

Pandas `get_dummies()` is used to convert categorical variables into dummy variables. Each category is transformed into a new column with binary value (1 or 0) indicating the presence of the category in the original data

```
import pandas as pd
```

```
# sample data
```

```
data = {'Color': ['Red', 'Green', 'Blue', 'Green', 'Red']}
```

```
# creating a DataFrame
```

```
df = pd.DataFrame(data)
```

```
print(df)
```

```
↗
  Color
0    Red
1  Green
2    Blue
3  Green
4    Red
```

```
# using get_dummies to convert the categorical column
```

```
d1 = pd.get_dummies(df['Color'])
```

```
print(d1)
```

```
↗
   Blue  Green   Red
0  False  False  True
1  False   True  False
2   True  False  False
3  False   True  False
4  False  False  True
```

```
# using get_dummies to convert the categorical column to float type
```

```
d2 = pd.get_dummies(df['Color'],dtype=float)
```

```
print(d2)
```

```
↗
   Blue  Green  Red
0  0.0    0.0  1.0
1  0.0    1.0  0.0
2  1.0    0.0  0.0
3  0.0    1.0  0.0
4  0.0    0.0  1.0
```

```
# using get_dummies to convert the categorical column to 1/0
```

```
d3 = pd.get_dummies(df['Color'],dtype=int)
```

```
print(d3)
```

```
↗
   Blue  Green  Red
0     0     0    1
1     0     1    0
2     1     0    0
3     0     1    0
4     0     0    1
```

```
# concatenating the dummies DataFrame with the original DataFrame
```

```
df = pd.concat([df, d3], axis=1)
```

```
print(df)
```

```
↗
  Color  Blue  Green  Red
0    Red     0     0    1
1  Green     0     1    0
2   Blue     1     0    0
3  Green     0     1    0
4    Red     0     0    1
```

```
# using get_dummies to convert the categorical column to 1/0
```

```
d3 = pd.get_dummies(df['Color'],dtype=int)
```

```
print(d3)
```

```
↗
   Blue  Green  Red
0     0     0    1
1     0     1    0
2     1     0    0
3     0     1    0
4     0     0    1
```

```
#drop first coumn using drop_first
```

```
# using get_dummies to convert the categorical column to 1/0
```

```
d3 = pd.get_dummies(df['Color'],dtype=int,drop_first=1)
```

```
print(d3)
```

```

→
   Green  Red
0       0    1
1       1    0
2       0    0
3       1    0
4       0    1

```

a. Determine the categorical columns in Titanic Dataset. Convert Columns with string data type to numerical data using encoding techniques.

```

# importing all the necessary libraries
import pandas as pd
import numpy as np
#we need to read the data
df=pd.read_csv("/content/drive/MyDrive/AI Tools Lab/nonnull_titanic.csv")
#print top 5 rows
df.isnull().mean()

```

```

→ PassengerId    0.0
   Survived      0.0
   Pclass        0.0
   Name          0.0
   Sex           0.0
   Age           0.0
   SibSp         0.0
   Parch         0.0
   Ticket        0.0
   Fare          0.0
   Embarked      0.0
dtype: float64

```

```
df.info()
```

```

→ <class 'pandas.core.frame.DataFrame'>
RangeIndex: 880 entries, 0 to 879
Data columns (total 11 columns):
 #   Column          Non-Null Count  Dtype
---  -
0   PassengerId     880 non-null    int64
1   Survived        880 non-null    int64
2   Pclass          880 non-null    int64
3   Name            880 non-null    object
4   Sex             880 non-null    object
5   Age             880 non-null    float64
6   SibSp           880 non-null    int64
7   Parch           880 non-null    int64
8   Ticket          880 non-null    object
9   Fare            880 non-null    float64
10  Embarked        880 non-null    object
dtypes: float64(2), int64(5), object(4)
memory usage: 75.8+ KB

