Name: Satwik Shirpurwar

Roll No: 52

Batch: A3

Subject: Machine Learning Lab

Sec: A

Experiment No-1

To study and apply Data Preprocessing techniques on given dataset.

Prepare a titanic dataset for training with machine learning algorithm by applying suitable data preprocessing techniques.

Perform basic EDA using functions like head(), tail(), describe(), shape.

Identify & handle missing value.

Check for duplicates and handle it.

Identify outliers & handle it.

Apply data encoding technique on suitable attribute.

Apply Univariate, Bivariate & Multivariate analysis

Apply feature scaling technique on suitable attribute.

Split the dataset in the ratio of 80:20

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

import warnings
warnings.filterwarnings('ignore')

df = pd.read_csv('titanic.csv')
```

```
df.head()
```

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500	NaN	S
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2833	C85	C
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9250	NaN	S
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.1000	C123	S
4	5	0	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8.0500	NaN	S
5	6	0	3	Moran, Mr. James	male	NaN	0	0	330877	8.4583	NaN	Q
6	7	0	1	McCarthy, Mr. Timothy J	male	54.0	0	0	17463	51.8625	E46	S

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 891 entries, 0 to 890
Data columns (total 12 columns):
 #   Column       Non-Null Count  Dtype
---  ------       --------------  -----
 0   PassengerId  891 non-null    int64
 1   Survived     891 non-null    int64
 2   Pclass       891 non-null    int64
 3   Name         891 non-null    object
 4   Sex          891 non-null    object
 5   Age          714 non-null    float64
 6   SibSp        891 non-null    int64
 7   Parch        891 non-null    int64
 8   Ticket       891 non-null    object
 9   Fare         891 non-null    float64
 10  Cabin        204 non-null    object
 11  Embarked     889 non-null    object
dtype: float64(2), int64(5), object(5)
memory usage: 83.7 KB
```

```
df.shape
```

```
(891, 12)
```

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 891 entries, 0 to 890
Data columns (total 12 columns):
 #   Column       Non-Null Count  Dtype
---  ------       --------------  -----
 0   PassengerId  891 non-null    int64
 1   Survived     891 non-null    int64
 2   Pclass       891 non-null    int64
 3   Name         891 non-null    object
 4   Sex          891 non-null    object
 5   Age          714 non-null    float64
 6   SibSp        891 non-null    int64
 7   Parch        891 non-null    int64
 8   Ticket       891 non-null    object
 9   Fare         891 non-null    float64
 10  Cabin        204 non-null    object
 11  Embarked     889 non-null    object
dtype: float64(2), int64(5), object(5)
memory usage: 83.7 KB
```

```
df.describe()
```

	PassengerId	Survived	Pclass	Age	SibSp	Parch	Fare
count	891.000000	891.000000	891.000000	714.000000	891.000000	891.000000	891.000000
mean	446.000000	0.383838	2.308642	29.699118	0.523008	0.381594	32.204208
std	257.353842	0.486592	0.836071	14.526497	1.102743	0.806057	49.693429
min	1.000000	0.000000	1.000000	0.420000	0.000000	0.000000	0.000000
25%	223.500000	0.000000	2.000000	20.125000	0.000000	0.000000	7.910400
50%	446.000000	0.000000	3.000000	28.000000	0.000000	0.000000	14.454200
75%	668.500000	1.000000	3.000000	38.000000	1.000000	0.000000	31.000000
max	891.000000	1.000000	3.000000	80.000000	8.000000	6.000000	512.329200

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

```
PassengerId    0
Survived       0
Pclass         0
Name           0
Sex            0
Age          177
SibSp          0
Parch          0
Ticket         0
Fare           0
Cabin        687
Embarked       2
dtype: int64
```

```
df[df.duplicated()]
```

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked

```
df.drop_duplicates(inplace=True)
```

```
df.reset_index(drop=True, inplace=True)
```

```
df.shape
```

```
(891, 12)
```

```
sns.boxplot('Age', data=df)
```

```
<Axes>
```

```
df[df['Age'] > (mean)]
```

```
df[df['Age'] < (mean)]
```

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

