**1.** **Kết nối Drive** 17](#_Toc123028691)

[**2.** **Nhập các thư viện** 17](#_Toc123028692)

[**3.** **Đọc dữ liệu** 17](#_Toc123028693)

[**4.** **Xóa các cột không liên quan** 17](#_Toc123028694)

[**5.** **Kiểm tra dữ liệu Null** 18](#_Toc123028695)

[**6.** **Kiểm tra kiểu dữ liệu các thuộc tính** 18](#_Toc123028696)

[**7.** **Kiểm tra outliers các thuộc tính** 18](#_Toc123028697)

[**8.** **Trực quan hóa dữ liệu** 18](#_Toc123028698)

[**8.1.** **Age và Exit** 19](#_Toc123028699)

[**8.2.** **Balance và Exit** 20](#_Toc123028700)

[**8.3.** **EstimatedSalary và Exit** 20](#_Toc123028701)

[**8.4.** **Geography và Exit** 21](#_Toc123028702)

[**9.** **Sử dụng one hot coding chuyển đổi dữ liệu cho 2 cột Gender và Age** 22](#_Toc123028703)

[**10.** **Sử dụng MinMaxScaler để chuyển đổi dữ liệu cho 4 cột CredictScore, EstimateSalary, Balance, Age.** 22](#_Toc123028704)

[**11.** **Tách dữ liệu vào X,Y và sử dụng train\_test\_plit** 22](#_Toc123028705)

[**12.** **Sử dụng thư viện TensorFlow và Keras để format cho mô hình** 22](#_Toc123028706)

[**PHẦN 6: PHẦN MỀM DỰ ĐOÁN** 25](#_Toc123028707)

[**1.** **Tổng quan về phần mềm** 25](#_Toc123028708)

[**1.1** **Thuật toán sử dụng** 25](#_Toc123028709)

[**1.2** **Các thuộc tính sử dụng** 26](#_Toc123028710)

[**2.** **Giao diện và testing** 26](#_Toc123028711)

[**2.1** **Giao diện** 26](#_Toc123028712)

[**2.2** **Testing** 27](#_Toc123028713)

[**3.** **Mã nguồn phần mềm** 29](#_Toc123028714)

[**a.** **Code phần giao diện:** 29](#_Toc123028715)

[**b.** **Code phần xử lý:** 34](#_Toc123028716)

[**PHẦN 7: KẾT LUẬN** 36](#_Toc123028717)

[**1.** **Kết quả đạt được** 36](#_Toc123028718)

[**2.** **So sánh, đánh giá** 36](#_Toc123028719)

[**3.** **Ưu điểm** 37](#_Toc123028720)

[**4.** **Khuyết điểm** 37](#_Toc123028721)

[**5.** **Hướng phát triển** 37](#_Toc123028722)

[**Tài liệu tham khảo** 39](#_Toc123028723)

[**Hết.** 39](#_Toc123028724)

# **PHẦN 1: TỔNG QUAN ĐỀ TÀI**

1. **Lý do chọn đề tài**

Hiện nay các công ty, doanh nghiệp bỏ ra một số tiền lớn để chi trả cho quảng cáo, marketing, khuyến mãi để có thể thu hút được nhiều khách hàng mới tham gia và cũng như là sử dụng dịch vụ hay sản phẩm của công ty. Ví dụ như các ví điện tử (momo, zalopay,..) thì họ luôn có các khuyến mãi kiểu như là : những ai chưa dùng dịch vụ thì mở ví và liên kết ngân hàng sẽ được tặng 200.000đ - 500.000đ. Và số tiền bỏ ra sẽ luôn lớn hơn gấp 4 đến 5 lần so với việc chúng ta bỏ ra để giữ chân khách hàng cũ. Vì vậy việc giữ chân khách hàng hiện tại là cách tối ưu chi phí và hợp lí để có thể duy trì một lượng khách hàng lớn.

Xác định được vấn đề chính, nhóm nhận thấy việc phân tích khách hàng nào sẽ rời bỏ dịch vụ là cần thiết. Từ các phân tích này các các công ty sẽ có thể dự đoán và biết được khách hàng nào sắp từ bỏ dịch vụ từ đó có những biện pháp chăm sóc, khuyến mãi, ưu đãi nhằm giữ chân khách hàng lại với công ty.

1. **Mục tiêu**

* Hiểu và nắm bắt được dữ liệu. Từ đó có thể xử lý được dữ liệu nhằm phục vụ cho
* yêu cầu đồ án.
* Biết cách sử dụng thư viện TensorFlow và xây dựng được mô hình dự đoán với độ
* chính xác lên tới 85%.
* So sánh và đánh giá sự hiệu quả giữa các mô hình.
* Khám phá, phân tích và trực quan hóa tập dữ liệu.

1. **Hướng tiếp cận**

Để hoàn thiện dự án này em sẽ thực hiện theo các bước như sau:

* Đầu tiên, xác định rõ mục tiêu trước và sau của dự án.
* Tiếp theo, tiến hành tìm nguồn và thu thập dữ liệu.
* Sau khi đã thu thập được dữ liệu, thực hiện chọn và xử lý dữ liệu chuyển dữ liệu thô thành dữ liệu có giá trị đối với bài toán. Cụ thể em sẽ tiến hành tìm kiếm dữ liệu bị lỗi, dữ liệu bị thiếu hoặc ngoại lệ sau đó xử lí chúng.
* Dữ liệu sau khi được xử lí và làm sạch, tiến hành phân tích và khám phá dữ liệu bằng cách trực quan hóa, biểu diễn mối liên hệ giữa mức lương với các thuộc tính khác.
* Sau đó, tiến hành xây dựng thuật toán và chỉ số đánh giá mô hình dự đoán mức lương trong lĩnh vực Công nghệ thông tin.
* Cuối cùng, so sánh kết quả, tính chỉnh xác giữa các mô hình và tổng kết dự án.

# **PHẦN 2: TỔNG QUAN VỀ TẬP DỮ LIỆU**

1. **Giới thiệu tập dữ liệu**

Để có thể dự đoán được khách hàng nào có nguy cơ yêu cầu xoá tài khoản của họ tại ngân hàng, nhóm sẽ lấy dữ liệu từ trên trang chủ Kaggle. Đây là tập dữ liệu về Dự đoán sự rời đi của khách hàng.

Tên tập dữ liệu: Customer Churn Prediction ANN, kích thước: 684.86 kB.

Tên file csv sử dụng: Churn Modeling.csv, kích thước: 684.86 kB.

