MIT | Academy of Engineering

EDS Project on:

Exploring the Titanic data

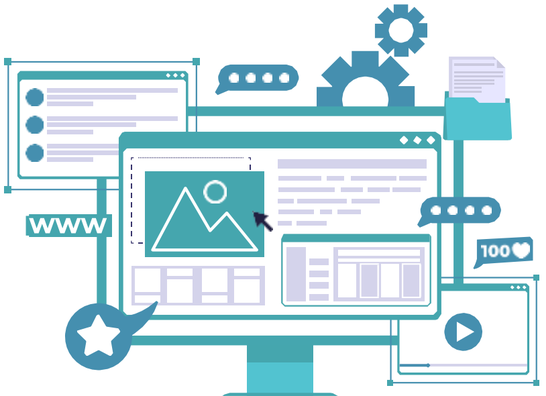
Presented By:

737 :Krushna kulkarni

731 :Piyush Jadhav

738 :Pradyumna Kulkarni

740 : Omkar Varote



[64]
```
import pandas as pd
import numpy as np
from sklearn.cluster import KMeans
```

0s
```
import numpy as np



df = pd.read_csv('/content/titanic.csv')

# Convert the DataFrame to a numpy array
numpy_array = df.to_numpy()
```

[69]
```
df.head()
```

	Age	Cabin	Embarked	Fare	Name	Parch	PassengerId	Pclass	Sex	SibSp	Survived	Ticket	Title	Family_Size
0	22.0	NaN	S	7.2500	Braund, Mr. Owen Harris	0	1	3	male	1	0.0	A/5 21171	Mr	1
1	38.0	C85	C	71.2833	Cumings, Mrs. John Bradley (Florence Briggs Th...	0	2	1	female	1	1.0	PC 17599	Mrs	1
2	26.0	NaN	S	7.9250	Heikkinen, Miss. Laina	0	3	3	female	0	1.0	STON/O2. 3101282	Miss	0
3	35.0	C123	S	53.1000	Futrelle, Mrs. Jacques Heath (Lily May Peel)	0	4	1	female	1	1.0	113803	Mrs	1
4	35.0	NaN	S	8.0500	Allen, Mr. William Henry	0	5	3	male	0	0.0	373450	Mr	0

0s [70]
```
df.shape
```

```
(891, 14)
```

+ Code + Text

Disk

Toggle header visibility

[71]
```
df.dtypes
```

```
Age            float64
Cabin           object
Embarked        object
Fare           float64
Name            object
Parch            int64
PassengerId      int64
Pclass           int64
Sex             object
SibSp            int64
Survived       float64
Ticket          object
Title           object
Family_Size      int64
dtype: object
```

```
df.isnull().sum()
```

```
Age              0
Cabin          687
Embarked         0
Fare             0
Name             0
Parch            0
PassengerId      0
Pclass           0
Sex              0
SibSp            0
Survived         0
Ticket           0
Title            0
Family_Size      0
dtype: int64
```