```
PassengerId    0
Survived       0
Pclass         0
Name           0
Sex            0
Age          177
SibSp          0
Parch          0
Ticket         0
Fare           0
Cabin        687
Embarked       2
dtype: int64
```

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

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

```
PassengerId    0
Survived       0
Pclass         0
Name           0
Sex            0
Age            0
SibSp          0
Parch          0
Ticket         0
Fare           0
Cabin        687
Embarked       2
dtype: int64
```

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

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

```
PassengerId    0
Survived       0
Pclass         0
Name           0
Sex            0
Age            0
SibSp          0
Parch          0
Ticket         0
Fare           0
Cabin        687
Embarked       2
dtype: int64
```

```
df['Cabin'].fillna(mode,inplace=True)
```

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

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

```
df['Embarked'].fillna(mode,inplace=True)
```

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

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

```
df['Fare']
```

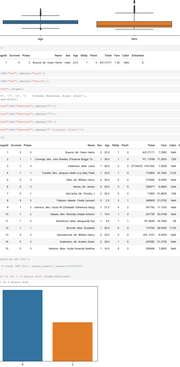
```
0      7.2500
1     71.2833
2      7.9250
3     53.1000
4      8.0500
5      8.4583
6     51.8625
7     21.0750
8     11.1333
9     30.0708
10    16.7000
11    26.5500
12    8.0500
13    31.2750
14    7.8542
15    16.0000
16    29.1250
17    13.0000
18    18.0000
19     7.2250
20    26.0000
21    13.5000
22    8.0292
23    35.5000
24    21.075
25    31.3875
26     7.8958
27    263.000
28    7.8792
29    7.8958
30    27.7208
31    146.5208
32    7.7500
33    10.5000
34    211.3375
35    13.5000
36    61.3792
37    21.075
38    30.0708
39    16.7000
40    7.7958
41    29.7000
42    7.8958
43    26.3875
44    16.1000
45    7.8958
46    35.5000
47    7.2292
48    17.8000
49    7.2292
50    26.3875
51    16.1000
52    7.8958
53    35.5000
54    227.525
55    211.5000
56    73.5000
57    10.5000
58    211.5000
59    73.5000
60    8.05000
61    39.6875
62    7.8958
63    77.2875
64    8.6542
65    7.8958
66    77.2875
67    8.6542
68    7.925
69    7.8958
70    73.5000
71    46.9000
72    7.8958
73    77.2875
74    56.4958
75    7.8958
76    7.8958
77    8.05
78    29.7000
79    7.8958
80    26.3875
81    22.3583
82    26.3875
83    7.8958
84    35.5000
85    27.7208
86    146.5208
87    7.8958
88    10.5167
89    211.5000
90    7.8958
91    8.05
92    14.4583
93    7.8958
94    113.275
95    7.8958
96    7.8958
97    134.5000
98    7.8958
99    7.8958
100   7.8958
101   7.8958
102   7.8958
103   8.05
104   15.9000
105   7.8958
106   7.75
107   23.0000
108   12.475
109   7.8958
110   211.3375
111   7.8958
112   7.2292
113   7.75
114   26.3875
115   7.8958
116   7.8958
117   26.2875
118   7.8958
119   26.3875
120   26.3875
121   262.375
122   26.3875
123   26.3875
124   26.3875
125   7.8958
126   26.3875
127   7.225
128   26.3875
129   15.5000
130   7.8958
131   7.8958
132   7.8958
133   8.6625
134   10.5000
135   7.8958
136   23.4500
137   7.8958
138   7.8958
139   8.05
140   15.5000
141   7.8958
142   77.2875
143   8.6542
144   7.8958
145   7.8958
146   73.5000
147   7.8958
