```
"""Name: Muhammad Hashir
Date: 09/90/25"""
```

```
'Name: Muhammad Hashir\nDate: 09/90/25'
```

```
import pandas as pd
df = pd.read_csv("train.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	female	35.0	1	0	113803	53.1000	C123	S

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

mode_embarked = df['Embarked'].mode()[0]
df['Embarked'].fillna(mode_embarked, inplace=True)

train_mean_age = mean_age
train_mode_embarked = mode_embarked

df.drop(columns=['Name', 'Ticket', 'Cabin'], inplace=True, errors='ignore')
y = df['Survived']
X = df.drop(columns=['Survived', 'PassengerId'])
```

```
/var/folders/m3/8kjpfvdd6td95pxz_k210f6m0000gn/T/ipykernel_72231/4267558607.py:2: FutureWarning: A value is trying to be
The behavior will change in pandas 3.0. This inplace method will never work because the intermediate object on which we

For example, when doing 'df[col].method(value, inplace=True)', try using 'df.method({col: value}, inplace=True)' or df[c


  df['Age'].fillna(mean_age, inplace=True)
/var/folders/m3/8kjpfvdd6td95pxz_k210f6m0000gn/T/ipykernel_72231/4267558607.py:6: FutureWarning: A value is trying to be
The behavior will change in pandas 3.0. This inplace method will never work because the intermediate object on which we

For example, when doing 'df[col].method(value, inplace=True)', try using 'df.method({col: value}, inplace=True)' or df[c


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

```
realcols = ["Age","Pclass","SibSp", "Parch", "Fare"]
train_scaling_stats = {}

for col in realcols:
  mean = X[col].mean()
  std = X[col].std()

  train_scaling_stats[col] = {'mean': mean, 'std': std}

  X[col] = (X[col] - mean) / std

traincol = X.columns.tolist()
```

```
from sklearn.linear_model import LogisticRegression
import numpy as np
from sklearn.model_selection import train_test_split

X = pd.get_dummies(X, columns=['Sex', 'Embarked'], drop_first=True)

train_cols_final = X.columns.tolist()

X = X.fillna(X.mean())


X_train, y_train = X, y
```

```
logisticReg = LogisticRegression()
logisticReg.fit(X_train, y_train)

print(f"Model trained successfully with {X_train.shape[0]} samples and {X_train.shape[1]} features.")
```

```
Model trained successfully with 891 samples and 8 features.
```

```
df_test = pd.read_csv('test.csv')
test_ids = df_test['PassengerId']

df_test.drop(columns=['Name', 'Ticket', 'Cabin'], inplace=True, errors="ignore")

df_test['Age'].fillna(train_mean_age, inplace=True)
df_test['Embarked'].fillna(train_mode_embarked, inplace=True)

train_fare_mean = train_scaling_stats['Fare']['mean']
df_test['Fare'].fillna(train_fare_mean, inplace=True)


X_test = df_test.drop(columns=['PassengerId'])

realcols = ["Age","Pclass","SibSp", "Parch", "Fare"]
for col in realcols:
  mean = train_scaling_stats[col]['mean']
  std = train_scaling_stats[col]['std']


  X_test[col] = (X_test[col] - mean) / std if std != 0 else X_test[col] - mean

X_test = pd.get_dummies(X_test, columns=['Sex', 'Embarked'], drop_first=True)

X_test = X_test.reindex(columns=train_cols_final, fill_value=0)
```

```
/var/folders/m3/8kjpfvdd6td95pxz_k210f6m0000gn/T/ipykernel_72231/3789877998.py:8: FutureWarning: A value is trying to be
The behavior will change in pandas 3.0. This inplace method will never work because the intermediate object on which we

For example, when doing 'df[col].method(value, inplace=True)', try using 'df.method({col: value}, inplace=True)' or df[c


  df_test['Age'].fillna(train_mean_age, inplace=True)
/var/folders/m3/8kjpfvdd6td95pxz_k210f6m0000gn/T/ipykernel_72231/3789877998.py:9: FutureWarning: A value is trying to be
The behavior will change in pandas 3.0. This inplace method will never work because the intermediate object on which we

For example, when doing 'df[col].method(value, inplace=True)', try using 'df.method({col: value}, inplace=True)' or df[c


  df_test['Embarked'].fillna(train_mode_embarked, inplace=True)
/var/folders/m3/8kjpfvdd6td95pxz_k210f6m0000gn/T/ipykernel_72231/3789877998.py:14: FutureWarning: A value is trying to b
The behavior will change in pandas 3.0. This inplace method will never work because the intermediate object on which we

For example, when doing 'df[col].method(value, inplace=True)', try using 'df.method({col: value}, inplace=True)' or df[c


  df_test['Fare'].fillna(train_fare_mean, inplace=True)
```

```
y_pred_survived = logisticReg.predict(X_test)
submission_df = pd.DataFrame({
    'PassengerId': test_ids,
    'Survived': y_pred_survived.astype(int)
})

print("Submission Data Format:")
print(submission_df.head())

submission_df.to_csv('submission_predictions.csv', index=False)
```

```
Submission Data Format:
   PassengerId  Survived
0          892         0
1          893         0
2          894         0
3          895         0
4          896         1
```

Start coding or generate with AI.