```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score, classification_report
import pickle

# 1. Load the dataset
#url = "https://docs.google.com/spreadsheets/d/1htBZNAk96XHc7V28PD1thdqq8x1wZhqOtP5IyJzxDRM/export?format=csv"
titanic_df = pd.read_csv("/content/titanic.csv")

# 2. Descriptive Analysis
print(titanic_df.info())
print(titanic_df.describe())
print(titanic_df.head())

# 3. Missing Value Handling
titanic_df.dropna(inplace=True)

# 4. Irrelevant Data Handling
# Drop irrelevant columns
titanic_df.drop(['PassengerId', 'Name', 'Ticket'], axis=1, inplace=True)

# 5. Feature Scaling
scaler = StandardScaler()
titanic_df[['Age', 'Fare']] = scaler.fit_transform(titanic_df[['Age', 'Fare']])

# 6. Categorical Encoding
titanic_df = pd.get_dummies(titanic_df, columns=['Sex', 'Embarked'], drop_first=True)

# 7. Split the data into train and test sets
X = titanic_df.drop('survived',axis=1)
y = titanic_df['survived']
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# 8. Choose a classification algorithm
model = RandomForestClassifier()

# 9. Train the model
model.fit(X_train, y_train)

# 10. Evaluation Metrics
y_pred = model.predict(X_test)
accuracy = accuracy_score(y_test, y_pred)
report = classification_report(y_test, y_pred)

print("Accuracy:", accuracy)
print("Classification Report:\n", report)

# 11. Save Model
with open('titanic_model.pkl', 'wb') as f:
    pickle.dump(model, f)
```