148   39.6875
149   7.8958
150   77.2875
151   8.6542
152   7.8958
153   7.8958
154   8.05
155   15.9000
156   7.8958
157   7.75
158   23.0000
159   12.475
160   7.8958
161   211.3375
162   7.8958
163   7.225
164   7.75
165   26.3875
166   7.8958
167   26.3875
168   26.3875
169   262.375
170   26.3875
171   26.3875
172   26.3875
173   7.8958
174   26.3875
175   7.225
176   26.3875
177   15.5000
178   7.8958
179   7.8958
180   7.8958
181   8.05
182   15.9000
183   7.8958
184   7.75
185   23.0000
186   12.475
187   7.8958
188   211.3375
189   7.8958
190   7.225
191   7.75
192   26.3875
193   7.8958
194   26.3875
195   26.3875
196   262.375
197   26.3875
198   26.3875
199   26.3875
200   7.8958
201   26.3875
202   7.225
203   26.3875
204   15.5000
205   7.8958
206   7.8958
207   7.8958
208   8.05
209   15.9000
210   7.8958
211   7.75
212   23.0000
213   12.475
214   7.8958
215   211.3375
216   7.8958
217   7.225
218   7.75
219   26.3875
220   7.8958
221   26.3875
222   26.3875
223   262.375
224   26.3875
225   26.3875
226   26.3875
227   7.8958
228   26.3875
229   7.225
230   26.3875
231   15.5000
232   7.8958
233   7.8958
234   7.8958
235   8.05
236   15.9000
237   7.8958
238   7.75
239   23.0000
240   12.475
241   7.8958
242   211.3375
243   7.8958
244   7.225
245   7.75
246   26.3875
247   7.8958
248   26.3875
249   26.3875
250   262.375
251   26.3875
252   26.3875
253   26.3875
254   7.8958
255   26.3875
256   7.225
257   26.3875
258   15.5000
259   7.8958
260   7.8958
261   7.8958
262   8.05
263   15.9000
264   7.8958
265   7.75
266   23.0000
267   12.475
268   7.8958
269   211.3375
270   7.8958
271   7.225
272   7.75
273   26.3875
274   7.8958
275   26.3875
276   26.3875
277   262.375
278   26.3875
279   26.3875
280   26.3875
281   7.8958
282   26.3875
283   7.225
284   26.3875
285   15.5000
286   7.8958
287   7.8958
288   7.8958
289   8.05
290   15.9000
291   7.8958
292   7.75
293   23.0000
294   12.475
295   7.8958
296   211.3375
297   7.8958
298   7.225
299   7.75
300   26.3875
301   7.8958
302   26.3875
303   26.3875
304   262.375
305   26.3875
306   26.3875
307   26.3875
308   7.8958
309   26.3875
310   7.225
311   26.3875
312   15.5000
313   7.8958
314   7.8958
315   7.8958
316   8.05
317   15.9000
318   7.8958
319   7.75
320   23.0000
321   12.475
322   7.8958
323   211.3375
324   7.8958
325   7.225
326   7.75
327   26.3875
328   7.8958
329   26.3875
330   26.3875
331   262.375
332   26.3875
333   26.3875
334   26.3875
335   7.8958
336   26.3875
337   7.225
338   26.3875
339   15.5000
340   7.8958
341   7.8958
342   7.8958
343   8.05
344   15.9000
345   7.8958
346   7.75
347   23.0000
348   12.475
349   7.8958
350   211.3375
351   7.8958
352   7.225
353   7.75
354   26.3875
355   7.8958
356   26.3875
357   26.3875
358   262.375
359   26.3875
360   26.3875
361   26.3875
362   7.8958
363   26.3875
364   7.225
365   26.3875
366   15.5000
367   7.8958
368   7.8958
369   7.8958
370   8.05
371   15.9000
372   7.8958
373   7.75
374   23.0000
375   12.475
376   7.8958
377   211.3375
378   7.8958
379   7.225
380   7.75
381   26.3875
382   7.8958
383   26.3875
384   26.3875
385   262.375
386   26.3875
387   26.3875
388   26.3875
389   7.8958
390   26.3875
391   7.225
392   26.3875
393   15.5000
394   7.8958