Nguồn: [Churn Modeling Dataset](https://www.kaggle.com/code/rithinashikh/customer-churn-prediction-ann/data?select=Churn+Modeling.csv)

Chi tiết tập dữ liệuCột “Class” là dữ liệu dự đoán gian đoạn. Với 2 kết quả 0 và 1.

* 0 thể hiện không có sự gian lận qua các giao dịch.
* 1 thể hiện có sự gian lận qua các giao dịch.

Table

Description automatically generated

**Hình 1: Tổng quan về dataset**

1. **Mô tả tập dữ liệu**

Tập dữ liệu Churn Modeling.csv bao gồm 14 cột dữ liệu với 2931 dòng dữ liệu với các đặc trưng như sau:

**Nhóm thuộc tính thông tin của khách hàng:**

|  |  |  |  |
| --- | --- | --- | --- |
| Tên cột | Ý nghĩa | Kiểu dữ liệu | Miền giá trị |
| CustomerId | Id của khách hàng trong tập dữ liệu. | Numeric | 350-850 |
| Surname | Họ của khách hàng. | String |  |
| Geography | Tỉnh thành hoặc khu vực khách hàng sinh sống. | String | 3 |
| Gender | Giới tính. | String | Female/Male |
| Age | Độ tuổi. | Numeric | 18-92 |
| EstimatedSalary | Mức lương ước tính. | Numeric | 11.6-200.000 |

**Nhóm thuộc tính dữ liệu ngân hàng của khách hàng:**

|  |  |  |  |
| --- | --- | --- | --- |
| Tên cột | Ý nghĩa | Kiểu dữ liệu | Miền giá trị |
| CreditScore | Điểm tín dụng của khách hàng. | Numeric | 350-850 |
| Tenure | Số năm khách hàng sử dụng dịch vụ. | Numeric | 0-10 |
| Balance | Tỉnh thành hoặc khu vực khách hàng sinh sống. | Numeric | 0-251.000 |
| NumOfProducts | Số lượng sản phẩm tiện ích của ngân hàng mà khách hàng đang sử dụng. | Numeric | 1-4 |
| HasCrCard | Khách hàng có thẻ tín dụng không ? | Binary | 1: có  0: không |
| IsActiveMember | Khách hàng có còn hoạt động không ? | Binary | 1: có  0: không |
| Exited | Khách hàng rời bỏ ? | Binary | 1: có  0: không |

**Các thuộc tính khác:**

|  |  |  |  |
| --- | --- | --- | --- |
| Tên cột | Ý nghĩa | Kiểu dữ liệu | Miền giá trị |
| RowNumber | Số lượng dữ liệu trong tập dữ liệu | Numeric | 1-10.000 |

# **PHẦN 3: GIỚI THIỆU VỀ TENSORFLOW**

1. **Khái niệm**

Sự phát triển của trí tuệ nhân tạo dẫn đến việc tìm hiểu về Machine Learning và Deep Learning đã trở thanh xu thế hiện nay.

Việc sử dụng các thư viện có sẵn để tính toán đã giúp việc tiếp cận các bài toán trở nên đơn giản hơn. Vì thế mà Google đã sử dụng Deep Neural Network để cả thiện các dịch vụ của mình, trong đó có gmail, hình ảnh, google search engine.

TensorFlow được phát triển bởi đội Google Brain. Đây ra một thư viện phần mềm mã nguồn mở và ra mắt lần đầu vào cuối năm 2015 và mãi đến năm 2017 nó mới có phiên bản ổn định.

1. **TensorFlow là gì?**

TensorFlow được biết đến như là thư viện mã nguồn mở, được ứng dụng để tính toán Machine Learning trên quy mô lớn. Thư viện này kết hợp nhiều mô hình, thuật toán Machine Learning và Deep Learning.

Từ các hàm được dựng sẵn trong thư viện cho từng bài toán, TensorFlow có thể xây dựng nhiều Neural Network khác nhau. Đồng thời, các hàm này còn giúp người dùng tính toán song song trên các máy tính hay thậm chí là trên nhiều CPU, GPU trên cùng một máy. Ngoài ra, bạn còn có thể tạo ra các đồ thị luồng dữ liệu để dựng lên các model (data flow graph) từ các hàm của Tensorflow.

Phần hiệu năng của Tensorflow cực kỳ tốt do được viết từ ngôn ngữ lập trình C++ và thao tác Interface bằng Python nhanh nhạy. Đối tượng sử dụng của Tensorflow cũng cực kỳ đa dạng, từ các nhà nghiên cứu cho tới các nhà khoa học dữ liệu và lập trình viên hiện nay.

1. **Cách hoạt động**

Tensorflow cho phép bạn xây dựng cấu trúc và biểu đồ luồng dữ liệu để miêu tả cách mà data di chuyển qua một seri các node đang xử lý hoặc qua biểu đồ. Mỗi node trong đồ thị lại đại diện cho một operation toán học, hay còn được gọi là tensor hay dữ liệu đa chiều.

Kiến trúc hoạt động của Tensorflow được chia thành 3 phần chính là:

* Tiền xử lý dữ liệu.
* Dựng model.
* Train và ước tính model cụ thể.

Bằng ngôn ngữ lập trình Python, Tensorflow sẽ cung cấp toàn bộ thông tin cho các lập trình viên. Python sẽ điều phối các luồng công việc cũng như kết nối chúng lại, node và tensor trong Tensorflow là các đối tượng của Python. Chính nhờ sự hỗ trợ đắc lực từ Python, quá trình phát triển phần mềm của lập trình viên được đơn giản hoá đi rất nhiều.

1. **Diagram, schematic

   Description automatically generatedCác thành phần chính**

**Hình 2: Thành phân chính của Tensorflow**

Gồm 2 thành phần chính là Tensor và Graph

1. Tensor

Tensor: là framework cốt lõi của TensorFlow. Toàn bộ các phép tính toán đều liên quan tới Tensor. Mỗi Tensor là một vector hoặc ma trận của “n” chiều không gian đại diện cho toàn bộ dữ liệu.

Các giá trị trong cùng một tensor đều chứa một loại dữ liệu giống hệt nhau với một shape (chiều của ma trận hay mảng) dữ liệu đã biết.

Mỗi tensor có thể được bắt nguồn từ dữ liệu Input hay một kết quả tính toán. Phần edge của các node chính là tensor, một cách để nhập operation với dữ liệu này.

1. Graph

Bên trong Tensorflow toàn bộ hoạt động được tiến hành trong một biểu đồ (graph). Mỗi biểu đồ là một tập hợp tính toán liên tiếp, mỗi operation được gọi là một op node và được kết nối lại với nhau.