[73]
```
mean_age = df['Age']
df['Age'].fillna(mean_age, inplace=True)
```

[75]
```
df.groupby('Sex')['Age'].mean()
```

```
Sex
female    27.648089
male      30.423172
Name: Age, dtype: float64
```

```
df = df.sort_values('Fare', ascending=False)
print(df.sort_values('Fare', ascending=False))
```

```
      Age        Cabin Embarked      Fare                                Name  \
258  35.0          NaN        C  512.3292                   Ward, Miss. Anna
679  36.0  B51 B53 B55        C  512.3292  Cardeza, Mr. Thomas Drake Martinez
737  35.0         B101        C  512.3292             Lesurer, Mr. Gustave J
88   23.0  C23 C25 C27        S  263.0000         Fortune, Miss. Mabel Helen
438  64.0  C23 C25 C27        S  263.0000                 Fortune, Mr. Mark
..    ...          ...      ...       ...                               ...
277  30.0          NaN        S    0.0000        Parkes, Mr. Francis "Frank"
271  25.0          NaN        S    0.0000       Tornquist, Mr. William Henry
597  49.0          NaN        S    0.0000               Johnson, Mr. Alfred
466  30.0          NaN        S    0.0000             Campbell, Mr. William
674  30.0          NaN        S    0.0000         Watson, Mr. Ennis Hastings

     Parch  PassengerId  Pclass     Sex  SibSp  Survived    Ticket Title  \
258      0          259       1  female      0       1.0  PC 17755  Miss
679      1          680       1    male      0       1.0  PC 17755    Mr
737      0          738       1    male      0       1.0  PC 17755    Mr
88       2           89       1  female      3       1.0     19950  Miss
438      4          439       1    male      1       0.0     19950    Mr
..     ...          ...     ...     ...    ...       ...       ...   ...
277      0          278       2    male      0       0.0    239853    Mr
271      0          272       3    male      0       1.0      LINE    Mr
597      0          598       3    male      0       0.0      LINE    Mr
466      0          467       2    male      0       0.0    239853    Mr
674      0          675       2    male      0       0.0    239856    Mr

     Family_Size
258            0
679            1
737            0
88             5
438            5
..           ...
277            0
271            0
597            0
466            0
674            0
```

+ Code + Text

```
df[df['Age'] > 30]
```

	Age	Cabin	Embarked	Fare	Name	Parch	PassengerId	Pclass	Sex	SibSp	Survived	Ticket	Title	Family_Size
258	35.0	NaN	C	512.3292	Ward, Miss. Anna	0	259	1	female	0	1.0	PC 17755	Miss	0
737	35.0	B101	C	512.3292	Lesurer, Mr. Gustave J	0	738	1	male	0	1.0	PC 17755	Mr	0
679	36.0	B51 B53 B55	C	512.3292	Cardeza, Mr. Thomas Drake Martinez	1	680	1	male	0	1.0	PC 17755	Mr	1
438	64.0	C23 C25 C27	S	263.0000	Fortune, Mr. Mark	4	439	1	male	1	0.0	19950	Mr	5
299	50.0	B58 B60	C	247.5208	Baxter, Mrs. James (Helene DeLaudeniere Chaput)	1	300	1	female	0	1.0	PC 17558	Mrs	1
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
597	49.0	NaN	S	0.0000	Johnson, Mr. Alfred	0	598	3	male	0	0.0	LINE	Mr	0
806	39.0	A36	S	0.0000	Andrews, Mr. Thomas Jr	0	807	1	male	0	0.0	112050	Mr	0
179	36.0	NaN	S	0.0000	Leonard, Mr. Lionel	0	180	3	male	0	0.0	LINE	Mr	0
263	40.0	B94	S	0.0000	Harrison, Mr. William	0	264	1	male	0	0.0	112059	Mr	0
822	38.0	NaN	S	0.0000	Reuchlin, Jonkheer. John George	0	823	1	male	0	0.0	19972	Mr	0

323 rows × 14 columns

```
df['Family_Total'] = df['SibSp'] + df['Parch']
```

```
[78] df = df.dropna(subset=['Embarked'])
     print(df.dropna(subset=['Embarked']))
```

```
      Age        Cabin Embarked      Fare                                Name  \
258  35.0          NaN        C  512.3292                  Ward, Miss. Anna
737  35.0         B101        C  512.3292              Lesurer, Mr. Gustave J
679  36.0  B51 B53 B55        C  512.3292  Cardeza, Mr. Thomas Drake Martinez
88   23.0  C23 C25 C27        S  263.0000          Fortune, Miss. Mabel Helen
27   19.0  C23 C25 C27        S  263.0000      Fortune, Mr. Charles Alexander
..    ...          ...      ...       ...                                 ...
633  30.0          NaN        S    0.0000       Parr, Mr. William Henry Marsh
413  30.0          NaN        S    0.0000     Cunningham, Mr. Alfred Fleming
822  38.0          NaN        S    0.0000    Reuchlin, Jonkheer. John George
732  30.0          NaN        S    0.0000               Knight, Mr. Robert J
674  30.0          NaN        S    0.0000         Watson, Mr. Ennis Hastings
```

