BỘ GIÁO DỤC VÀ ĐÀO TẠO

TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT TP.HCM

BÁO CÁO CUỐI KỲ MÔN TƯƠNG TÁC DỮ LIỆU TRỰC QUAN

ĐỀ TÀI

PHÂN TÍCH THẢM HỌA TÀU TITANIC

GVHD: ThS Lê Minh Tân

Lớp: Thứ 4 (tiết 1-4)

Sinh viên thực hiện: Nhóm

Nguyễn Hiếu Gia Cường MSSV: 20133027

Huỳnh Công Hậu MSSV: 20133039

Phan Hoàng Việt MSSV: 17133072

LỜI CẢM ƠN

Lời đầu tiên, nhóm em xin được gởi lời cảm ơn đặc biệt đến Thầy -Lê Minh Tân - Giảng viên phụ trách môn Tương Tác Dữ Liệu Trực Quan – trường đại học Sư Phạm Kỹ Thuật Tp Hồ Chí Minh .

Trong thời gian nhóm em làm đồ án , tụi em đã nhận được nhiều sự giúp đỡ từ thầy. Thầy đã cung cấp đầy đủ kiến thức, chỉ bảo và đóng góp những ý kiến quý báu giúp tụi em có thể hoàn thành được đồ án môn học của mình một cách tốt nhất.

Xuất phát từ mục đích học tập, tìm hiểu sâu hơn các kiến thức về dữ liệu và các thuật toán, cũng như tìm hiểu kỹ về quy trình lên ý tưởng, xây dựng dashboard. Nhóm chúng em đã thực hiện đồ án "Xây dựng dashboard để phân tích sự kiện tàu Titanic.". Trong quá trình thực hiện đồ án, dựa trên kiến thức được Thầy cung cấp qua các buổi học lý thuyết cũng như thực hành trên lớp, kết hợp với việc tự tìm hiểu những công cụ và kiến thức mới , nhóm đã cố gắng thực hiện đồ án một cách tốt nhất .Tuy nhiên, đồ án còn chưa được hoàn hiện và có nhiều sai sót.

Nhóm rất mong nhận được sự góp ý từ thầy nhằm rút ra những kinh nghiệm quý báu và hoàn thiện vốn kiến thức để nhóm có thể hoàn thành những đồ án, dự án khác trong tương lai .

Nhóm chúng em xin chân thành cảm ơn thầy!

Lời Nhận Xét Của Giảng Viên

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

CHƯƠNG 1: LÝ DO CHỌN DATASET VÀ GIỚI THIỆU TỔNG QUAN DATASET

Giới thiệu tổng quan về dataset, lý do hình thành dự án, nguồn dữ liệu thực hiện. Khảo sát, nghiên cứu và phân tích các báo cáo nghiệp vụ cần phục vụ cho dự án.

1. Lời mở đầu

1.1. Vấn đề nhận thấy

Dự án được hình thành từ tập dữ liệu của chuyến tàu Titanic bởi vì đây là một trong những sự kiện lịch sử nổi tiếng và đầy cảm xúc, khiến cho nhiều người quan tâm và muốn tìm hiểu.

Dữ liệu được thu thập từ hành khách và phi hành đoàn trên tàu, bao gồm thông tin về tuổi, giới tính, hạng ghế, điểm đến và liệu họ đã sống sót hay không trong vụ đắm tàu. Bằng cách phân tích tập dữ liệu này, ta có thể đưa ra được những thông tin và kiến thức mới về sự kiện đắm tàu Titanic và phương pháp xây dựng mô hình dự đoán khả năng sống sót của một hành khách.

1.2. Giải pháp

Dựa trên nhu cầu thống kê, phân tích và khai thác dữ liệu các hành khách trên tàu. Giải pháp là xây dựng dashboard phục vụ mục đích phân tích, khai thác, và tạo báo cáo tổng. Đưa ra các kết quả phù hợp.

1.3. Mục tiêu và ý nghĩa của dự án

Thứ nhất là nghiên cứu học tập xây dựng một dashboard phân tích dữ liệu.

Xây dựng và phát triển ứng dụng nhằm phục vụ việc phân tích, khai thác, tạo cáo nhằm nắm rỏ xu hướng công nghệ đang được ưa chuộng, sự phân cập giữa các lập trình viên, mức lương của họ theo từng quốc gia. Việc này giúp dễ dàng nhận biết xu hướng công nghệ, so sánh mức lương giữa các quốc gia.

Mục tiêu của dự án là phân tích và dự đoán khả năng sống sót của hành khách trên chuyến tàu Titanic dựa trên các thông tin về đặc điểm cá nhân, hành lý và hạng ghế của họ. Dự án này có ý nghĩa quan trọng trong việc nghiên cứu về sự kiện Titanic, giúp chúng ta hiểu rõ hơn về nguyên nhân, tác động và hậu quả của thảm họa này đối với con người và xã hội. Ngoài ra, kết quả của dự án cũng có thể áp dụng vào việc dự

đoán rủi ro và đưa ra các biện pháp phòng ngừa cho các tình huống khẩn cấp tương tự trong tương lai.

Hướng tới đối tượng sử dụng là các nhà phát triển muốn cập nhật thông tin nhằm phát triển nên các biện pháp đối phó với thảm họa tự nhiên, các doanh nghiệp có nhu cầu sản xuất các thiết bị vận tải, …

1.4. Giới thiệu tổng quan Dataset

1.4.1. Nguồn dữ liệu sử dụng

Nguồn dữ liệu được thu thập từ kaggle.com, dataset kaggle.com **Titanic – Machine Learning from Disaster.**

1.4.2. Giới thiệu nơi cấp dữ liệu

Kaggle là một trang web chuyên về kho dữ liệu và cuộc thi phân tích dữ liệu trực tuyến. Trang web này cung cấp các tập dữ liệu từ nhiều lĩnh vực khác nhau, từ kinh doanh đến khoa học và công nghệ. Ngoài ra, Kaggle còn tổ chức các cuộc thi về phân tích dữ liệu và học máy, cho phép các nhà khoa học dữ liệu, chuyên gia phân tích và lập trình viên trên toàn thế giới tham gia để giải quyết các thách thức về dữ liệu. Việc tham gia các cuộc thi này giúp các chuyên gia phát triển kỹ năng phân tích dữ liệu và học máy, cũng như giúp các công ty và tổ chức giải quyết các vấn đề phức tạp liên quan đến dữ liệu của mình.