Graph là một thành phần quan trọng của Tensorflow, tập hợp và mô tả toàn bộ chuỗi tính toán được thực hiện trong quá trình training. Nó mang đến nhiều lợi thế như:

* Chạy trên nhiều GPU hoặc CPU ngay cả với hệ điều hành trên điện thoại.
* Bảo toàn các tính toán quan trọng ngay sau đó nhờ tính năng di động của biểu đồ. Đồng thời, biểu đồ này cũng có thể được lưu trữ và sử dụng lại trong tương lai.
* Tất cả tính toán trong biểu đồ đều được thực hiện bằng cách kết nối các tensor lại với nhau, bao gồm 1 tensor, 1 node và 1 edge. Node mang operation toán học và sản xuất các output ở đầu cuối, còn các edge giải thích mối quan hệ input/output giữa các node.

1. **Lợi ích từ TensorFlow**

Lợi ích dễ thấy nhưng quan trọng nhất mà TensorFlow cung cấp cho việc lập trình machine learning chính là abstraction. Thay vì phải đối phó với những tình huống rườm rà từ việc thực hiện triển khai các thuật toán, hay tìm ra cách hợp lý để chuyển output của 1 chức năng sang input của 1 chức năng khác, giờ đây bạn có thể tập trung vào phần logic tổng thể của 1 ứng dụng hơn. TensorFlow sẽ chăm sóc phần còn lại thay cho bạn.

Ngoài ra TensorFlow còn ung cấp các tiện ích bổ sung cho các lập trình viên cần debug cũng như giúp bạn tự suy xét các ứng dụng TensorFlow. Chế độ Eagerexecution cho phép bạn đánh giá và sửa đổi từng operation của biểu đồ 1 cách riêng biệt và minh bạch, thay vì phải dựng toàn bộ biểu đồ dưới dạng 1 đối tượng độc lập vốn khá mơ hồ hay phải đánh giá chung tổng thể. Cuối cùng, 1 tính năng khá độc đáo của TensorFlow là TensorBoard. TensorBoard cho phép bạn quan sát 1 cách trực quan những gì TensorFlow đang làm.

TensorFlow còn có nhiều cải tiến từ sự hậu thuẫn từ các ekíp thương mại hạng A tại Google. Google không những tiếp lửa cho tiến độ nhanh chóng cho sự phát triển đằng sau dự án, mà còn tạo ra nhiều phục vụ độc đáo xung quanh TensorFlow để nó dễ dàng deploy và sử dụng: như silicon TPU mình đã nói ở trên để tăng tốc hiệu suất đám mây Google, 1 online hub cho việc chia sẻ các model được tạo với framework, sự hiện diện của in-browser và gần gũi với mobile của framework, và nhiều hơn thế nữa…

1. **Các thuật toán nổi bật được hỗ trợ bởi TensorFlow**

* **Linear regression:** tf.estimator.LinearRegression.
* **Classification:** tf.estimator.LinearClassifier.
* **Deep learning classification:** tf.estimator.DNNClassifier.
* **Deep learning wipe and deep:** tf.estimator.DNNLinearCombinedClassifier.
* **Boosted tree regression:** tf.estimator.BoostedTreesRegressor.
* **Bossted tree classification:** tf.estimator.BoostedTreesClassifier

# **PHẦN 4: GIỚI THIỆU VỀ ARTIFICIAL NEURAL NETWORK**

1. **Artificial Neural Network là gì ?**

ANN là mạng neural nhân tạo và là mô hình toán học hoặc mô hình toán được xây dựng thông qua các neural sinh học. Nó bao gồm các nhóm việc, neural nhân tạo có thể nối hút với nhau và xử lý các thông tin bằng biện pháp truyền theo các kết nối rồi tính toán giá trị mới tại các nút.

Ở nhiều trường hợp, mạng Artificial Neural Network là hệ thống thích ứng có thể tự thay đổi cấu trúc của mình dựa vào thông tin bên ngoài hoặc bên trong chảy qua mạng ở quá trình học. Nhiều mạng Artificial Neural Network còn là công cụ giúp mô hình hóa dữ liệu thống kê phi tuyến. Chúng còn được sử dụng để mô hình hóa cho các mối quan hệ có tính phức tạp giữa các dữ liệu vào hoặc giữa kết quả để kiếm tìm mẫu trong dữ liệu.

1. **Cấu trúc và thành phần của ANN**

ANN lấy ý tưởng từ cách hoạt động của bộ não con người - tạo ra các kết nối phù hợp. Do đó, ANN đã sử dụng các silicon và dây điện để làm nơ ron và đuôi gai sống cho mình.Trong cơ thể con người, 1 phần não đã bao gồm 86 tỉ tế bào thần kinh và chúng được kết nối với hàng nghìn tế bào khác thông qua Axons. Bởi vì con người có rất nhiều đầu vào thông tin khác nhau từ các giác quan, nên cơ thể cũng có nhiều đuôi gai để giúp truyền thông tin này.Chúng sẽ tạo ra xung điện để di chuyển, truyền thông tin trong mạng lưới nơ ron thần kinh này. Và điều này cũng tương tự cho mạng nơ ron nhân tạo ANN - Khi cần xử lý các vấn đề khác nhau, nơ-ron sẽ gửi một thông điệp đến một nơ-ron khác.

Diagram

Description automatically generated

**Hình 3: Cấu trúc cơ bản của mạng nơ ron nhân tạo ANN**

Do đó, chúng ta có thể nói rằng ANN sẽ bao gồm nhiều nút bên trong, chúng bắt chước các tế bào thần kinh sinh học bên trong não người. Các mạng ANN sẽ kết nối các nơ-ron này bằng các liên kết và chúng có tương tác với nhau.

Các nút trong ANN được sử dụng để lấy dữ liệu đầu vào. Hơn nữa, việc thực hiện các thao tác trên dữ liệu cũng rất đơn giản. Sau khi thực hiện những thao tác với dữ liệu, các hoạt động này được chuyển cho các tế bào thần kinh khác. Đầu ra tại mỗi nút được gọi là giá trị kích hoạt hoặc giá trị nút của nó.

Mỗi liên kết trong mạng ANN đều có liên quan với trọng lượng. Ngoài ra, chúng có khả năng học hỏi. Điều đó sẽ diễn ra bằng cách thay đổi các giá trị trọng lượng. Dưới đây là một hình minh họa về một ANN đơn giản:

**Chart, diagram, bubble chart

Description automatically generated**

**Hình 4: Mô hình ANN đơn giản**

Artificial Neuron Network (ANN) gồm 3 thành phần chính: Input layer và Output layer chỉ gồm 1 layer , Hidden layer có thể có 1 hay nhiều layer tùy vào bài toán cụ thể. ANN hoạt động theo hướng mô tả lại cách hoạt động của hệ thần kinh với các neuron được kết nối với nhau.