395   7.8958
396   7.8958
397   8.05
398   15.9000
399   7.8958
400   7.75
401   23.0000
402   12.475
403   7.8958
404   211.3375
405   7.8958
406   7.225
407   7.75
408   26.3875
409   7.8958
410   26.3875
411   26.3875
412   262.375
413   26.3875
414   26.3875
415   26.3875
416   7.8958
417   26.3875
418   7.225
419   26.3875
420   15.5000
421   7.8958
422   7.8958
423   7.8958
424   8.05
425   15.9000
426   7.8958
427   7.75
428   23.0000
429   12.475
430   7.8958
431   211.3375
432   7.8958
433   7.225
434   7.75
435   26.3875
436   7.8958
437   26.3875
438   26.3875
439   262.375
440   26.3875
441   26.3875
442   26.3875
443   7.8958
444   26.3875
445   7.225
446   26.3875
447   15.5000
448   7.8958
449   7.8958
450   7.8958
451   8.05
452   15.9000
453   7.8958
454   7.75
455   23.0000
456   12.475
457   7.8958
458   211.3375
459   7.8958
460   7.225
461   7.75
462   26.3875
463   7.8958
464   26.3875
465   26.3875
466   262.375
467   26.3875
468   26.3875
469   26.3875
470   7.8958
471   26.3875
472   7.225
473   26.3875
474   15.5000
475   7.8958
476   7.8958
477   7.8958
478   8.05
479   15.9000
480   7.8958
481   7.75
482   23.0000
483   12.475
484   7.8958
485   211.3375
486   7.8958
487   7.225
488   7.75
489   26.3875
490   7.8958
491   26.3875
492   26.3875
493   262.375
494   26.3875
495   26.3875
496   26.3875
497   7.8958
498   26.3875
499   7.225
500   26.3875
501   15.5000
502   7.8958
503   7.8958
504   7.8958
505   8.05
506   15.9000
507   7.8958
508   7.75
509   23.0000
510   12.475
511   7.8958
512   211.3375
513   7.8958
514   7.225
515   7.75
516   26.3875
517   7.8958
518   26.3875
519   26.3875
520   262.375
521   26.3875
522   26.3875
523   26.3875
524   7.8958
525   26.3875
526   7.225
527   26.3875
528   15.5000
529   7.8958
530   7.8958
531   7.8958
532   8.05
533   15.9000
534   7.8958
535   7.75
536   23.0000
537   12.475
538   7.8958
539   211.3375
540   7.8958
541   7.225
542   7.75
543   26.3875
544   7.8958
545   26.3875
546   26.3875
547   262.375
548   26.3875
549   26.3875
550   26.3875
551   7.8958
552   26.3875
553   7.225
554   26.3875
555   15.5000
556   7.8958
557   7.8958
558   7.8958
559   8.05
560   15.9000
561   7.8958
562   7.75
563   23.0000
564   12.475
565   7.8958
566   211.3375
567   7.8958
568   7.225
569   7.75
570   26.3875
571   7.8958
572   26.3875
573   26.3875
574   262.375
575   26.3875
576   26.3875
577   26.3875
578   7.8958
579   26.3875
580   7.225
581   26.3875
582   15.5000
583   7.8958
584   7.8958
585   7.8958
586   8.05
587   15.9000
588   7.8958
589   7.75
590   23.0000
591   12.475
592   7.8958
593   211.3375
594   7.8958
595   7.225
596   7.75
597   26.3875
598   7.8958
599   26.3875
600   26.3875
601   262.375
602   26.3875
603   26.3875
604   26.3875
605   7.8958
606   26.3875
607   7.225
608   26.3875
609   15.5000
610   7.8958
611   7.8958
612   7.8958
613   8.05
614   15.9000
615   7.8958
616   7.75
617   23.0000
618   12.475
619   7.8958
620   211.3375
621   7.8958
622   7.225
623   7.75
624   26.3875
625   7.8958
626   26.3875
627   26.3875
628   262.375
629   26.3875
630   26.3875
631   26.3875
632   7.8958
633   26.3875
634   7.225
635   26.3875
636   15.5000
637   7.8958
638   7.8958
639   7.8958
640   8.05
641   15.9000
642   7.8958
643   7.75