+ Code + Text

[78]
```
     Family_Size  Family_Total
258            0             0
737            0             0
679            1             1
88             5             5
27             5             5
..           ...           ...
633            0             0
413            0             0
822            0             0
732            0             0
674            0             0

[891 rows x 15 columns]
```

[79]
```
df = df.rename(columns={'Survived': 'IsSurvived'})
```

[80]
```
df.to_csv('modified_data.csv', index=False)
```

[81]
```
mean_age = np.mean(numpy_array[:, 0])
print(np.mean(numpy_array[:, 0]))
```

```
29.44519640852974
```

```
median_fare = np.median(numpy_array[:, 3])
print(np.median(numpy_array[:, 3]))
```

```
14.4542
```

[83]
```
max_age = np.max(numpy_array[:, 0])
print(np.max(numpy_array[:, 0]))
```

```
80.0
```

[84]
```
std_family_size = np.std(numpy_array[:, 13])
print(np.std(numpy_array[:, 13]))
```

```
1.6125528671095077
```

```
[85] num_males = np.count_nonzero(numpy_array[:, 8] == 'male')
     print(np.count_nonzero(numpy_array[:, 8] == 'male'))
```

```
577
```

```
[86] num_females = np.count_nonzero(numpy_array[:, 8] == 'female')
     print(np.count_nonzero(numpy_array[:, 8] == 'female'))
```

```
314
```

```
reshaped_array = numpy_array.reshape((-1, 2))
print(numpy_array.reshape((-1, 2)))
```

```
[[22.0 nan]
 ['S' 7.25]
 ['Braund, Mr. Owen Harris' 0]
 ...
 ['male' 0]
 [0.0 '370376']
 ['Mr' 0]]
```

```
[ ] transposed_array = np.transpose(numpy_array)
    print(numpy_array)
```

```
[[22.0 nan 'S' ... 'A/5 21171' 'Mr' 1]
 [38.0 'C85' 'C' ... 'PC 17599' 'Mrs' 1]
 [26.0 nan 'S' ... 'STON/O2. 3101282' 'Miss' 0]
 ...
 [22.0 nan 'S' ... 'W./C. 6607' 'Miss' 3]
 [26.0 'C148' 'C' ... '111369' 'Mr' 0]
 [32.0 nan 'Q' ... '370376' 'Mr' 0]]
```

```
[ ] print(numpy_array.ndim)
```

```
2
```