Trong ANN, trừ Input layer thì tất cả các node thuộc các layer khác đều full-connected với các node thuộc layer trước nó. Mỗi node thuộc Hidden layer nhận vào ma trận đầu vào từ layer trước và kết hợp với trọng số để ra được kết quả.

1. **Ứng dụng của ANN vào đời sống**

* Trong không gian vũ trụ: sử dụng ANN cho máy bay lái tự động, dùng để phát hiện các lỗi còn sót trên máy bay.
* Trong quân sự: định hướng và sử dụng vũ khí, theo dõi mục tiêu.
* Trong thiết bị điện tử: dự đoán chuỗi mã, cách bố trí chip IC và phân tích lỗi chip.
* Trong y khoa: phân tích tế bào ung thư, phân tích điện tâm đồ.
* Trong vận tải: xây dựng hệ thống chuẩn đoán phan xe tải hoặc hệ thống lập trình cho xe di chuyển.
* Giọng nói: nhận dạng giọng nói và phân loại giọng nói

# **PHẦN 5: XỬ LÝ VÀ MÔ HÌNH HÒA TRỰC QUAN DỮ LIỆU**

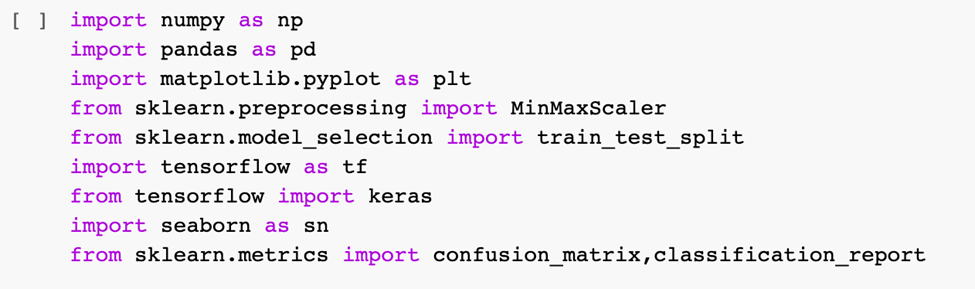
1. **Kết nối Drive**

Graphical user interface, text, application

Description automatically generated

**Hình 5: Kết nối google drive**

1. **Nhập các thư viện**



**Hình 6: Import các thư viện**

1. **Đọc dữ liệu**

A picture containing table

Description automatically generated

**Hình 7: Đọc dữ liệu**

1. **Xóa các cột không liên quan**

A picture containing graphical user interface

Description automatically generated

**Hình 8: Xóa các cột không liên quan**

1. **Kiểm tra dữ liệu Null**

Graphical user interface, application

Description automatically generated

**Hình 9: Kiểm tra dữ liệu Null**

1. **Kiểm tra kiểu dữ liệu các thuộc tính**

8.Table

Description automatically generated

**Hình 10: Kiểm tra kiểu dữ liệu các cột**

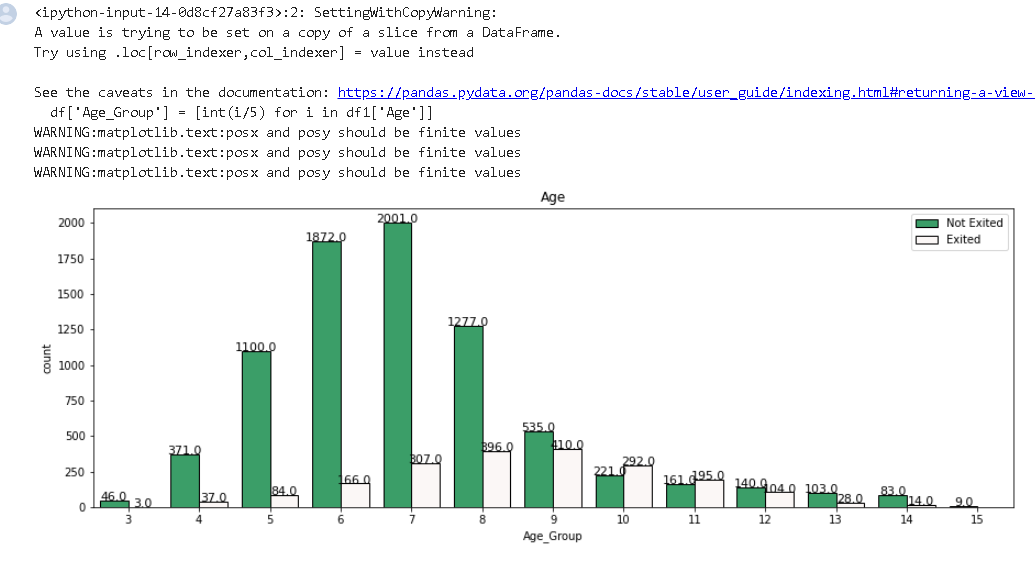
1. **Kiểm tra outliers các thuộc tính**

Graphical user interface, application

Description automatically generated

**Hình 11: Kiểm tra outliers các thuộc tính**

1. **Trực quan hóa dữ liệu**
   1. **Age và Exit**



**Hình 12: Biểu đồ quan hệ giữa Age và Exit**

**Nhận xét:** Có thể thấy số lượng khách hàng trong tầm 30 đén 45 tuổi là những khách hàng sẽ tiếp tục sử dụng dịch vụ cao nhất. Nhưng bên cạnh đó đây cũng là tệp khách có số lượng rời bỏ cao nhất => có thể do độ tuổi này họ muốn cố định về việc sử dụng tài khoản ngân hàng do đã có công việc ổn định hoặc cũng có thể do công ty yêu cầu họ sử dụng.

* 1. **Balance và Exit**

Chart

Description automatically generated

**Hình 13: Biểu đồ quan hệ giữa Balance và Exit**

**Nhận xét:** Có thể thấy số lượng khách hàng có số dư tài khoản thấp từ khoản 0 đến 250000 là những khách hàng sẽ tiếp tục sử dụng dịch vụ cao nhất => đây có thẻ là những người vừa mới tham gia dịch vụ.