Kaggle có:

- Hơn 4 triệu truy cập mỗi tháng (theo SimilarWeb)
- Hơn 22.000 câu hỏi (tính đến tháng 4 năm 2021)
- Hơn 35.000 câu hỏi từ nhiều lĩnh vực khác nhau như tài chính, y tế, thể thao, v.v
- Hơn 120.000 câu trả lời
- Số cuộc thi: Kaggle đã tổ chức hơn 300 cuôc thi về khoa học dữ liệu và AI từ 2010

1.4.3. Hướng dẫn tải dataset thực hiện trong đồ án và các dataset khác của nhà cung cấp

Link tải dataset:

https://www.kaggle.com/competitions/titanic/data?select=train.csv

https://www.kaggle.com/competitions/titanic/data?select=test.csv

https://www.kaggle.com/competitions/titanic/data?select=gender_submission.csv

1.5. Mô tả chi tiết dữ liệu

Tập dữ liệu Titanic là một trong những tập dữ liệu phổ biến nhất trong lĩnh vực khoa học dữ liệu. Nó chứa thông tin về hành khách trên chuyến tàu Titanic bao gồm tên, giới tính, tuổi, hạng ghế, số lượng người thân, địa chỉ, số phiếu đặt chỗ, giá tiền vé, khu vực lên tàu, thông tin về hành trang và sống hay chết trong thảm họa đắm tàu. Tập dữ liệu này được sử dụng rộng rãi để phân tích về các yếu tố có ảnh hưởng đến khả năng sống sót của hành khách trên tàu.

Ngoài ra, tập dữ liệu này còn được sử dụng để huấn luyện và đánh giá mô hình học máy, xây dựng mô hình dự đoán khả năng sống sót của hành khách trên tàu dựa trên các thông tin có sẵn và nhiều ứng dụng khác trong lĩnh vực khoa học dữ liệu.

1.6. Thông số dataset

Dữ liệu gồm có 2 bảng:

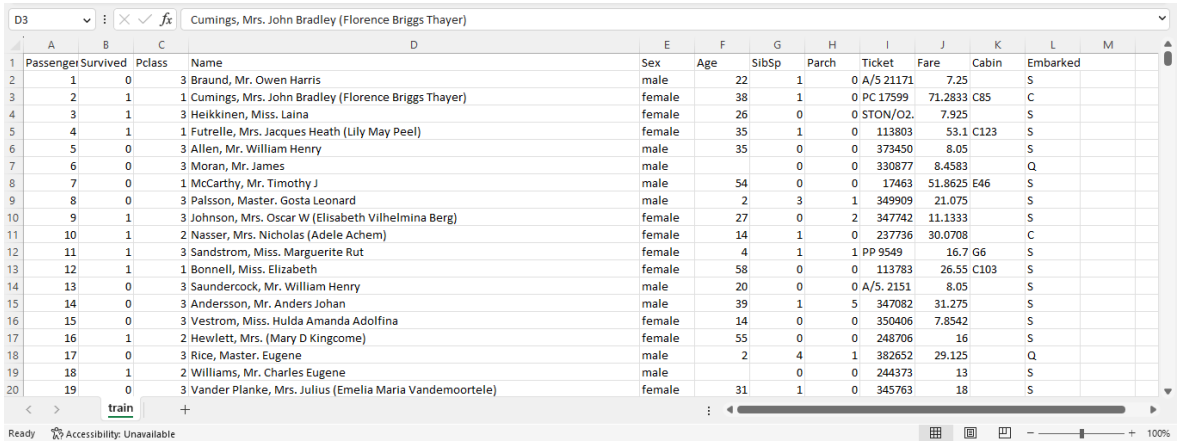
Train.csv có: 892 (dòng) * 12(cột), với mỗi dòng thể hiện một hành khách trên tàu.

Test.csv có: 419(dòng) * 11(cột), với mỗi dòng thể hiện một hành khách trên tàu.

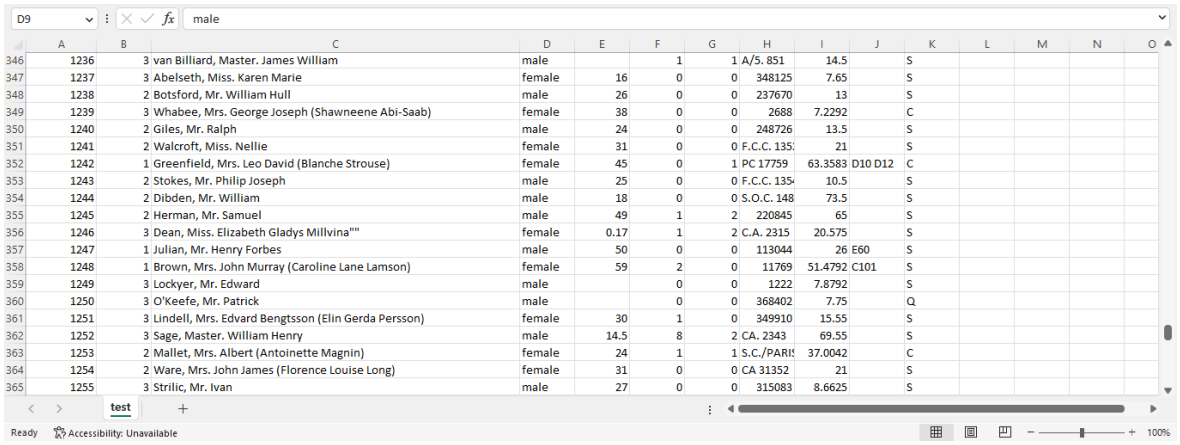
Gender_submission.csv có: 419(dòng) * 2(cột), với mỗi dòng thể hiện một id của hành khách trên tàu.

1.7. Dữ liệu sau khi trích xuất

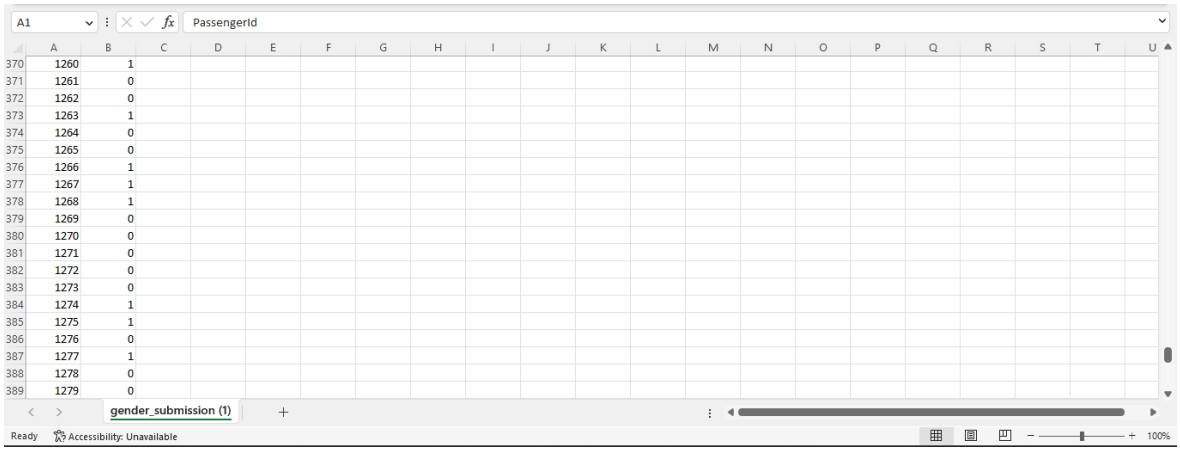
Thực hiện lấy dữ liệu từ bảng **Train.csv, Test.csv** và **Gender_submission.csv:**



PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
1	0	3	Braund, Mr. Owen Harris	male	22	1	0	A/5 21171	7.25		S
2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Thayer)	female	38	1	0	PC 17599	71.2833	C85	C
3	1	3	Heikkinen, Miss. Laina	female	26	0	0	STON/O2.	7.925		S
4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35	1	0	113803	53.1	C123	S
5	0	3	Allen, Mr. William Henry	male	35	0	0	373450	8.05		S
6	0	3	Moran, Mr. James	male		0	0	330877	8.4583		Q
7	0	1	McCarthy, Mr. Timothy J	male	54	0	0	17463	51.8625	E46	S
8	0	3	Palsson, Master. Gosta Leonard	male	2	3	1	349909	21.075		S
9	1	3	Johnson, Mrs. Oscar W (Elisabeth Vilhelmina Berg)	female	27	0	2	347742	11.1333		S
10	1	2	Nasser, Mrs. Nicholas (Adele Achem)	female	14	1	0	237736	30.0708		C
11	1	3	Sandstrom, Miss. Marguerite Rut	female	4	1	1	PP 9549	16.7	G6	S
12	1	1	Bonnell, Miss. Elizabeth	female	58	0	0	113783	26.55	C103	S
13	0	3	Saundercock, Mr. William Henry	male	20	0	0	A/5. 2151	8.05		S
14	0	3	Andersson, Mr. Anders Johan	male	39	1	5	347082	31.275		S
15	0	3	Vestrom, Miss. Hulda Amanda Adolfina	female	14	0	0	350406	7.8542		S
16	1	2	Hewlett, Mrs. (Mary D Kingcome)	female	55	0	0	248706	16		S
17	0	3	Rice, Master. Eugene	male	2	4	1	382652	29.125		Q
18	1	2	Williams, Mr. Charles Eugene	male		0	0	244373	13		S
19	0	3	Vander Planke, Mrs. Julius (Emelia Maria Vandemoortele)	female	31	1	0	345763	18		S



1236	3	van Billiard, Master. James William	male		1	1	A/5. 851	14.5		S		
1237	3	Abelseth, Miss. Karen Marie	female	16	0	0	348125	7.65		S		
1238	2	Botsford, Mr. William Hull	male	26	0	0	237670	13		S		
1239	3	Whabee, Mrs. George Joseph (Shawneene Abi-Saab)	female	38	0	0	2688	7.2292		C		
1240	2	Giles, Mr. Ralph	male	24	0	0	248726	13.5		S		
1241	2	Walcroft, Miss. Nellie	female	31	0	0	F.C.C. 135	21		S		
1242	1	Greenfield, Mrs. Leo David (Blanche Strouse)	female	45	0	1	PC 17759	63.3583	D10 D12	C		
1243	2	Stokes, Mr. Philip Joseph	male	25	0	0	F.C.C. 135	10.5		S		
1244	2	Dibden, Mr. William	male	18	0	0	S.O.C. 148	73.5		S		
1245	2	Herman, Mr. Samuel	male	49	1	2	220845	65		S		
1246	3	Dean, Miss. Elizabeth Gladys Millvina""	female	0.17	1	2	C.A. 2315	20.575		S		
1247	1	Julian, Mr. Henry Forbes	male	50	0	0	113044	26	E60	S		
1248	1	Brown, Mrs. John Murray (Caroline Lane Lamson)	female	59	2	0	11769	51.4792	C101	S		
1249	3	Lockyer, Mr. Edward	male		0	0	1222	7.8792		S		
1250	3	O'Keefe, Mr. Patrick	male		0	0	368402	7.75		Q		
1251	3	Lindell, Mrs. Edvard Bengtsson (Elin Gerda Persson)	female	30	1	0	349910	15.55		S		
1252	3	Sage, Master. William Henry	male	14.5	8	2	CA. 2343	69.55		S		
1253	2	Mallet, Mrs. Albert (Antoinette Magnin)	female	24	1	1	S.C./PARIS	37.0042		C		
1254	2	Ware, Mrs. John James (Florence Louise Long)	female	31	0	0	CA 31352	21		S		
1255	3	Strilic, Mr. Ivan	male	27	0	0	315083	8.6625		S		



PassengerId	
1260	1
1261	0
1262	0
1263	1
1264	0
1265	0
1266	1
1267	1
1268	1
1269	0
1270	0
1271	0
1272	0
1273	0
1274	1
1275	1
1276	0
1277	1
1278	0
1279	0

1.8. Mô tả chi tiết các thuộc tính trong dataset

Tên thuộc tính	Mô tả
PassengerId	ID hành khách trên tàu
Survived	1 – còn sống, 0 – chết
Pclass	Lớp ghế mà hành khách đang ngồi trên tàu (1 = hạng nhất, 2 = hạng 2, 3 = hạng 3)
Name	Họ tên hành khách
Sex	Giới tính của hành khách (nam = male, nữ = female)
Age	Độ tuổi của hành khách
SibSp	Số lượng anh/chị/em của hành khách cùng đi trong chuyến đi
Parch	Số lượng cha/mẹ/con của hành khách cùng đi trong chuyến đi
Ticket	Số vé của hành khách
Fare	Giá vé của hành khách
Cabin	Số hiệu phòng của hành khách
Embarked	Cảng lên tàu của hành khách (C = Cherbourg, Q = Queenstown, S = Southampton)

1.9. Giới thiệu các công cụ được sử dụng trong đồ án

Công cụ được sử dụng trong đồ án này là: Zeppelin

Ngôn ngữ lập trình: Python

1.9.1. Tổng quan Zeppelin

Apache Zeppelin là một ứng dụng web mã nguồn mở để thực thi, quản lý và chia sẻ các notebook tương tác cho phép thực hiện phân tích dữ liệu, truy xuất cơ sở dữ liệu, thực thi các thuật toán và trình bày các kết quả dưới dạng tài liệu tương tác. Zeppelin hỗ trợ nhiều ngôn ngữ lập trình và các công cụ phân tích dữ liệu phổ biến như Python, R, Spark, SQL, Flink, Cassandra và nhiều hơn nữa. Nó cung cấp cho người dùng một cách tiếp cận tương tác để phân tích dữ liệu và có khả năng tạo ra tài liệu tương tác với các biểu đồ, bảng và hình ảnh.

