Eda trial mid term

1. Summary Statistics on Numeric Variables

import pandas as pd

import numpy as np

import seaborn as sns

import matplotlib.pyplot as plt

from sklearn.model\_selection import train\_test\_split, GridSearchCV

from sklearn.preprocessing import StandardScaler, OneHotEncoder

from sklearn.impute import SimpleImputer

from sklearn.pipeline import Pipeline

from sklearn.compose import ColumnTransformer

from sklearn.ensemble import RandomForestClassifier

from sklearn.metrics import roc\_auc\_score

1. Read Dataset

import pandas as pd

df = pd.read\_csv('train.csv')

1. EDA

print(df.describe())

1. Target Variable

sns.countplot(x='Survived', data=df)

plt.title("Survival Distribution")

plt.show()

1. Age Distribution

sns.histplot(df['Age'].dropna(), bins=30, kde=True)

plt.title("Age Distribution")

plt.show()

1. Survive Counts by Gender

sns.countplot(x='Sex', hue='Survived', data=df)

plt.title("Survival by Gender")

plt.show()

1. Survive Counts by Passenger Class

sns.countplot(x='Pclass', hue='Survived', data=df)

plt.title("Survival by Class")

plt.show()

1. Data Processing and feature preprocessing

num\_features = ['Age', 'Fare']

cat\_features = ['Pclass', 'Sex', 'Embarked']

num\_pipeline = Pipeline([

('imputer', SimpleImputer(strategy='median')),

('scaler', StandardScaler())

])

cat\_pipeline = Pipeline([

('imputer', SimpleImputer(strategy='most\_frequent')),

('encoder', OneHotEncoder(handle\_unknown='ignore'))

])

preprocessor = ColumnTransformer([

('num', num\_pipeline, num\_features),

('cat', cat\_pipeline, cat\_features)

])

1. Prepare Data

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

X = df.drop(columns=['Survived', 'Name', 'Ticket', 'Cabin'])

y = df['Survived']

X\_train, X\_temp, y\_train, y\_temp = train\_test\_split(X, y, test\_size=0.3, random\_state=42)

X\_val, X\_test, y\_val, y\_test = train\_test\_split(X\_temp, y\_temp, test\_size=0.5, random\_state=42)

1. Model Training

model = Pipeline([

('preprocessor', preprocessor),

('classifier', RandomForestClassifier(random\_state=42))

])

param\_grid = {'classifier\_\_n\_estimators': [50, 100, 200], 'classifier\_\_max\_depth': [None, 10, 20]}

grid\_search = GridSearchCV(model, param\_grid, scoring='roc\_auc', cv=5)

grid\_search.fit(X\_train, y\_train)

1. Model Evaluation

from sklearn.metrics import roc\_auc\_score

best\_model = grid\_search.best\_estimator\_

y\_train\_pred = best\_model.predict\_proba(X\_train)[:, 1]

y\_val\_pred = best\_model.predict\_proba(X\_val)[:, 1]

y\_test\_pred = best\_model.predict\_proba(X\_test)[:, 1]

auc\_train = roc\_auc\_score(y\_train, y\_train\_pred)

auc\_val = roc\_auc\_score(y\_val, y\_val\_pred)

auc\_test = roc\_auc\_score(y\_test, y\_test\_pred)

print(f"AUC - Train: {auc\_train:.4f}, Validation: {auc\_val:.4f}, Test: {auc\_test:.4f}")