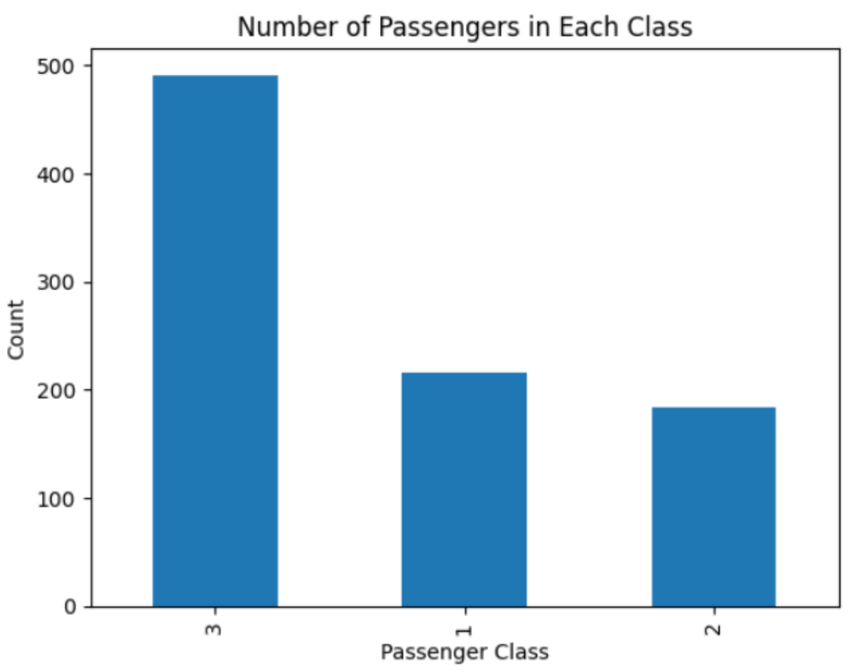
```
column_index = 3  # Assuming you want to sort by the 4th column
sorted_indices = numpy_array[:, column_index].argsort()
print(sorted_indices)
```

```
[271 597 302 633 277 413 674 263 466 732 179 806 481 822 815 378 872 326
 843 818 371 202 654 143 411 825 129 804 477 611 884 131 465 210 363 784
 721 631 243 590 433 761 127 354 553 598 522 875 661 244 693  19 203 773
  26 798 780 524 531 532  60 762 568  57 352 367  36 832 859 296 376 470
   0  94 365 592 320 250 227 212 785 491 425 154 514 786 663 478 845 648
 649 235 502 715  75 699 106 468 388 697 629 421 156 778 727 703 214 428
 767 208 776 198 196 790 828 749 775 274 264 459 406 368 501 510 525 260
 560 593 613 300 289 280 276 573 141 890  32  47 116 126 162 231 408 851
 379 442 675 246 267 764 107 805 807 554 667 719  82 756 688 499 395 391
 146  51 552 653  91 192 396 623 833 315 281  14 175 704 640 771 569 422
 358  28  44 359 810 870 606 223 108 601 321 287 384 656  42 410 650 646
 420 794 335 847 877 100 313 101 628 294 455 105 878  76 521 753 881 738
 130 519 189 566 561  29 739 400 636 115 382  68 744 414 729 216 528 104
 173 579 816 840   2 392 664  22 304 454   4  37 283 574 814 511 696  12
 668 429 563  89  90 837 372 564 401 676 152  87 157 415  45 758 461 338
  95 614 121 488 603 220 222 112 482 588 204 494  77 589 464 444 680  67
 834 769 197 652   5 803 103 404  69 163 534 836 349 471 821 725 500 285
 158 844 797 584 293  80 873 138 682 350 855 225  40 713 282 355 441 286
 770 868  81 200 752 503 483 113 402 474 677 876 687 251 205 234 226 841
 398 238 570  70 219 812 772 883 619  33 265  66 439 516 526 717 458  56
  84 672 242 882 869 172   8  39 125 144 861 757 529 389 239 479 626 322
 303 823 751 777  79 150 399 292 418 417 635 800 463 795 576 443 343 346
 228 723 864 345 344 808 666 288 342 865 673  17  21 327 722 714 221 733
 387 134 149 886 658 747 357 178 191 199 123 734 213 190 691 232 562 695
 706 473 866 547 317 860 616 423 111  73 702 620 240 830 362 114 495 578
 247 575 755 153 538 132 760 586 135 181 497 140  65 255 852 709 657 718
 364 241 188 186 612  46 605 381 622 161 142  85 206 403 489 348  15 735
 160 743 253 347 624 431 639 617 394  10 559  49 353 333  38  18 407 831
 437 207 731 469 448 858 644 272 446 451 490 254 424 279 746 548 328 165
 788  93 405 476 726 117 211  41   7  24 567 374  48 184 128 533 508 774
 750 651  98 301 330 888 783 874 308 595 799 565 768 419 517 109 811 409
 229 176 485 662 168 796 862 426 854 506 236  20 728 543  53 594 249 284
 259 705 133 312 582 193 880 316 148 546 518  99 530 432 340 427 545 551
 397 791 801 637 237 440 535 314 136 707 701 512 572 555 430 536 187 447
 507 456 694 711 857 604 467  11 252 460 217 600 194  30 361 295  64  58
 472 450 452 819 360 642 634  63 167 331 177  78 323 171 787 885 278  16
 487 273 839 740 580 630 782 887 889 122   9 449 607 632 298 492 174 793
 842 209 270 541 610 813 542 119  13 850 261 233  25 182 625 416 596 720
 848 170 515 147 436 736  86  96 351 647  55  23 339 145 549 827 817 462
 618 670 684 183 853 556 766 638 686 164  50 824 266 583  43 685 608 602
 683  71 386 480 678  59  83 539 493 710 185 867 571 457   6 671 475  35
 383 712 669 110 248 871 621 137 748 809   3 724 166 356 370 434 577 692
  74 826 838 509 643 169 309 599 690 781 523 329 513  92 155  54  97 754
 615 336 151 369 641 792 863 159 846 180 324 201 745 540   1 120 385 665
  72 655 366 218 681  52 645 124 102 765 627 275 591 496 290 741 587 789
```