* 1. **EstimatedSalary và Exit**

Chart, bar chart

Description automatically generated

**Hình 14: Biểu đồ quan hệ giữa Gender và Exit**

**Nhận xét:** Có thể thấy rằng tỉ lệ khách hàng có thu nhập cao vẫn còn sử dụng các dịch vụ cao hơn khách hàng có thu nhập thấp. Và ở chiều ngược lại thì tỉ lệ khách hàng thu nhập thấp rút khỏi các dịch vụ cao hơn khách hàng thu nhập cao.

* 1. **Geography và Exit**

Chart, waterfall chart

Description automatically generated

**Hình 15: Biểu đồ quan hệ giữa Geography và Exit**

**Nhận xét:** Có thể thấy rằng ở Pháp có số lượng khách hàng sẽ sử dụng dịch vụ lâu hơn so với 2 nước còn lại. Có thể do người dân ở đây không quá khắt khe về việc chất lượng dịch vụ.

1. **Sử dụng one hot coding chuyển đổi dữ liệu cho 2 cột Gender và Age**

**Table

Description automatically generated**

**Hình 16: Dùng hot coding chuyển dữ liệu cột Gender và Age thành dạng 1;0**

1. **Sử dụng MinMaxScaler để chuyển đổi dữ liệu cho 4 cột CredictScore, EstimateSalary, Balance, Age.**

**Text

Description automatically generated**

**Hình 17: Sử dụng MinMaxScaler để chuyển đổi dữ liệu**

1. **Tách dữ liệu vào X,Y và sử dụng train\_test\_plit**

**Graphical user interface, text, application

Description automatically generated**

**Hình 18: Tách dữ liệu và dùng train\_test\_split**

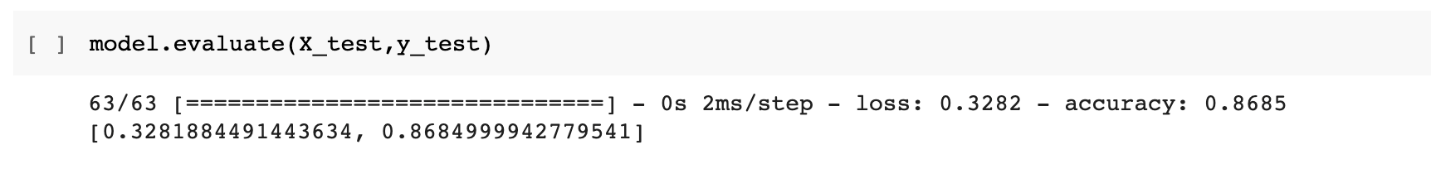
1. **Sử dụng thư viện TensorFlow và Keras để format cho mô hình**

Graphical user interface, text

Description automatically generated

**Hình 19: Dùng TensorFlow và Keras để format**

1. **Training Model**



1. **Các tiêu chí đánh giá**

Table

Description automatically generated

Recall: Cho biết có bao nhiêu điểm dữ liệu thực sự ở lớp Positive được mô hình phân lớp đúng trong mọi điểm dữ liệu thực sự ở lớp Positive.

Text

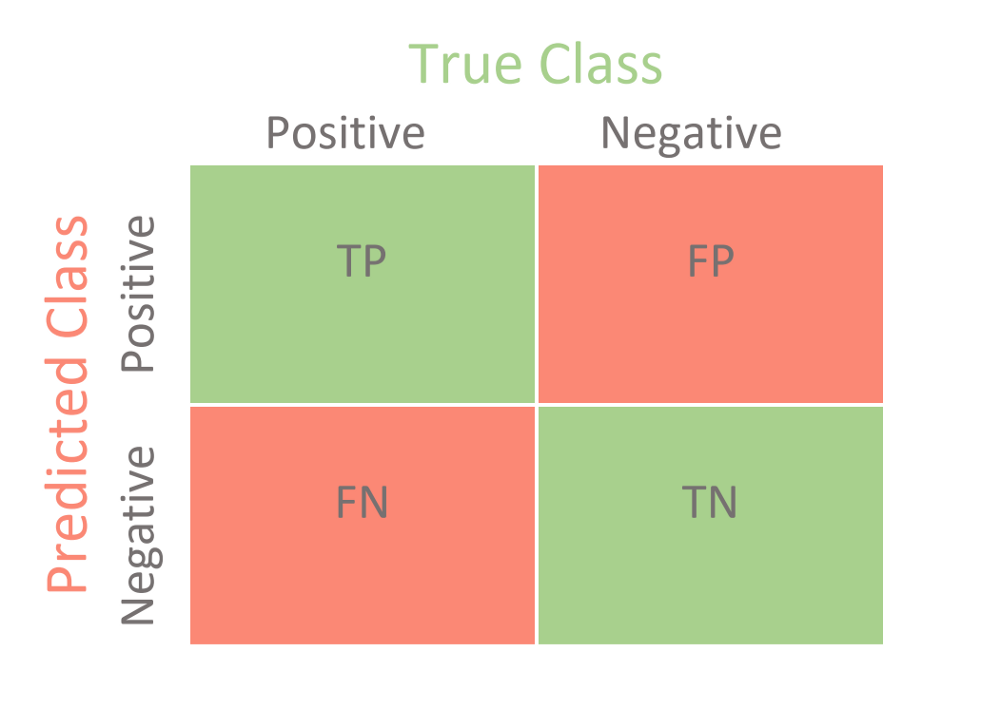
Description automatically generated

Precision: Cho biết trong số các điểm dữ liệu được mô hình phân loại hay dự đoán vào lớp Positive có bao nhiêu điểm thực sự thuộc về lớp Positive

Text

Description automatically generated with medium confidence

* Precision và Recall có giá trị trong khoảng [0,1], hai giá trị này càng gần với 1 thì mô hình càng chính xác.



Là một kỹ thuật đánh giá hiệu năng của mô hình dự đoán cho các bài toán phân lớp thể hiện số lượng điểm dữ liệu thuộc vào 1 lớp và được dự đoán thuộc về 1 lớp. Confusion Matrix cung cấp thêm thông tin về tỉ lệ phân lớp đúng giữa các lớp, hay giúp phát hiện các lớp có tỉ lệ phân lớp nhầm cao nhờ vào các khái niệm True (False), Positive (Negative).

* **True Positive (TP)**: đối tượng ở lớp Positive, mô hình phân đối tượng vào lớp Positive (dự đoán đúng).
* **True Negative (TN)**: đối tượng ở lớp Negative, mô hình phân đối tượng vào lớp Negative (dự đoán đúng).
* **False Positive (FP)**: đối tượng ở lớp Negative, mô hình phân đối tượng vào lớp Positive (dự đoán sai) – Type I Error.
* **False Negative (FN)**: đối tượng ở lớp Positive, mô hình phân đối tượng vào lớp Negative (dự đoán sai) – Type II Error.

