

-Titanic-Machine-Learning-from-Disaster-

Kaggle Notebook Link:

<https://www.kaggle.com/tejashrikelhe1/kernel3d93c1995d?scriptVersionId=31141631>

CODE

```
import math

import numpy as np # linear algebra

import pandas as pd # data processing, CSV file I/O (e.g. pd.read_csv)

import os

for dirname, _, filenames in os.walk('/kaggle/input'):
    for filename in filenames:
        print(os.path.join(dirname, filename))

train_df=pd.read_csv("/kaggle/input/titanic/train.csv")
train_df.head()

test_df=pd.read_csv("/kaggle/input/titanic/test.csv")
test_df.head()

combine=[train_df, test_df]

train_df

print(train_df.columns.values)

train_df.describe(percentiles=None, include=None, exclude=None)

train_df.describe(percentiles=None, include='O', exclude=None)

print('Percent of missing records in AGE is %.2f%%'
      %((train_df['Age'].isnull().sum()/train_df.shape[0])*100))

print('Percent of missing records in EMBARKED is %.2f%%'
      %((train_df['Embarked'].isnull().sum()/train_df.shape[0])*100))

print('Percent of missing records in CABIN is %.2f%%'
      %((train_df['Cabin'].isnull().sum()/train_df.shape[0])*100))

age_mean=train_df["Age"].mean()

print(age_mean)

agetest_mean=test_df["Age"].mean()

print(agetest_mean)
```

```

num_rows=train_df.shape[0]
print(num_rows)
numtest_rows=test_df.shape[0]
print(numtest_rows)
for i in range(num_rows):
    if math.isnan(train_df.Age[i]):
        train_df.Age[i]=age_mean
for i in range(numtest_rows):
    if math.isnan(test_df.Age[i]):
        test_df.Age[i]=age_mean
train_df
test_df
train_df["Embarked"].fillna(train_df["Embarked"].value_counts().idxmax(), inplace=True)
train_df
test_df["Embarked"].fillna(test_df["Embarked"].value_counts().idxmax(), inplace=True)
test_df
train_df.drop(['Cabin'], axis=1, inplace=True)
test_df.drop(['Cabin'], axis=1, inplace=True)
test_df.isnull().sum()
train_df.isnull().sum()
train_df.head()
train_df = pd.get_dummies(train_df,columns=['Sex','Pclass'])
train_df.drop(['Sex_male'], axis=1, inplace=True)
train_df.drop(['Sex_female'], axis=1, inplace=True)
train_df
train_df.head()
train_df.isnull().sum()
test_df = pd.get_dummies(test_df,columns=['Sex','Pclass'])
test_df.drop(['Sex_male'], axis=1, inplace=True)
test_df.drop(['Sex_female'], axis=1, inplace=True)
test_df.head()

```

```

train_df = pd.get_dummies(train_df,columns=['Embarked'])
train_df.head()

test_df = pd.get_dummies(test_df,columns=['Embarked'])
test_df.head()

test_df.drop(['Name', 'Ticket'], axis=1, inplace=True)

train_df['TravelAlone']=np.where((train_df["SibSp"]+train_df["Parch"])>0, 0, 1)
train_df.drop(['SibSp', 'Parch', 'Name', 'Ticket'], axis=1, inplace=True)
train_df.head()

test_df['TravelAlone']=np.where((test_df["SibSp"]+test_df["Parch"])>0, 0, 1)
test_df.drop(['SibSp', 'Parch'], axis=1, inplace=True)
test_df.head()

train_df.isnull().sum()
test_df.isnull().sum()

from sklearn.model_selection import train_test_split

X_train, X_test, y_train, y_test = train_test_split(train_df.drop('Survived',axis=1),
train_df['Survived'], test_size=0.2, random_state=101)

from sklearn.linear_model import LogisticRegression

log_model = LogisticRegression()

log_model.fit(X_train, y_train)

predict = log_model.predict(X_test)

print(predict)

from sklearn.metrics import classification_report

print(classification_report(y_test,predict))

```