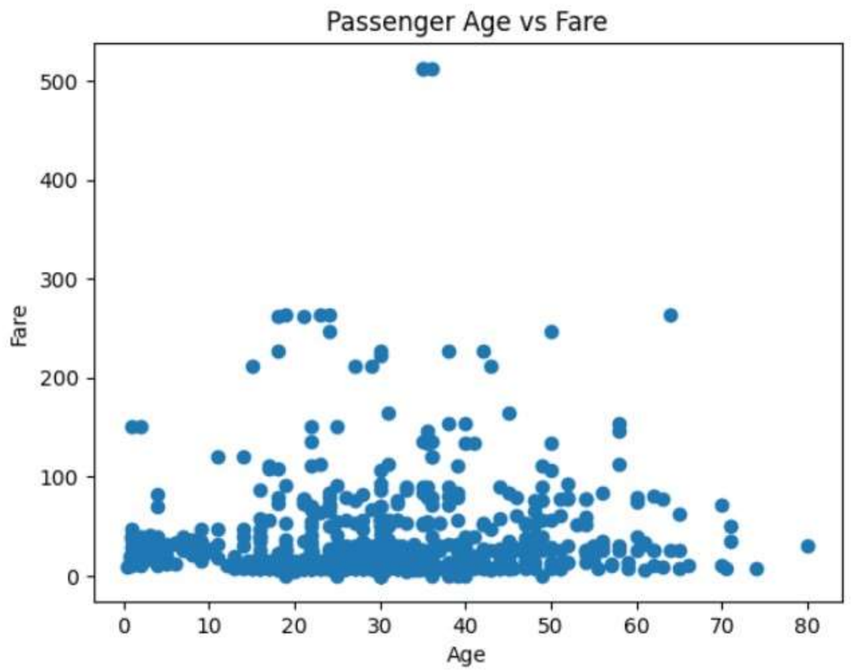
+ Code + Text

RAM Disk

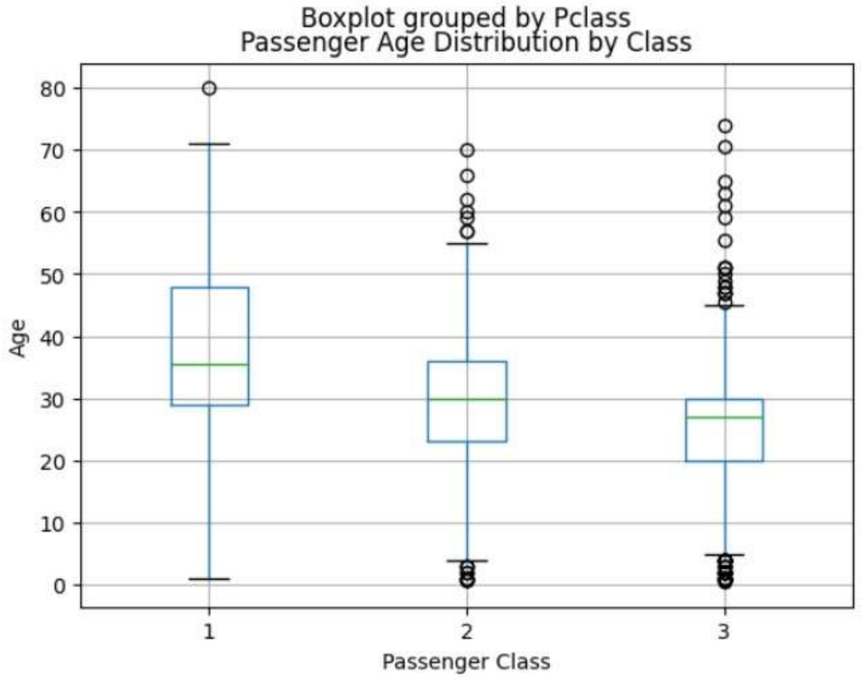
```
passenger_class_counts = data['Pclass'].value_counts()
passenger_class_counts.plot(kind='bar')
plt.xlabel('Passenger Class')
plt.ylabel('Count')
plt.title('Number of Passengers in Each Class')
plt.show()
```



```
plt.scatter(data['Age'], data['Fare'])
plt.xlabel('Age')
plt.ylabel('Fare')
plt.title('Passenger Age vs Fare')
plt.show()
```