644   23.0000
645   12.475
646   7.8958
647   211.3375
648   7.8958
649   7.225
650   7.75
651   26.3875
652   7.8958
653   26.3875
654   26.3875
655   262.375
656   26.3875
657   26.3875
658   26.3875
659   7.8958
660   26.3875
661   7.225
662   26.3875
663   15.5000
664   7.8958
665   7.8958
666   7.8958
667   8.05
668   15.9000
669   7.8958
670   7.75
671   23.0000
672   12.475
673   7.8958
674   211.3375
675   7.8958
676   7.225
677   7.75
678   26.3875
679   7.8958
680   26.3875
681   26.3875
682   262.375
683   26.3875
684   26.3875
685   26.3875
686   7.8958
687   26.3875
688   7.225
689   26.3875
690   15.5000
691   7.8958
692   7.8958
693   7.8958
694   8.05
695   15.9000
696   7.8958
697   7.75
698   23.0000
699   12.475
700   7.8958
701   211.3375
702   7.8958
703   7.225
704   7.75
705   26.3875
706   7.8958
707   26.3875
708   26.3875
709   262.375
710   26.3875
711   26.3875
712   26.3875
713   7.8958
714   26.3875
715   7.225
716   26.3875
717   15.5000
718   7.8958
719   7.8958
720   7.8958
721   8.05
722   15.9000
723   7.8958
724   7.75
725   23.0000
726   12.475
727   7.8958
728   211.3375
729   7.8958
730   7.225
731   7.75
732   26.3875
733   7.8958
734   26.3875
735   26.3875
736   262.375
737   26.3875
738   26.3875
739   26.3875
740   7.8958
741   26.3875
742   7.225
743   26.3875
744   15.5000
745   7.8958
746   7.8958
747   7.8958
748   8.05
749   15.9000
750   7.8958
751   7.75
752   23.0000
753   12.475
754   7.8958
755   211.3375
756   7.8958
757   7.225
758   7.75
759   26.3875
760   7.8958
761   26.3875
762   26.3875
763   262.375
764   26.3875
765   26.3875
766   26.3875
767   7.8958
768   26.3875
769   7.225
770   26.3875
771   15.5000
772   7.8958
773   7.8958
774   7.8958
775   8.05
776   15.9000
777   7.8958
778   7.75
779   23.0000
780   12.475
781   7.8958
782   211.3375
783   7.8958
784   7.225
785   7.75
786   26.3875
787   7.8958
788   26.3875
789   26.3875
790   262.375
791   26.3875
792   26.3875
793   26.3875
794   7.8958
795   26.3875
796   7.225
797   26.3875
798   15.5000
799   7.8958
800   7.8958
801   7.8958
802   8.05
803   15.9000
804   7.8958
805   7.75
806   23.0000
807   12.475
808   7.8958
809   211.3375
810   7.8958
811   7.225
812   7.75
813   26.3875
814   7.8958
815   26.3875
816   26.3875
817   262.375
818   26.3875
819   26.3875
820   26.3875
821   7.8958
822   26.3875
823   7.225
824   26.3875
825   15.5000
826   7.8958
827   7.8958
828   7.8958
829   8.05
830   15.9000
831   7.8958
832   7.75
833   23.0000
834   12.475
835   7.8958
836   211.3375
837   7.8958
838   7.225
839   7.75
840   26.3875
841   7.8958
842   26.3875
843   26.3875
844   262.375
845   26.3875
846   26.3875
847   26.3875
848   7.8958
849   26.3875
850   7.225
851   26.3875
852   15.5000
853   7.8958
854   7.8958
855   7.8958
856   8.05
857   15.9000
858   7.8958
859   7.75
860   23.0000
861   12.475
862   7.8958
863   211.3375
864   7.8958
865   7.225
866   7.75
867   26.3875
868   7.8958
869   26.3875
870   26.3875
871   262.375
872   26.3875
873   26.3875
874   26.3875
875   7.8958
876   26.3875
877   7.225
878   26.3875
879   15.5000
880   7.8958
881   7.8958
882   7.8958
883   8.05
884   15.9000
885   7.8958
886   7.75
887   23.0000
888   12.475
889   7.8958
890   211.3375
```