Zeppelin là một công cụ mạnh mẽ và phổ biến trong cộng đồng phân tích dữ liệu và được sử dụng rộng rãi trong các dự án phát triển và nghiên cứu khoa học.

1.9.2. Giới thiệu Python3

Python là một ngôn ngữ lập trình bậc cao cho các mục đích lập trình đa năng, do Guido van Rossum tạo ra và lần đầu ra mắt vào năm 1991. Python được thiết kế với ưu điểm mạnh là dễ đọc, dễ học và dễ nhớ. Python là ngôn ngữ có hình thức rất đơn giản, cấu trúc rõ ràng, thuận tiện cho người mới học lập trình. Cấu trúc của Python còn cho phép người sử dụng viết mã lệnh với số lần gõ phím tối thiểu. Vào tháng 7 năm 2018, Van Rossum đã từ chức Leader trong cộng đồng ngôn ngữ Python sau 30 năm lãnh đạo.

Python hoàn toàn tạo kiểu động và dùng cơ chế cấp phát bộ nhớ tự động, do vậy nó tương tự như Perl, Ruby, Scheme, Smalltalk, và Tcl. Python được phát triển trong một dự án mã mở, do tổ chức phi lợi nhuận Python Software Foundation quản lý.

Ban đầu, Python được phát triển để chạy trên nền Unix. Nhưng rồi theo thời gian, nó đã "bành trướng" sang mọi hệ điều hành từ MS-DOS đến Mac OS, OS/2, Windows, Linux và các hệ điều hành khác thuộc họ Unix. Mặc dù sự phát triển của Python có sự đóng góp của rất nhiều cá nhân, nhưng Guido van Rossum hiện nay vẫn là tác giả chủ yếu của Python. Ông giữ vai trò chủ chốt trong việc quyết định hướng phát triển của Python.

Sau đây là các đặc điểm của **Python**:

- Ngữ pháp đơn giản, dễ đọc.
- Vừa hướng thủ tục (procedural-oriented), vừa hướng đối tượng (object-oriented)
- Hỗ trợ module và hỗ trợ gói (package)
- Xử lý lỗi bằng ngoại lệ (Exception)
- Kiểu dữ liệu động ở mức cao.
- Có các bộ thư viện chuẩn và các module ngoài, đáp ứng tất cả các nhu cầu lập trình.
- Có thể nhúng vào ứng dụng như một giao tiếp kịch bản (scripting interface).

CHƯƠNG 2: THIẾT KẾ XÂY DỰNG DASHBOARD

Trình bày chi tiết các bước thực hiện trong đồ án

2.1. Quá trình nạp dữ liệu vào zeppeline

Import tập dữ liệu **train.csv**, **test.csv** và **gender_submission.csv**

FINISHED

```
%pyspark
import pandas as pd
import plotly.graph_objs as go
# Đọc dữ liệu từ file CSV
df1 = pd.read_csv("C:/test.csv")
df2 = pd.read_csv("C:/gender_submission.csv")
df3 = pd.read_csv("C:/train.csv")
print(df1)
```

```
     PassengerId  Pclass  ... Cabin Embarked
0            892       3  ...   NaN        Q
1            893       3  ...   NaN        S
2            894       2  ...   NaN        Q
3            895       3  ...   NaN        S
4            896       3  ...   NaN        S
..           ...     ...  ...   ...      ...
413         1305       3  ...   NaN        S
414         1306       1  ...  C105        C
415         1307       3  ...   NaN        S
416         1308       3  ...   NaN        S
417         1309       3  ...   NaN        C

[418 rows x 11 columns]
```

Took 0 sec. Last updated by anonymous at May 07 2023, 3:58:15 PM.

2.2. Kết hợp các bảng

Kết hợp 2 tập dữ liệu **test.csv** và **gender_submission.csv**

```
%pyspark

#join 2 file test và gender_submission dựa trên trường PassengerId
merge_df12 = pd.merge(df1, df2, on = 'PassengerId')
print(merge_df12)
```

FINISHED

```
     PassengerId  Pclass  ... Embarked Survived
0            892       3  ...        Q        0
1            893       3  ...        S        1
2            894       2  ...        Q        0
3            895       3  ...        S        0
4            896       3  ...        S        1
..           ...     ...  ...      ...      ...
413         1305       3  ...        S        0
414         1306       1  ...        C        1
415         1307       3  ...        S        0
416         1308       3  ...        S        0
417         1309       3  ...        C        0

[418 rows x 12 columns]
```

Took 0 sec. Last updated by anonymous at May 07 2023, 1:40:32 PM. (outdated)

Kết hợp 2 tập dữ liệu **train.csv và merge.df12** để tạo thành tập dữ liệu hoàn chỉnh

```
%pyspark
#nối 2 file test và merge_df12 bằng lệnh concat
merge_df = pd.concat([df3, merge_df12], ignore_index=True)
print(merge_df)
```

FINISHED

```
      PassengerId  Survived  Pclass  ...      Fare Cabin  Embarked
0               1         0       3  ...    7.2500   NaN         S
1               2         1       1  ...   71.2833   C85         C
2               3         1       3  ...    7.9250   NaN         S
3               4         1       1  ...   53.1000  C123         S
4               5         0       3  ...    8.0500   NaN         S
...           ...       ...     ...  ...       ...   ...       ...
1304         1305         0       3  ...    8.0500   NaN         S
1305         1306         1       1  ...  108.9000  C105         C
1306         1307         0       3  ...    7.2500   NaN         S
1307         1308         0       3  ...    8.0500   NaN         S
1308         1309         0       3  ...   22.3583   NaN         C

[1309 rows x 12 columns]
```

Took 0 sec. Last updated by anonymous at May 07 2023, 1:23:45 PM.

2.3. Quá trình EDA

2.3.1. Xem số lượng các dòng bị thiếu trong DataFrame

```
%pyspark
#xem số lượng bị thiếu của từng cột trong dataframe
print(merge_df.isnull().sum())
```

FINISHED

```
PassengerId       0
Survived          0
Pclass            0
Name              0
Sex               0
Age             263
SibSp             0
Parch             0
Ticket            0
Fare              1
Cabin          1014
Embarked          2
dtype: int64
```

Took 0 sec. Last updated by anonymous at May 07 2023, 1:24:04 PM.