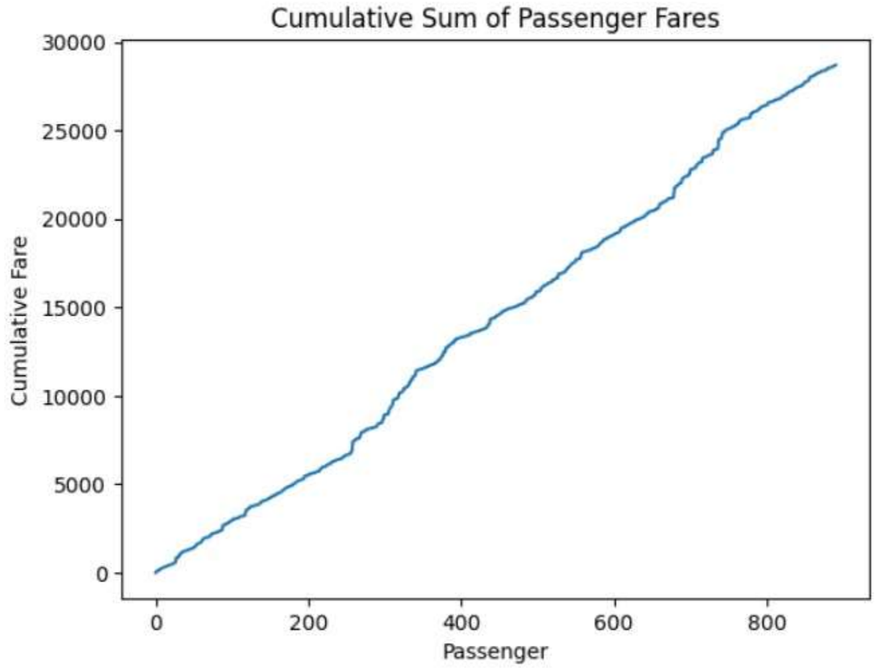
```
data.boxplot(column='Age', by='Pclass')
plt.xlabel('Passenger Class')
plt.ylabel('Age')
plt.title('Passenger Age Distribution by Class')
plt.show()
```



+ Code + Text

RAM Disk

```
cumulative_fares = data['Fare'].cumsum()
plt.plot(cumulative_fares)
plt.xlabel('Passenger')
plt.ylabel('Cumulative Fare')
plt.title('Cumulative Sum of Passenger Fares')
plt.show()
```



[9]
```
count_1= len(df.loc[df['Pclass']==1])
count_2= len(df.loc[df['Pclass']==2])
count_3= len(df.loc[df['Pclass']==3])

print("No of passangers of 1st class:" ,count_1)
print("No of passangers of 2nd class:" ,count_2)
print("No of passangers of 3rd class:" ,count_3)
```

```
No of passangers of 1st class: 216
No of passangers of 2nd class: 184
No of passangers of 3rd class: 491
```

+ Code + Text

RAM Disk

```
[106] count = len(df.loc[df['Survived']==1])
      print("No of passangers who survived:",count)
```

```
No of passangers who survived: 342
```