```

```

print("each unique value and respective counts in Sex column\n",df['Sex'].value_counts())
#creating another data frame using get_dummies
sex_df = pd.get_dummies(df['Sex'])
sex_df.head()

```

```

→ each unique value and respective counts in Sex column
Sex
male      572
female    308
Name: count, dtype: int64

   female  male
0  False   True
1  False   True
2  False   True
3  False   True
4  False   True

```

```


#creating another data frame for Sex column by dropping first column in get dummies
sex_df = pd.get_dummies(df['Sex'],drop_first=True,dtype=int)
sex_df.head()

```



	male
0	1
1	1
2	1
3	1
4	1


```
print("each unique value and respective counts in Sex column\n",df['Embarked'].value_counts())
# creating dummies for Embarked
embark_df = pd.get_dummies(df['Embarked'],drop_first=True,dtype=int)
embark_df.head()
```



```
each unique value and respective counts in Sex column
Embarked
S    642
C    161
Q     77
Name: count, dtype: int64
```


	Q	S
0	0	1
1	0	1
2	0	0
3	0	0
4	1	0

```
old_data = df.copy()
# we need to drop the sex and embarked columns and replace them with the newly created dummies data frames
# as Name and Ticket is not making any impact on the output label, we can drop them also
df.drop(['Sex','PassengerId','Embarked','Name','Ticket'],axis=1,inplace=True)
df.head()
```



	Survived	Pclass	Age	SibSp	Parch	Fare
0	1	1	80.0	0	0	30.0000
1	0	3	74.0	0	0	7.7750
2	0	1	71.0	0	0	34.6542
3	0	1	71.0	0	0	49.5042
4	0	3	70.5	0	0	7.7500


```
# After dropping the Sex and Embarked columns, we are replacing them with out new data frames
data = pd.concat([df,sex_df,embark_df],axis=1)
data.head()
```



	Survived	Pclass	Age	SibSp	Parch	Fare	male	Q	S
0	1	1	80.0	0	0	30.0000	1	0	1
1	0	3	74.0	0	0	7.7750	1	0	1
2	0	1	71.0	0	0	34.6542	1	0	0
3	0	1	71.0	0	0	49.5042	1	0	0
4	0	3	70.5	0	0	7.7500	1	1	0


b. Convert data in each numerical column so that it lies in the range [0,1]

```
# Scaling the data using minmax scaler so that values should be lies btw [0,1]
from sklearn.preprocessing import MinMaxScaler
scaler = MinMaxScaler()
data[['Age','Pclass','Survived','SibSp','Parch','Fare','male','Q','S']] = scaler.fit_transform(data[['Age','Pclass','Survived','SibSp',
# after scaling the data
data.head()
```



	Survived	Pclass	Age	SibSp	Parch	Fare	male	Q	S
0	1.0	0.0	1.000000	0.0	0.0	0.131854	1.0	0.0	1.0
1	0.0	1.0	0.924604	0.0	0.0	0.034172	1.0	0.0	1.0
2	0.0	0.0	0.886906	0.0	0.0	0.152309	1.0	0.0	0.0
3	0.0	0.0	0.886906	0.0	0.0	0.217577	1.0	0.0	0.0
4	0.0	1.0	0.880623	0.0	0.0	0.034062	1.0	1.0	0.0

```
data.info()
```



```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 880 entries, 0 to 879
Data columns (total 9 columns):
#   Column      Non-Null Count  Dtype
---  ---
0    Survived    880 non-null    float64
1    Pclass      880 non-null    float64
2    Age         880 non-null    float64
3    SibSp       880 non-null    float64
4    Parch       880 non-null    float64
5    Fare        880 non-null    float64
6    male        880 non-null    float64
7    Q           880 non-null    float64
8    S           880 non-null    float64
dtypes: float64(9)
memory usage: 62.0 KB
```

```
data.to_csv("/content/drive/MyDrive/AI Tools Lab/titanic6.csv")
```

Start coding or [generate](#) with AI.