2.3.2. Xóa các dòng có giá trị null

```
%pyspark
#xóa tất cả các dòng có giá trị null
merge_df_dropnull = merge_df.dropna()
print(merge_df_dropnull)
```

FINISHED

```
      PassengerId  Survived  Pclass  ...      Fare Cabin  Embarked
1               2         1       1  ...   71.2833   C85         C
3               4         1       1  ...   53.1000  C123         S
6               7         0       1  ...   51.8625   E46         S
10             11         1       3  ...   16.7000    G6         S
11             12         1       1  ...   26.5500  C103         S
...           ...       ...     ...  ...       ...   ...       ...
1295         1296         0       1  ...   27.7208   D40         C
1296         1297         0       2  ...   13.8625   D38         C
1298         1299         0       1  ...  211.5000   C80         C
1302         1303         1       1  ...   90.0000   C78         Q
1305         1306         1       1  ...  108.9000  C105         C
```

alhost:8080/#/

2.3.3. Xem mô tả của từng cột

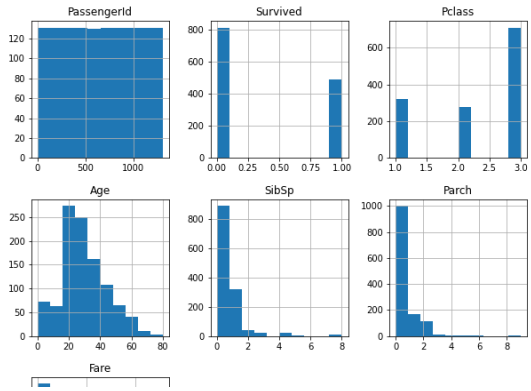
```
%pyspark
#xem mô tả thống kê của từng cột trong df
print(merge_df.describe())
```

```
       PassengerId     Survived  ...        Parch         Fare
count  1309.000000  1309.000000  ...  1309.000000  1308.000000
mean    655.000000     0.377387  ...     0.385027    33.295479
std     378.020061     0.484918  ...     0.865560    51.758668
min       1.000000     0.000000  ...     0.000000     0.000000
25%     328.000000     0.000000  ...     0.000000     7.895800
50%     655.000000     0.000000  ...     0.000000    14.454200
75%     982.000000     1.000000  ...     0.000000    31.275000
max    1309.000000     1.000000  ...     9.000000   512.329200

[8 rows x 7 columns]
```

Took 0 sec. Last updated by anonymous at May 07 2023, 1:25:07 PM.

2.3.4. Tạo biểu đồ histogram để thấy phân phối các biến số

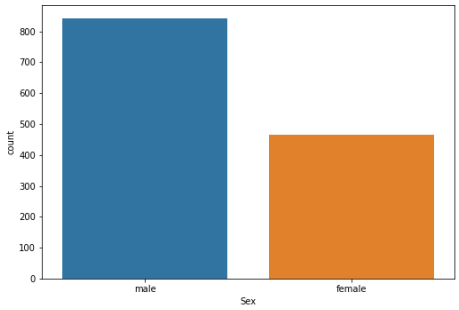
```
%pyspark
import matplotlib.pyplot as plt

#Tạo biểu đồ histogram để thấy phân phối của các biến số:
merge_df.hist(figsize=(10, 10))
plt.show()
```



2.3.5. Tạo biểu đồ countplot để xem lượng giá trị của các biến phân loại

```
%pyspark
import pandas as pd
import numpy as np
import seaborn as sns
#Tạo biểu đồ countplot để xem số lượng giá trị của các biến phân loại:
sns.countplot(x='Sex', data=merge_df)
plt.show()
```

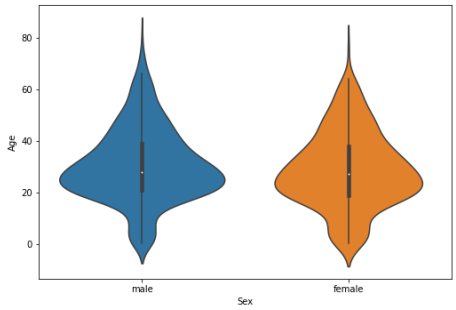
FINISHED



2.3.6. Tạo biểu đồ violinplot để xem phân bố của các biến số với các biến phân loại

```
%pyspark
#Tạo biểu đồ violinplot để xem phân bố của các biến số với các biến phân loại:
sns.violinplot(x='Sex', y='Age', data=merge_df)
plt.show()
```

FINISHED



Took 0 sec. Last updated by anonymous at May 07 2023, 1:41:50 PM.

2.3.7. Xóa cột SibSp Parch

```
%pyspark
#xóa SibSp Parch
df_drop = merge_df.drop(['SibSp', 'Parch'], axis = 1)
print(df_drop)
```

FINISHED

```
      PassengerId  Survived  Pclass  ...      Fare Cabin  Embarked
0               1         0       3  ...    7.2500   NaN         S
1               2         1       1  ...   71.2833   C85         C
2               3         1       3  ...    7.9250   NaN         S
3               4         1       1  ...   53.1000  C123         S
4               5         0       3  ...    8.0500   NaN         S
...           ...       ...     ...  ...       ...   ...       ...
1304         1305         0       3  ...    8.0500   NaN         S
1305         1306         1       1  ...  108.9000  C105         C
1306         1307         0       3  ...    7.2500   NaN         S
1307         1308         0       3  ...    8.0500   NaN         S
1308         1309         0       3  ...   22.3583   NaN         C

[1309 rows x 10 columns]
```

:alhost:8080/#/ Last updated by anonymous at May 07 2023, 1:42:05 PM.

2.3.8. Sort theo độ tuổi

```
%pyspark
#sort theo độ tuổi
df_sort_by_age = df_drop.sort_values('Age', ascending = True)
print(df_sort_by_age)
```

FINISHED

```
      PassengerId  Survived  Pclass  ...     Fare Cabin  Embarked
1245         1246         1       3  ...  20.5750   NaN         S
1092         1093         0       3  ...  14.4000   NaN         S
803           804         1       3  ...   8.5167   NaN         C
755           756         1       2  ...  14.5000   NaN         S
469           470         1       3  ...  19.2583   NaN         C
...           ...       ...     ...  ...      ...   ...       ...
1299         1300         1       3  ...   7.7208   NaN         Q
1301         1302         1       3  ...   7.7500   NaN         Q
1304         1305         0       3  ...   8.0500   NaN         S
1307         1308         0       3  ...   8.0500   NaN         S
1308         1309         0       3  ...  22.3583   NaN         C

[1309 rows x 10 columns]
```

Took 0 sec. Last updated by anonymous at May 07 2023, 1:42:15 PM.

2.3.9. Tính toán trung bình tuổi

```
%pyspark
#tính toán trung bình tuổi
mean_age = merge_df['Age'].mean()
print(mean_age)
```

FINISHED

```
29.881137667304014
```

Took 0 sec. Last updated by anonymous at May 07 2023, 1:42:30 PM.

2.3.10. Tạo cột giá trị tuổi trung bình với những giá trị null được thay thế bằng mean(age)

```
%pyspark
#tạo cột giá trị tuổi trung bình với những giá trị null được thay thế bằng mean(age)
replace_null_values = merge_df['Age'].fillna(mean_age)
print(replace_null_values)
```

FINISHED

```
0       22.000000
1       38.000000
2       26.000000
3       35.000000
4       35.000000
          ...
1304    29.881138
1305    39.000000
1306    38.500000
1307    29.881138
1308    29.881138
Name: Age, Length: 1309, dtype: float64
```

Took 0 sec. Last updated by anonymous at May 07 2023, 1:42:39 PM.

