**Exp no: 3 Develop a linear regression model for forecasting time series data**

**Date: 18/02/25**

**Aim:**

To develop a linear regression model for forecasting time series data

### ****Steps for Time Series Sales Data Preprocessing****:

**Step 1: Load the Dataset**

1. **Load** the sales dataset from a CSV file into a Pandas DataFrame.
2. **Check** for missing values in the dataset and identify any columns with null values.

url = '/content/airline-passengers.csv'

data = pd.read\_csv(url, parse\_dates=['Month'], index\_col='Month')

data.head()

**Output:**



**Step 2: Visualize Missing Values**

plt.figure(figsize=(10, 6))

plt.plot(data, label='Monthly airline passengers')

plt.title('Airline Passengers Over Time')

plt.xlabel('Date')

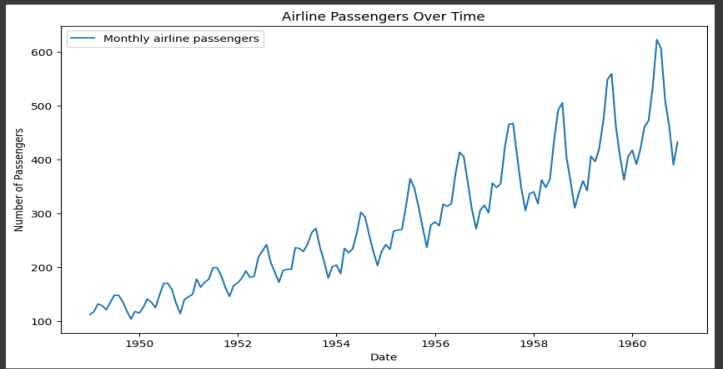
plt.ylabel('Number of Passengers')

plt.legend()

plt.savefig('timeseries\_plot.png')

plt.show()

**Output:**



**Step 3: preparing the data for linear regression**

data['Month'] = data.index

data['Month'] = data['Month'].map(pd.Timestamp.toordinal)

# Defining the feature (X) and target (y) variables

X = data[['Month']]

y = data['Passengers']

# Splitting the data into training and test sets in proportion of 80% and 20% respectively

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, shuffle=False)Step 4: Remove Duplicate Rows

1. Check for duplicate rows in the dataset.
2. Remove duplicates to ensure that there is no redundancy in the dataset, which could bias the model.

# Initializing the linear regression model

model = LinearRegression()

# Training the model on the training data available

model.fit(X\_train, y\_train)

**Step 4:Making predictions on the testing data**

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

# Evaluating the models using MSE metric

mse = mean\_squared\_error(y\_test, y\_pred)

print(f'Mean Squared Error: {mse}')

**Step 5: Plotting the actual vs predicted values inorder to visualize model performance**

plt.figure(figsize=(10, 6))

plt.plot(data.index, data['Passengers'], label='Actual')

plt.plot(X\_test.index, y\_pred, label='Predicted', color='red')

plt.title('Actual vs Predicted Airline Passengers')

plt.xlabel('Date')

plt.ylabel('Number of Passengers')

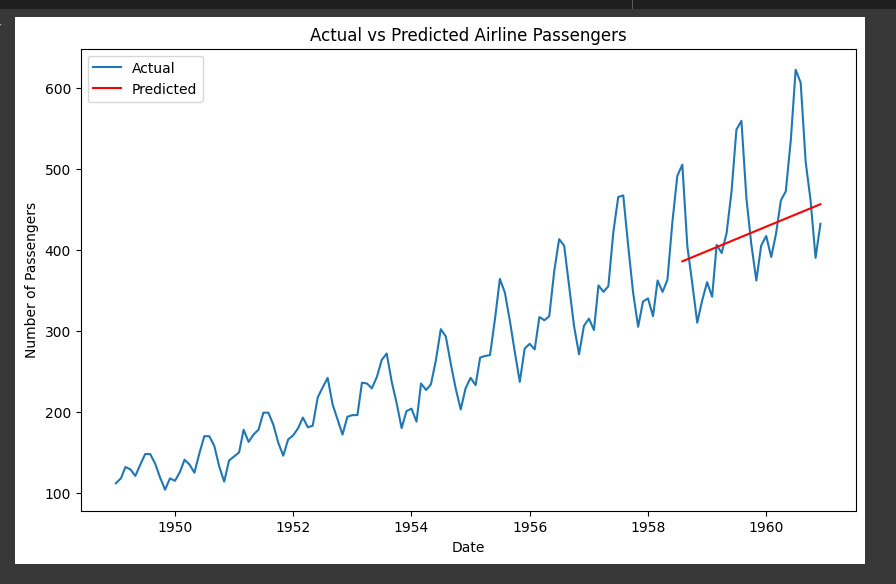
plt.legend()

# Saving the plot as an image

plt.savefig('actual\_vs\_predicted.png')

plt.show()

**Output:**



Result:

Thus the program is implemented. 221501024