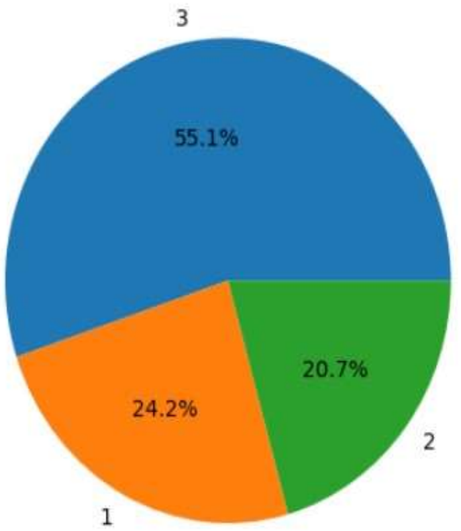
```
[104] df = pd.read_csv('/content/titanic.csv')
      count = len(df.loc[df['Survived']==0])
      print("No of passangers who didnt survived:",count)
```

```
No of passangers who didnt survived: 549
```

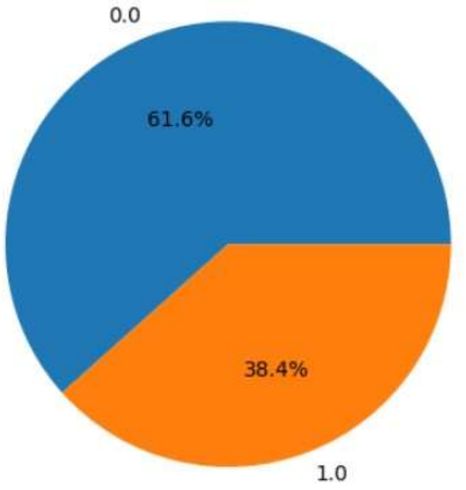
```
a=df.groupby("Embarked").count()

print(a)
```

```
          Age  Cabin  Fare  Name  Parch  PassengerId  Pclass  Sex  SibSp  \
Embarked
C         169     70   169   169    169          169     169  169    169
Q          77      4    77    77     77           77      77   77     77
S         645    130   645   645    645          645     645  645    645

          Survived  Ticket  Title  Family_Size
Embarked
C              169     169    169          169
Q               77      77     77           77
S              645     645    645          645
```

```
class_counts = df['Pclass'].value_counts()

plt.pie(class_counts, labels=class_counts.index, autopct='%1.1f%%')
plt.title('Passenger Class Distribution')
plt.show()
```