2.3.11. Thay thế cột Age

```
%pyspark
#xóa cột cũ trong dataframe
del_col_df = merge_df.drop("Age", axis = 1)

#thêm cột mới
add_col_df = del_col_df.assign(MeanAge=replace_null_values)

#hiển thị 2 số sau dấu phẩy
add_col_df['MeanAge'] = add_col_df['MeanAge'].round(2)

print(add_col_df)
```

FINISHED

```
      PassengerId  Survived  Pclass  ... Cabin Embarked  MeanAge
0               1         0       3  ...   NaN        S    22.00
1               2         1       1  ...   C85        C    38.00
2               3         1       3  ...   NaN        S    26.00
3               4         1       1  ...  C123        S    35.00
4               5         0       3  ...   NaN        S    35.00
...           ...       ...     ...  ...   ...      ...      ...
1304         1305         0       3  ...   NaN        S    29.88
1305         1306         1       1  ...  C105        C    39.00
1306         1307         0       3  ...   NaN        S    38.50
1307         1308         0       3  ...   NaN        S    29.88
1308         1309         0       3  ...   NaN        C    29.88

[1309 rows x 12 columns]
```

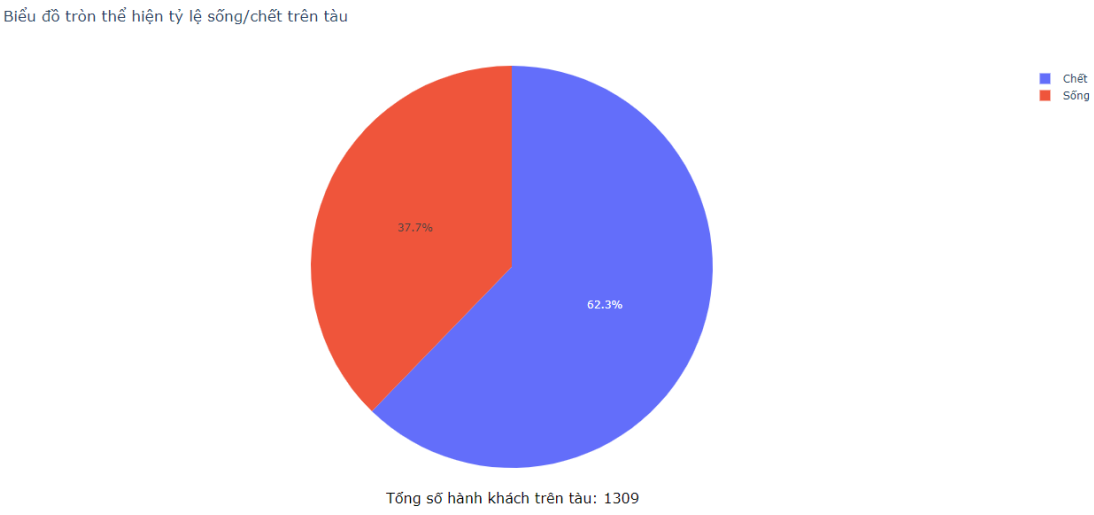
Took 0 sec. Last updated by anonymous at May 07 2023, 2:14:31 PM.

2.4. Biểu đồ tròn thể hiện phần trăm sống sót trên tàu

```
%pyspark
#biểu đồ tròn thể hiện phần trăm sống sót trên tàu

import plotly.express as px

# Tính tổng số hành khách sống/chết trên tàu
survived_count = add_col_df[add_col_df['Survived'] == 1]['Survived'].count()
dead_count = add_col_df[add_col_df['Survived'] == 0]['Survived'].count()
total_passengers = add_col_df['Survived'].count()

# Tạo dataframe mới với dữ liệu tổng hợp sống/chết
df_survived = pd.DataFrame({'Status': ['Sống', 'Chết'],
                            'Count': [survived_count, dead_count]})

# Tạo biểu đồ tròn
fig = px.pie(df_survived, values='Count', names='Status', title='Biểu đồ tròn thể hiện tỷ lệ sống/chết trên tàu')

# Thêm chú thích với tổng số hành khách
fig.update_layout(annotations=[dict(text=f"Tổng số hành khách trên tàu: {total_passengers}",
                                    x=0.5, y=-0.1, showarrow=False,
                                    font=dict(size=16, color='black'))])

# Hiển thị biểu đồ
fig.show()
```

FINISHED

Took 0 sec. Last updated by anonymous at May 07 2023, 2:24:01 PM.



2.5. Biểu đồ tròn thể hiện số hành khách nam sống/chết trên tàu

```
#Biểu đồ tròn thể hiện số hành khách nam sống/chết trên tàu
import plotly.express as px
# Tính tổng số hành khách nam sống/chết trên tàu
survived_count_male = add_col_df[(add_col_df['Survived'] == 1) & (add_col_df['Sex'] == 'male')]['Survived'].count()
dead_count_male = add_col_df[(add_col_df['Survived'] == 0) & (add_col_df['Sex'] == 'male')]['Survived'].count()
total_passengers_male = add_col_df[add_col_df['Sex'] == 'male']['Survived'].count()

# Tạo dataframe mới với dữ liệu tổng hợp sống/chết nam
df_survived_male = pd.DataFrame({'Status': ['Survived', 'Dead'],
                                 'Count': [survived_count_male, dead_count_male]})

# Tạo biểu đồ tròn
fig_male = px.pie(df_survived_male, values='Count', names='Status',
                  title='Biểu đồ tròn thể hiện tỷ lệ sống/chết của hành khách nam trên tàu')

# Thêm chú thích với tổng số hành khách nam
fig_male.update_layout(annotations=[dict(text=f"Tổng số hành khách nam trên tàu: {total_passengers_male}",
                                         x=0.5, y=-0.1, showarrow=False,
                                         font=dict(size=16, color='black'))])

#Hiển thị biểu đồ
fig_male.show()
```