Chart, treemap chart

Description automatically generated

# **PHẦN 6: PHẦN MỀM DỰ ĐOÁN**

* 1. **Tổng quan về phần mềm**
  2. **Thuật toán sử dụng**

**Dựa trên kết quả thu được ở phần trước, nhóm sử dụng thuật ANN cho phần mềm này. Theo kết quả, thuật toán này có độ chính xác ổn.**

* 1. **Các thuộc tính sử dụng**

Text

Description automatically generated

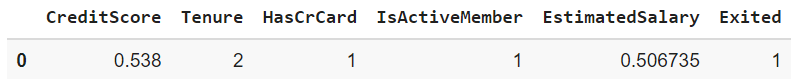
* Sau các bước trước, nhóm thu được 5 thuộc tính: CreditScore, Tenure, HasCrCard, IsActiveMember, EstimatedSalary là những thuộc tính đầu vào và thuộc tính Exited là thuộc tính dự đoán để đưa vào phần mềm.
  1. **Giao diện và testing**
  2. **Giao diện**

Graphical user interface, text, application

Description automatically generated

* 1. **Testing**

Test dataset:



Kết quả:

Graphical user interface, text, application

Description automatically generated

Graphical user interface, text, application

Description automatically generated

The software results come out in line with the original data.

* 1. **Mã nguồn phần mềm**
  2. **Code phần giao diện:**

from PyQt5 import QtCore, QtGui, QtWidgets

# from f import \*

class Ui\_MainWindow(object):

    def setupUi(self, MainWindow):

        MainWindow.setObjectName("MainWindow")

        MainWindow.resize(800, 524)

        self.centralwidget = QtWidgets.QWidget(MainWindow)

        self.centralwidget.setObjectName("centralwidget")

        self.label = QtWidgets.QLabel(self.centralwidget)

        self.label.setGeometry(QtCore.QRect(30, 10, 741, 51))

        font = QtGui.QFont()

        font.setPointSize(32)

        font.setBold(True)

        font.setWeight(75)

        self.label.setFont(font)

        self.label.setAlignment(QtCore.Qt.AlignCenter)

        self.label.setObjectName("label")

        self.label\_2 = QtWidgets.QLabel(self.centralwidget)

        self.label\_2.setGeometry(QtCore.QRect(20, 120, 150, 31))

        font = QtGui.QFont()

        font.setPointSize(11)

        font.setKerning(False)

        self.label\_2.setFont(font)

        self.label\_2.setObjectName("label\_2")

        self.label\_3 = QtWidgets.QLabel(self.centralwidget)

        self.label\_3.setGeometry(QtCore.QRect(20, 170, 150, 31))

        font = QtGui.QFont()

        font.setPointSize(11)

        font.setKerning(False)

        self.label\_3.setFont(font)

        self.label\_3.setObjectName("label\_3")

        self.label\_4 = QtWidgets.QLabel(self.centralwidget)

        self.label\_4.setGeometry(QtCore.QRect(20, 220, 150, 31))

        font = QtGui.QFont()

        font.setPointSize(11)

        font.setKerning(False)

        self.label\_4.setFont(font)

        self.label\_4.setObjectName("label\_4")

        self.label\_5 = QtWidgets.QLabel(self.centralwidget)

        self.label\_5.setGeometry(QtCore.QRect(20, 270, 150, 31))

        font = QtGui.QFont()

        font.setPointSize(11)

        font.setKerning(False)

        self.label\_5.setFont(font)

        self.label\_5.setObjectName("label\_5")

        self.label\_1 = QtWidgets.QLabel(self.centralwidget)

        self.label\_1.setGeometry(QtCore.QRect(20, 70, 150, 31))

        font = QtGui.QFont()

        font.setPointSize(11)

        font.setKerning(False)

        self.label.setFont(font)

        self.label.setAlignment(QtCore.Qt.AlignCenter)

        # self.label\_6 = QtWidgets.QLabel(self.centralwidget)

        # self.label\_6.setGeometry(QtCore.QRect(30, 200, 111, 31))

        # font = QtGui.QFont()

        # font.setPointSize(14)

        # self.label\_6.setFont(font)

        # self.label\_6.setObjectName("label\_6")

        font = QtGui.QFont()

        font.setPointSize(11)

        font.setKerning(False)

        self.label\_1.setFont(font)

        self.label\_1.setObjectName("label\_1")

        self.lineEdit\_2 = QtWidgets.QLineEdit(self.centralwidget)

        self.lineEdit\_2.setGeometry(QtCore.QRect(180, 120, 550, 31))

        self.lineEdit\_2.setObjectName("lineEdit\_2")

        self.lineEdit\_3 = QtWidgets.QLineEdit(self.centralwidget)

        self.lineEdit\_3.setGeometry(QtCore.QRect(180, 170, 550, 31))

        self.lineEdit\_3.setObjectName("lineEdit\_3")

        self.lineEdit\_4 = QtWidgets.QLineEdit(self.centralwidget)

        self.lineEdit\_4.setGeometry(QtCore.QRect(180, 220, 550, 31))

        self.lineEdit\_4.setObjectName("lineEdit\_4")

        self.lineEdit\_1 = QtWidgets.QLineEdit(self.centralwidget)

        self.lineEdit\_1.setGeometry(QtCore.QRect(180, 70, 550, 31))

        self.lineEdit\_1.setObjectName("lineEdit\_1")

        self.lineEdit\_5 = QtWidgets.QLineEdit(self.centralwidget)

        self.lineEdit\_5.setGeometry(QtCore.QRect(180, 270, 550, 31))

        self.lineEdit\_5.setObjectName("lineEdit\_5")

        self.pushButton = QtWidgets.QPushButton(self.centralwidget)

        self.pushButton.setGeometry(QtCore.QRect(50, 360, 251, 71))