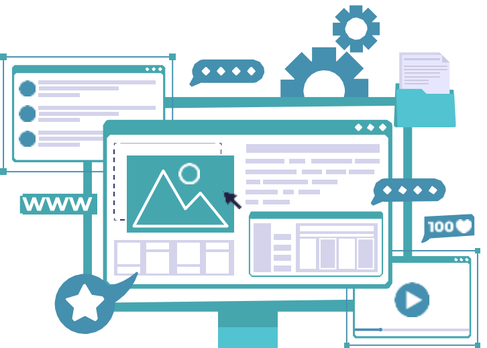


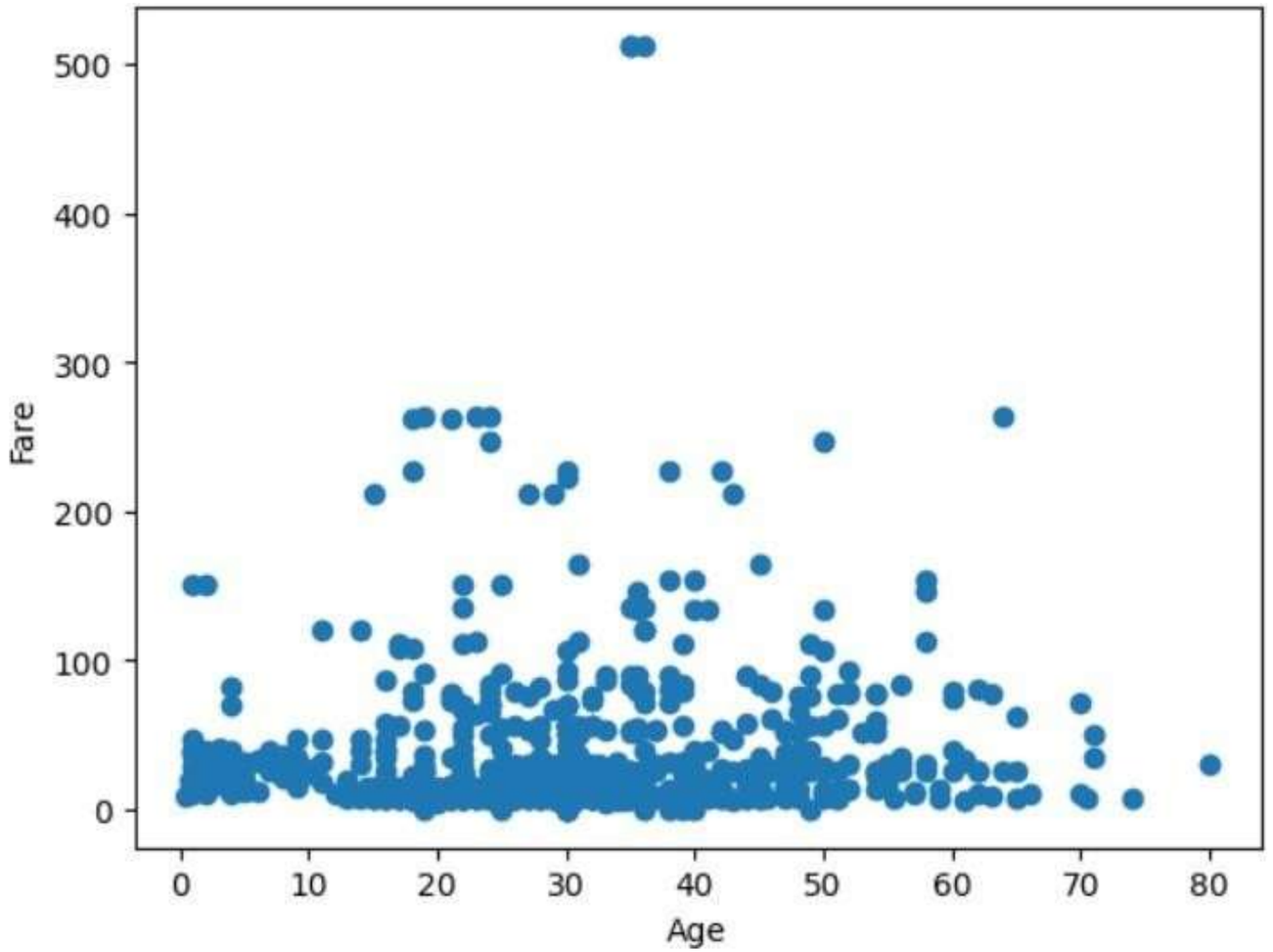

```
[99] survivor = df['Survived'].value_counts()

plt.pie(survivor.values, labels=survivor.index, autopct='%1.1f%%')
plt.title('Passenger Survival Rate')
plt.show()
```



Predictive Technique (K Means)

```
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.cluster import KMeans

df = pd.read_csv("/content/Titanic.csv")
Data = {'x': df["Age"], 'y': df["Fare"]}
df = pd.DataFrame(Data, columns=['x', 'y'])

plt.xlabel("Age")
plt.ylabel("Fare")
plt.scatter(df['x'], df['y'])

plt.show()
```





```python
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.cluster import KMeans

df = pd.read_csv("/content/Titanic.csv")
Data = {'x': df["Age"], 'y': df["Fare"]}
df = pd.DataFrame(Data, columns=['x', 'y'])

km = KMeans(n_clusters=3).fit(df)
centroids = km.cluster_centers_

plt.xlabel("Age")
plt.ylabel("Fare")
plt.scatter(df['x'], df['y'], c=km.labels_.astype(float), s=60, alpha=1)
plt.scatter(centroids[:, 0], centroids[:, 1], c='red', s=190)

plt.show()
```