FINISHED

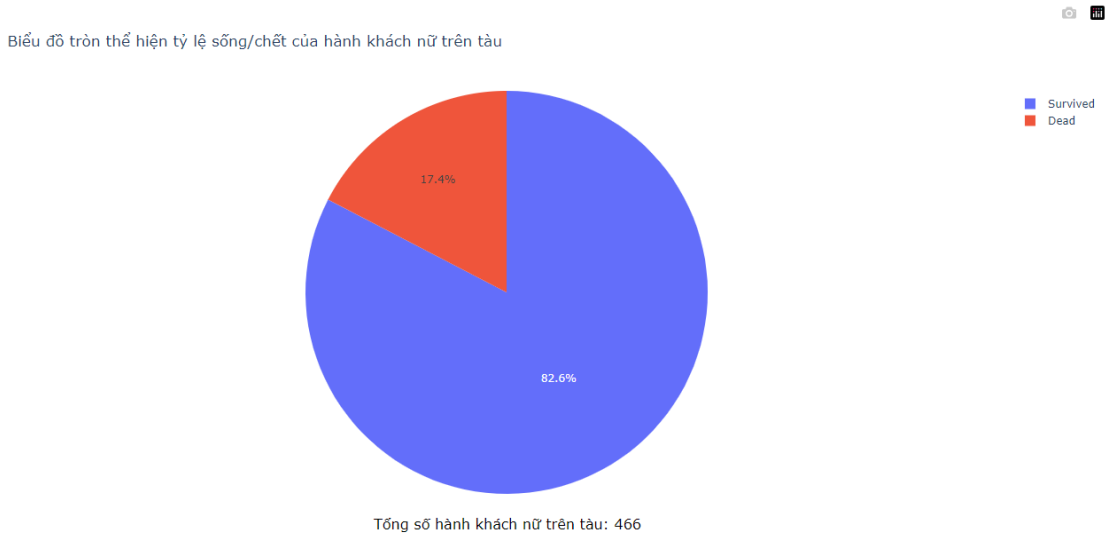
Took 1 sec. Last updated by anonymous at May 07 2023, 2:25:57 PM. (outdated)

2.6. Biểu đồ tròn thể hiện số hành khách nữ sống/chết trên tàu

```
#Biểu đồ tròn thể hiện số hành khách nữ sống/chết trên tàu
import plotly.express as px
# Tính tổng số hành khách nữ sống/chết trên tàu
survived_count_female = add_col_df[(add_col_df['Survived'] == 1) & (add_col_df['Sex'] == 'female')]['Survived'].count()
dead_count_female = add_col_df[(add_col_df['Survived'] == 0) & (add_col_df['Sex'] == 'female')]['Survived'].count()
total_passengers_female = add_col_df[add_col_df['Sex'] == 'female']['Survived'].count()

# Tạo dataframe mới với dữ liệu tổng hợp sống/chết nữ
df_survived_female = pd.DataFrame({'Status': ['Survived', 'Dead'],
                                   'Count': [survived_count_female, dead_count_female]})

# Tạo biểu đồ tròn
fig_female = px.pie(df_survived_female, values='Count', names='Status',
                    title='Biểu đồ tròn thể hiện tỷ lệ sống/chết của hành khách nữ trên tàu')

# Thêm chú thích với tổng số hành khách nữ
fig_female.update_layout(annotations=[dict(text=f"Tổng số hành khách nữ trên tàu: {total_passengers_female}",
                                           x=0.5, y=-0.1, showarrow=False,
                                           font=dict(size=16, color='black'))])

# Hiển thị biểu đồ
fig_female.show()
```

FINISHED

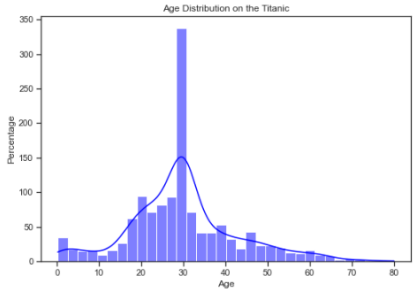
Took 0 sec. Last updated by anonymous at May 07 2023, 2:26:09 PM. (outdated)



2.7. Biểu đồ cột thể hiện phân phối độ tuổi trên tàu

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

#Biểu đồ cột thể hiện phân phối độ tuổi trên tàu
# Tạo biểu đồ phân phối sử dụng histplot từ seaborn
sns.set(style='ticks')
sns.histplot(add_col_df['MeanAge'], kde=True, color='blue')

# Thiết lập tiêu đề và nhãn trục
plt.title('Age Distribution on the Titanic')
plt.xlabel('Age')
plt.ylabel('Percentage')

# Hiển thị biểu đồ
plt.show()
```

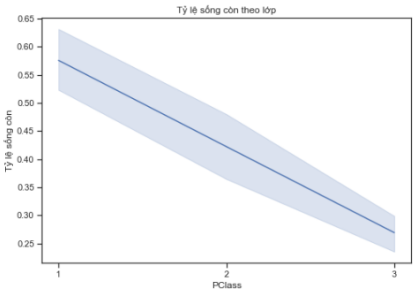
FINISHED



Took 0 sec. Last updated by anonymous at May 07 2023, 2:32:44 PM. (outdated)

2.8. Tạo biểu đồ đường so sánh tỷ lệ sống còn theo PCLASS

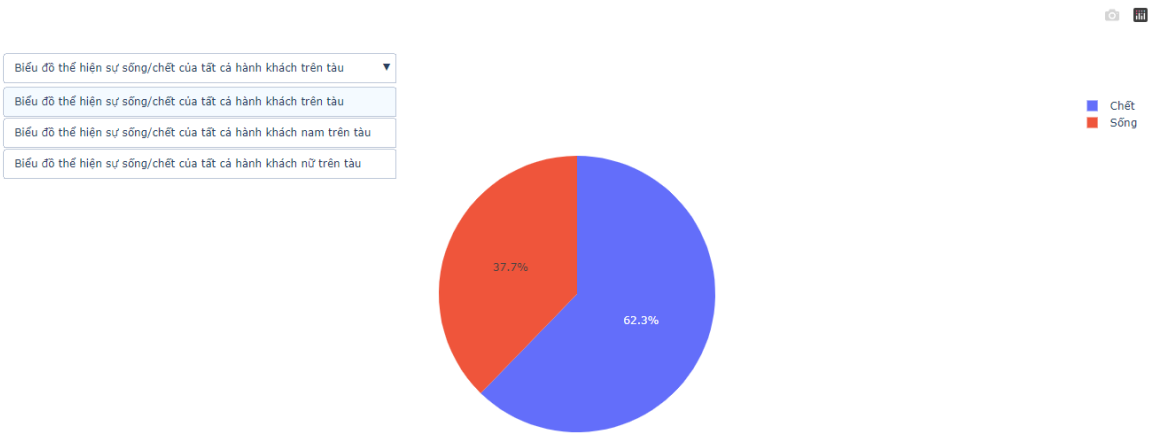
```
import seaborn as sns
import matplotlib.pyplot as plt

# Tạo biểu đồ đường so sánh tỷ lệ sống còn theo PCLASS
sns.lineplot(x="Pclass", y="Survived", data=add_col_df)


plt.xticks([1, 2, 3])

# Thiết lập tiêu đề và nhãn trục
plt.title("Tỷ lệ sống còn theo lớp")
plt.xlabel("PClass")
plt.ylabel("Tỷ lệ sống còn")

# Hiển thị biểu đồ
plt.show()
```



Took 1 sec. Last updated by anonymous at May 07 2023, 2:36:04 PM.

2.9. Tạo ComboBox hiển thị 3 biểu đồ tròn