        # self.lineEdit\_7 = QtWidgets.QLineEdit(self.centralwidget)

        # self.lineEdit\_7.setGeometry(QtCore.QRect(150, 230, 611, 31))

        # self.lineEdit\_7.setObjectName("lineEdit\_7")

        # self.lineEdit\_8 = QtWidgets.QLineEdit(self.centralwidget)

        # self.lineEdit\_8.setGeometry(QtCore.QRect(150, 260, 611, 31))

        # self.lineEdit\_8.setObjectName("lineEdit\_8")

        # self.lineEdit\_9 = QtWidgets.QLineEdit(self.centralwidget)

        # self.lineEdit\_9.setGeometry(QtCore.QRect(150, 290, 611, 31))

        # self.lineEdit\_9.setObjectName("lineEdit\_9")

        # self.lineEdit\_10 = QtWidgets.QLineEdit(self.centralwidget)

        # self.lineEdit\_10.setGeometry(QtCore.QRect(150, 320, 611, 31))

        # self.lineEdit\_10.setObjectName("lineEdit\_10")

        # self.lineEdit\_10 = QtWidgets.QLineEdit(self.centralwidget)

        # self.lineEdit\_10.setGeometry(QtCore.QRect(150, 350, 611, 31))

        # self.lineEdit\_10.setObjectName("lineEdit\_10")

        font = QtGui.QFont()

        font.setPointSize(32)

        self.pushButton.setFont(font)

        self.pushButton.setObjectName("pushButton")

        self.pushButton\_2 = QtWidgets.QPushButton(self.centralwidget)

        self.pushButton\_2.setGeometry(QtCore.QRect(490,360, 251, 71))

        font = QtGui.QFont()

        font.setPointSize(32)

        self.pushButton\_2.setFont(font)

        self.pushButton\_2.setObjectName("pushButton\_2")

        # self.lineEdit\_6 = QtWidgets.QLineEdit(self.centralwidget)

        # self.lineEdit\_6.setGeometry(QtCore.QRect(150, 200, 611, 31))

        # self.lineEdit\_6.setObjectName("lineEdit\_6")

        # self.label\_7 = QtWidgets.QLabel(self.centralwidget)

        # self.label\_7.setGeometry(QtCore.QRect(30, 230, 111, 31))

        # font = QtGui.QFont()

        # font.setPointSize(14)

        # font.setKerning(False)

        # self.label\_7.setFont(font)

        # self.label\_7.setObjectName("label\_7")

        MainWindow.setCentralWidget(self.centralwidget)

        self.menubar = QtWidgets.QMenuBar(MainWindow)

        self.menubar.setGeometry(QtCore.QRect(0, 0, 800, 21))

        self.menubar.setObjectName("menubar")

        MainWindow.setMenuBar(self.menubar)

        self.statusbar = QtWidgets.QStatusBar(MainWindow)

        self.statusbar.setObjectName("statusbar")

        MainWindow.setStatusBar(self.statusbar)

        self.retranslateUi(MainWindow)

        QtCore.QMetaObject.connectSlotsByName(MainWindow)

        self.pushButton.clicked.connect(self.Crun)

        self.pushButton\_2.clicked.connect(self.Clr)

        # train()

    def retranslateUi(self, MainWindow):

        \_translate = QtCore.QCoreApplication.translate

        MainWindow.setWindowTitle(\_translate("MainWindow", "Phần mềm dự đoán"))

        self.label.setText(\_translate("MainWindow", "DỰ ĐOÁN KHẢ NĂNG KHÁCH HÀNG RỜI BỎ DỊCH VỤ"))

        self.label\_1.setText(\_translate("MainWindow", "CreditScore"))

        self.label\_2.setText(\_translate("MainWindow", "Tenure"))

        self.label\_3.setText(\_translate("MainWindow", "HasCrCard"))

        self.label\_4.setText(\_translate("MainWindow", "IsActiveMember"))

        self.label\_5.setText(\_translate("MainWindow", "EstimatedSalary"))

        self.pushButton.setText(\_translate("MainWindow", "RUN"))

        self.pushButton\_2.setText(\_translate("MainWindow", "DELETE"))

    def Clr(self) -> None:

        self.lineEdit\_1.clear()

        self.lineEdit\_2.clear()

        self.lineEdit\_3.clear()

        self.lineEdit\_4.clear()

        self.lineEdit\_5.clear()

    def Crun(self) -> None:

        my\_dict =   {"CreditScore":float(self.lineEdit\_1.text()), "Tenure":float(self.lineEdit\_2.text()), "HasCrCard":float(self.lineEdit\_3.text())

        , "IsActiveMember":float(self.lineEdit\_4.text()), "EstimatedSalary":float(self.lineEdit\_5.text())}

        t=str('Khách Hàng')

        print(my\_dict)

        output = check\_input(my\_dict)

        print(output)

        msg = QtWidgets.QMessageBox()

        msg.setIcon(QtWidgets.QMessageBox.Information)

        a = ""

        if output == 0:

            a="KHÔNG CÓ KHẢ NĂNG"

            msg.setInformativeText(" {} {} rời bỏ dịch vụ".format(t,str(a)))

        elif output ==1:

            a="CÓ KHẢ NĂNG"

            msg.setInformativeText(" {} {} rời bỏ dịch vụ".format(t,str(a)))

        msg.setWindowTitle("Kết quả")

        msg.exec\_()

    # from sklearn.metrics import accuracy\_score

if \_\_name\_\_ == '\_\_main\_\_':

        train()

        app = QtWidgets.QApplication(sys.argv)

        MainWindow = QtWidgets.QMainWindow()

        ui = Ui\_MainWindow()

        ui.setupUi(MainWindow)

        MainWindow.show()

        sys.exit(app.exec\_())

* 1. **Code phần xử lý:**

from cgi import test

from re import X

from tkinter import Y

from PyQt5 import QtCore, QtGui, QtWidgets

#  phần xử lí

import pandas as pd

from sklearn.metrics import accuracy\_score

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

import os

import sys

import pickle

import numpy as np

#For training

def train() -> None:

    with open('Churn Modeling.csv') as f:

        df = pd.read\_csv(f)

    df\_filtered = df.replace('unknown',np.nan)

    df\_filtered.dropna(inplace=True)

    df\_filtered.drop(['RowNumber','CustomerId','Surname'],axis=1,inplace = True)

    df\_filtered.reset\_index(drop=True, inplace=True)

    dataset = df\_filtered.copy()

    # from sklearn.preprocessing import LabelEncoder

    # le = LabelEncoder()

    # for col in dataset.columns[[i == object for i in dataset.dtypes]]:

    #     dataset.loc[:,col] = le.fit\_transform(dataset[col])

    dataset = dataset[['CreditScore', 'Tenure', 'HasCrCard', 'IsActiveMember', 'EstimatedSalary', 'Exited']]

    from sklearn.preprocessing import MinMaxScaler

    scaler = MinMaxScaler()

    dataset[['CreditScore','EstimatedSalary']] = scaler.fit\_transform(

    dataset[['CreditScore','EstimatedSalary']])

    x = dataset.iloc[:, :-1].values

    y = dataset.iloc[:, -1].values

    # x = dataset.drop('Exited',axis = 1)

    # y = dataset['Exited']

    from sklearn.compose import ColumnTransformer

    from sklearn.preprocessing import OneHotEncoder

    ct = ColumnTransformer(transformers=[], remainder='passthrough' )

    x = np.array(ct.fit\_transform(x))

    from sklearn.preprocessing import LabelEncoder

    le = LabelEncoder()

    y = le.fit\_transform(y)

