# Titanic Classification Machine Learning Tutorial

## Introduction

This tutorial guides you through a complete machine learning workflow to predict Titanic passenger survival using Python and scikit-learn. The process includes data loading, cleaning, preprocessing, model training, evaluation, and prediction.

## Importing Essential Libraries

import pandas as pd import seaborn as sns

import matplotlib.pyplot as plt

from sklearn.model\_selection import train\_test\_split from sklearn.ensemble import RandomForestClassifier

from sklearn.metrics import accuracy\_score, classification\_report

 **pandas**: Data manipulation

 **seaborn** and **matplotlib**: Visualization

 **scikit-learn**: Machine learning models and evaluation

## Loading the Dataset

data = pd.read\_csv(r"C:\Users\LENOVO T450S\OneDrive\Desktop\titanic.csv")

*Update the file path to where your Titanic CSV is located.*

## Data Cleaning and Preparation

data.columns = data.columns.str.strip()

if 'Age' in data.columns:

data['Age'] = data['Age'].fillna(data['Age'].median())

if 'Embarked' in data.columns:

data['Embarked'] = data['Embarked'].fillna(data['Embarked'].mode()[^0])

for col in ['Name', 'Ticket', 'Cabin']: if col in data.columns:

data.drop(col, axis=1, inplace=True)

categorical\_cols = []

if 'Sex' in data.columns: categorical\_cols.append('Sex')

if 'Embarked' in data.columns:

categorical\_cols.append('Embarked')

if categorical\_cols:

data = pd.get\_dummies(data, columns=categorical\_cols, drop\_first=True)

### Steps:

 Strips whitespace from column names  Fills missing Age with the median

 Fills missing Embarked with the mode  Drops Name, Ticket, Cabin columns

 Encodes categorical columns into numeric variables

## Splitting the Dataset

X = data.drop("Survived", axis=1) y = data["Survived"]

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=42)

 20% of the data is for testing, 80% for training.

## Model Training

The model learns from the training data to predict survival.

model = RandomForestClassifier() model.fit(X\_train, y\_train) y\_pred = model.predict(X\_test)

## Model Evaluation

print("Accuracy:", accuracy\_score(y\_test, y\_pred)) print(classification\_report(y\_test, y\_pred))

### Sample Output:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | precision | recall | f1-score | support |
| 0 | 0.83 | 0.88 | 0.85 | 105 |
| 1 | 0.81 | 0.74 | 0.77 | 74 |
| accuracy |  |  | 0.82 | 179 |
| macro avg | 0.82 | 0.81 | 0.81 | 179 |
| weighted avg | 0.82 | 0.82 | 0.82 | 179 |

 **Accuracy**: Proportion of correct predictions

Accuracy: 0.82

 **Precision, Recall, F1-score**: Detailed metrics for each class

## Making Predictions

y\_pred = model.predict(X\_test) print(y\_pred)

 Outputs an array of 0s and 1s 0 = did not survive, 1 = survived).

## Summary Table

|  |  |
| --- | --- |
| Step | Description |
| Import Libraries | Load Python packages for ML, data, and visualization |
| Load Data | Read Titanic CSV into DataFrame |
| Clean/Prepare Data | Handle missing values, drop unused columns, encode categories |
| Split Data | Separate features/target, create train/test sets |
| Train Model | Fit RandomForestClassifier on training data |
| Evaluate Model | Print accuracy and classification report |
| Predict Outcomes | Output predictions for test set |

1. **How to Execute**

### Install Required Libraries

pip install pandas scikit-learn matplotlib seaborn

**Prepare Your Data**

 Download the Titanic dataset CSV) and update the file path in the code.

### Run the Notebook

 Open the .ipynb file in Jupyter Notebook or compatible editor.  Execute each cell in order.

### Interpret Results

 Review the printed accuracy and classification report.

### End of Document

Copy and paste this structure into Microsoft Word. You can further enhance it with headings, bullet points, and code formatting as needed for clarity and presentation 1  .

⁂

Titanic-Classification.ipynb