```
import plotly.graph_objects as go
import plotly.subplots as sp
import plotly.express as px

# Tạo bố cục subplot
fig_combined = sp.make_subplots(rows=1, cols=3, subplot_titles=['', '', ''], specs=[[{'type': 'pie'}, {'type': 'pie'}, {'type': 'pie'}]])


# Cấu hình dropdown menu
fig_combined.update_layout(
    updatemenus=[
        dict(
            buttons=[
                {'label': 'Biểu đồ tròn', 'method': 'update', 'args': [{'visible': [True, False, False]}, {'title': ' '}]},
                {'label': 'Biểu đồ nam', 'method': 'update', 'args': [{'visible': [False, True, False]}, {'title': ' '}]},
                {'label': 'Biểu đồ nữ', 'method': 'update', 'args': [{'visible': [False, False, True]}, {'title': ''}]}
            ],
            direction='down',
            showactive=True,
            x=-0.1,
            xanchor='left',
            y=1.1,
            yanchor='top'
        ),
    ]
)


# Thêm biểu đồ tròn và biểu đồ nam vào subplot
fig_combined.add_trace(fig.data[0], row=1, col=2)
fig_combined.add_trace(fig_male.data[0], row=1, col=2)
fig_combined.add_trace(fig_female.data[0], row=1, col=2)

# Cấu hình hiển thị ban đầu
fig_combined.data[0].visible = True  # Hiển thị biểu đồ tròn
fig_combined.data[1].visible = False  # Ẩn biểu đồ nam
fig_combined.data[2].visible = False  # Ẩn biểu đồ nữ

# Cập nhật hàm callback của dropdown menu
def update_dropdown(change):
    if change['new'] == 'Biểu đồ tròn':
        fig_combined.data[0].visible = True
        fig_combined.data[1].visible = False
        fig_combined.data[2].visible = False
        fig_combined.update_layout(title='Biểu đồ tròn')
```



CHƯƠNG 3: KẾT LUẬN

Báo cáo kết quả được sau khi hoàn thành đồ án, nêu ra những hạn chế còn tồn tại, định hướng phát triển trong trương lai, các tài liệu tham khảo. Bảng phân công công việc cụ thể và mức độ hoàn thành công việc.

3.1. Kết quả đạt được

Trong đề tài này, nhóm đã tìm hiểu và vận dụng kiến thức đạt được kết quả như sau :

- Áp dụng được biểu đồ có thể tương tác bằng cách trỏ chuột hiện chú thích và đổi màu.
- Năm vững kiến thức và có thể vận dụng , xây dựng một dashboard hoàn chỉnh dùng để khai thác dữ liệu.
- Áp dụng được biểu đồ có sử dụng combobox.
- Có sử dụng nhiều bảng dữ liệu kết hợp với nhau.
- Áp dụng các thao tác xóa cột, nhóm (group), nối (concat), kết hợp (join), lọc (filter), thay thế cột bằng tính toán dữ liệu.
- Tạo ra được biểu đồ thể hiện đường mục tiêu.

3.2. Kết luận rút ra được từ dashboard

Dựa trên phân tích dữ liệu từ dự án Titanic trên Kaggle, chúng ta thu được các kết quả sau:

- Tỷ lệ sống còn: Tỷ lệ sống còn trên tàu Titanic là khoảng 38%, điều này cho thấy thảm họa Titanic đã gây ra nhiều thiệt hại và gây mất mát lớn.

- Giới tính: Tỷ lệ sống còn của phụ nữ là khoảng 74%, trong khi tỷ lệ sống còn của nam giới chỉ là khoảng 19%. Điều này cho thấy có sự ưu tiên trong việc cứu hộ cho phụ nữ trong thảm họa này.

- Lớp hành khách: Hành khách ở lớp hạng nhất có tỷ lệ sống còn cao hơn so với hành khách ở lớp hạng thứ hai và thứ ba. Điều này cho thấy sự ưu tiên trong việc cứu hộ cho nhóm hành khách giàu có và có địa vị xã hội cao.

- Tuổi: Tỷ lệ sống còn của trẻ em (dưới 18 tuổi) cao hơn so với người lớn. Tuy nhiên, người cao tuổi (trên 65 tuổi) có tỷ lệ sống còn thấp nhất. Điều này có thể cho thấy việc ưu tiên cứu hộ cho nhóm yếu thế và khả năng di chuyển hạn chế của người cao tuổi.

Dựa trên các kết quả trên, chúng ta có thể rút ra một số nhận định và khuyến nghị như sau:

- Nâng cao quy định và chuẩn bị cho các biện pháp cứu hộ để tăng cường khả năng sống sót trong các tình huống thảm họa tương tự.
- Tăng cường sự chú trọng đến an toàn và chuẩn bị sẵn sàng cho nhóm yếu thế như trẻ em và người cao tuổi trong các kế hoạch cứu hộ.
- Đảm bảo sự công bằng và không phân biệt đối xử dựa trên giới tính trong quá trình cứu hộ và phân chia tài nguyên trong tình huống khẩn cấp.

Tuy nhiên, cần lưu ý rằng kết quả phân tích chỉ dựa trên dữ liệu từ dự án Titanic và không thể áp dụng trực tiếp vào các tình huống khác. Để có kết quả phân tích chi tiết và đáng tin cậy hơn.

3.3. Những hạn chế

- Chưa hoàn chỉnh ComboBox (định dạng layout, hiển thị title của đồ thị).
- Quá trình EDA rườm rà và có thể chưa đầy đủ.
- Biểu đồ tương tác còn khá đơn giản.
- Chưa xử lý thay đổi dữ liệu.

3.4. Bảng phân công nhiệm vụ trong nhóm

STT	Nhiệm vụ	Người phụ trách	Ghi chú
1	EDA, Word	Cường, Hậu, Việt	Hoàn thành 100%
2	Tìm dataset	Hậu, Cường, Việt	Hoàn thành 100%
3	Vẽ biểu đồ	Việt, Hậu, Cường	Hoàn thành 100%
4			
5			
6			
7			
8			
9			
10			
11			
12			

3.5. Tài liệu tham khảo

- Slide bài giảng của thầy Lê Minh Tân
- Công cụ sử dụng:
    - Pham Thi Hong Anh, Giới thiệu Visualization với ploty với tập dataset Titanic, 16/01/2018, https://viblo.asia/p/gioi-thieu-visualization-data-voi-plotly-voi-tap-dataset-titanic-gAm5y8bqldb, truy cập ngày: 5/5/2023.
    - Vietnambiz, Biểu đồ hộp (Box Plot) là gì ? Đặc trưng và ví dụ, 12/11/2019, https://vietnambiz.vn/bieu-do-hop-box-plot-la-gi-dac-trung-va-vi-du-20191112102052212.htm, truy cập ngày 6/5/2023.