#train test split

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

    x\_train, x\_test, y\_train, y\_test = train\_test\_split(x, y, test\_size=0.3)

    # from sklearn.ensemble import RandomForestClassifier

    # classifier = RandomForestClassifier(n\_estimators =10, criterion='entropy', random\_state=0)

    # classifier.fit(x\_train, y\_train)

    # R= classifier.fit(x\_train,y\_train)

    import tensorflow as tf

    from tensorflow import keras

    model = keras.Sequential([

     keras.layers.Dense(32,input\_shape = (5,),activation='relu'),

     keras.layers.Dense(16,activation='relu'),

     keras.layers.Dense(1,activation='sigmoid')

    ])

    model.compile(optimizer='adam',loss = 'binary\_crossentropy',metrics=['accuracy'])

    model.fit(x\_train,y\_train,epochs = 50)

    R = model.fit(x\_train,y\_train)

#Save Model As Pickle File

    with open('R.pkl','wb') as m:

        pickle.dump(R,m)

# #Test accuracy of the model

# def test(X\_test):

#     with open('12\_19521694.pkl','rb') as mod:

#         p=pickle.load(mod)

#     pre=p.predict(X\_test)

#     print (accuracy\_score(X\_test,pre)) #Prints the accuracy of the model

def find\_data\_file(filename):

    if getattr(sys, "frozen", False): # The application is frozen.

        datadir = os.path.dirname(sys.executable)

    else:

# The application is not frozen.

        datadir = os.path.dirname( \_\_file\_\_)

    return os.path.join(datadir, filename)

def check\_input(data) ->int :

    df=pd.DataFrame(data=data,index=[0])

    with open(find\_data\_file('R.pkl'),'rb') as model:

        p=pickle.load(model)

    op=p.predict(df)

    return op

# **PHẦN 7: KẾT LUẬN**

1. **Kết quả đạt được**

Kết quả mà nhóm đạt được sau khi hoàn thành đồ án:

* Hiểu được vấn đề và cách giải quyết bài toán phân loại.
* Áp dụng được các phương pháp để tiền xử xử lý dữ liệu.
* So sánh và đánh giá được kết quả các mô hình.
* Tối ưu hóa được các thuật toán sử dụng
* Đưa ví dụ thực tế vào trong bài toán.

1. **So sánh, đánh giá**

|  |
| --- |
| **Thuật toán Artificial Neutral Network** |
| Giá trị Precision = 0.46 và Recall = 0.75 của mô hình còn thấp do đó độ chính xác của mô hình dự đoán Artificial Neutral Network tạm được. |
| Có 1529 khách hàng được dự đoán không có khả năng rời bỏ sử dụng dịch vụ đúng với thực tế. |
| Có 184 khách hàng được dự đoán có khả năng rời bỏ sử dụng dịch vụ đúng với thực tế. |
| Có 216 khách hàng thực tế có khả năng rời bỏ sử dụng dịch vụ nhưng mô hình dự báo đưa ra không có khả năng. (FN) (Type II Error).  Có 62 khách hàng thực tế không rời bỏ sử dụng dịch vụ nhưng mô hình dự báo kết luận có khả năng. (FP) (Type I Error). |

**Nhận xét:**

Thuật toán ANN cho kết quả dự đoán chưa tối ưu.

1. **Ưu điểm**

Những ưu điểm của đồ án:

* Áp dụng được nhiều phương pháp so sánh và đánh giá mô hình.
* Ứng dụng được ví dụ thực tế vào trong đồ án là dự đoán người dùng rời bỏ sử dụng dịch vụ .

1. **Khuyết điểm**

Vì thời gian để nghiên cứu và hiện thực đề tài còn giới hạn do đó còn tồn tại một số hạn chế, như sau:

* Độ chính xác thuật toán không cao.
* Thời gian thực thi thuận toán chậm.
* Chưa nghiên cứu tìm ra các thuật toán tối ưu hơn

1. **Hướng phát triển**

Nhóm sẽ tiếp tục phát triển đồ án để:

* Ứng dụng hệ thống vào các phần mềm chuẩn đoán trong kinh tế.
* Sử dụng nhiều tập dataset để đánh giá hiệu năng của mô hình.
* Tiếp tục nguyên cứu tìm ra các thuật toán mới tối ưu tốt hơn.
* Nghiên cứu sử dụng thêmnhiều thuật toán khác để đánh giá và so sánh từ đó tối ưu kết quả đạt được.
* Tiếp tục tối ưu các thuật toán sử dụng để đưa ra kết quả chính xác và thời gian nhanh hơn.
* Kết hợp các thuật toán lại thành một mô hình hoàn chỉnh.

# **Tài liệu tham khảo**

[1]. Link Dataset “Customer Churn Prediction ANN” - [Customer Churn Prediction ANN | Kaggle](https://www.kaggle.com/code/rithinashikh/customer-churn-prediction-ann/data?select=Churn+Modeling.csv)

[2]. “Artificial Neutral Network” - [Tổng quan về Artificial Neural Network (viblo.asia)](https://viblo.asia/p/tong-quan-ve-artificial-neural-network-1VgZvwYrlAw)

[3]. “ANN - Wikipedia” - [Mạng thần kinh nhân tạo – Wikipedia tiếng Việt](https://vi.wikipedia.org/wiki/M%E1%BA%A1ng_th%E1%BA%A7n_kinh_nh%C3%A2n_t%E1%BA%A1o)

[4]. “Các phương pháp đánh giá mô hình dự báo”- [Các phương pháp đánh giá mô hình học máy, học sâu (Machine learning & Deep learning) (rabiloo.com)](https://rabiloo.com/vi/blog/cac-phuong-phap-danh-gia-mo-hinh-machine-learning-va-deep-learning)

[5]. “Confusion matrix” - [Confusion Matrix / Ma trận nhầm lẫn / Ma trận lỗi - Viblo](https://viblo.asia/p/confusion-matrix-ma-tran-nham-lan-ma-tran-loi-p1-V3m5WQB7ZO7)

# **Hết.**