```
sns.countplot('Pclass')
```

```
<Axes>
```

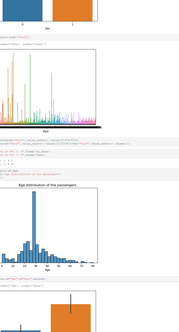
```
sns.countplot('Sex')
```

```
<Axes>
```




```
sns.countplot('Survived')
```

```
<Axes>
```




```
sns.countplot('Embarked')
```

```
<Axes>
```

```
sns.countplot('SibSp')
```

```
<Axes>
```

```
sns.countplot('Parch')
```

```
<Axes>
```



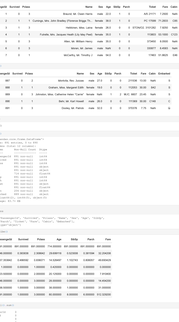
```
import matplotlib.pyplot as plt
import seaborn as sns
import numpy as np
%matplotlib inline
df.hist(bins=50,figsize=(20,15))
plt.show()
```

```
array([[<AxesSubplot:>, <AxesSubplot:>, <AxesSubplot:>,
        <AxesSubplot:>],
       [<AxesSubplot:>, <AxesSubplot:>, <AxesSubplot:>,
        <AxesSubplot:>],
       [<AxesSubplot:>, <AxesSubplot:>, <AxesSubplot:>,
        <AxesSubplot:>],
       [<AxesSubplot:>, <AxesSubplot:>, <AxesSubplot:>,
        <AxesSubplot:>]]
```



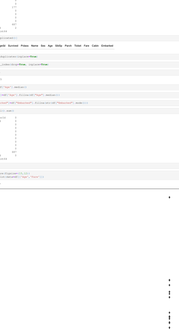
```
sns.boxplot('Pclass','Fare',data=df)
```

```
<Axes>
```




```
sns.boxplot('Sex','Fare',data=df)
```

```
<Axes>
```




```
sns.boxplot('Age','Fare',data=df)
```

```
<Axes>
```



```
sns.boxplot('Survived','Fare',data=df)
```

```
<Axes>
```



```
import matplotlib.pyplot as plt
import seaborn as sns
import numpy as np
%matplotlib inline
df.corr(method='pearson',min_periods=1)
```

	PassengerId	Survived	Pclass	Age	SibSp	Parch	Fare
PassengerId	1.000000	-0.005007	-0.035144	0.036847	-0.057527	-0.001652	0.012658
Survived	-0.005007	1.000000	-0.338481	-0.077221	-0.035322	0.081629	0.257307
Pclass	-0.035144	-0.338481	1.000000	-0.369226	0.083081	0.018443	-0.549500
Age	0.036847	-0.077221	-0.369226	1.000000	-0.308247	-0.189119	0.096067
SibSp	-0.057527	-0.035322	0.083081	-0.308247	1.000000	0.414838	0.159651
Parch	-0.001652	0.081629	0.018443	-0.189119	0.414838	1.000000	0.216225
Fare	0.012658	0.257307	-0.549500	0.096067	0.159651	0.216225	1.000000

```
sns.pairplot(df,hue='Survived',palette='Spectral')
```

```
<Seaborn>
```



```
sns.heatmap(df.corr(),annot=True,cmap='Blues')
```

```
<Axes>
```



```
from sklearn.model_selection import train_test_split
```

```
Sex=pd.get_dummies(df['Sex'],drop_first=True)
Embarked=pd.get_dummies(df['Embarked'],drop_first=True)
df=df.drop(['Sex','Embarked'],axis=1)
df=pd.concat([df,Sex,Embarked],axis=1)
df.drop(['PassengerId','Name','Ticket'],inplace=True)
```

```
df.head()
```

	Survived	Pclass	Age	SibSp	Parch	Fare	Cabin	male	Q	S
0	0	3	22.0	1	0	7.2500	NaN	1	0	1
1	1	1	38.0	1	0	71.2833	C85	0	0	0
2	1	3	26.0	0	0	7.9250	NaN	0	0	1
3	1	1	35.0	1	0	53.1000	C123	0	0	1
4	0	3	35.0	0	0	8.0500	NaN	1	0	1