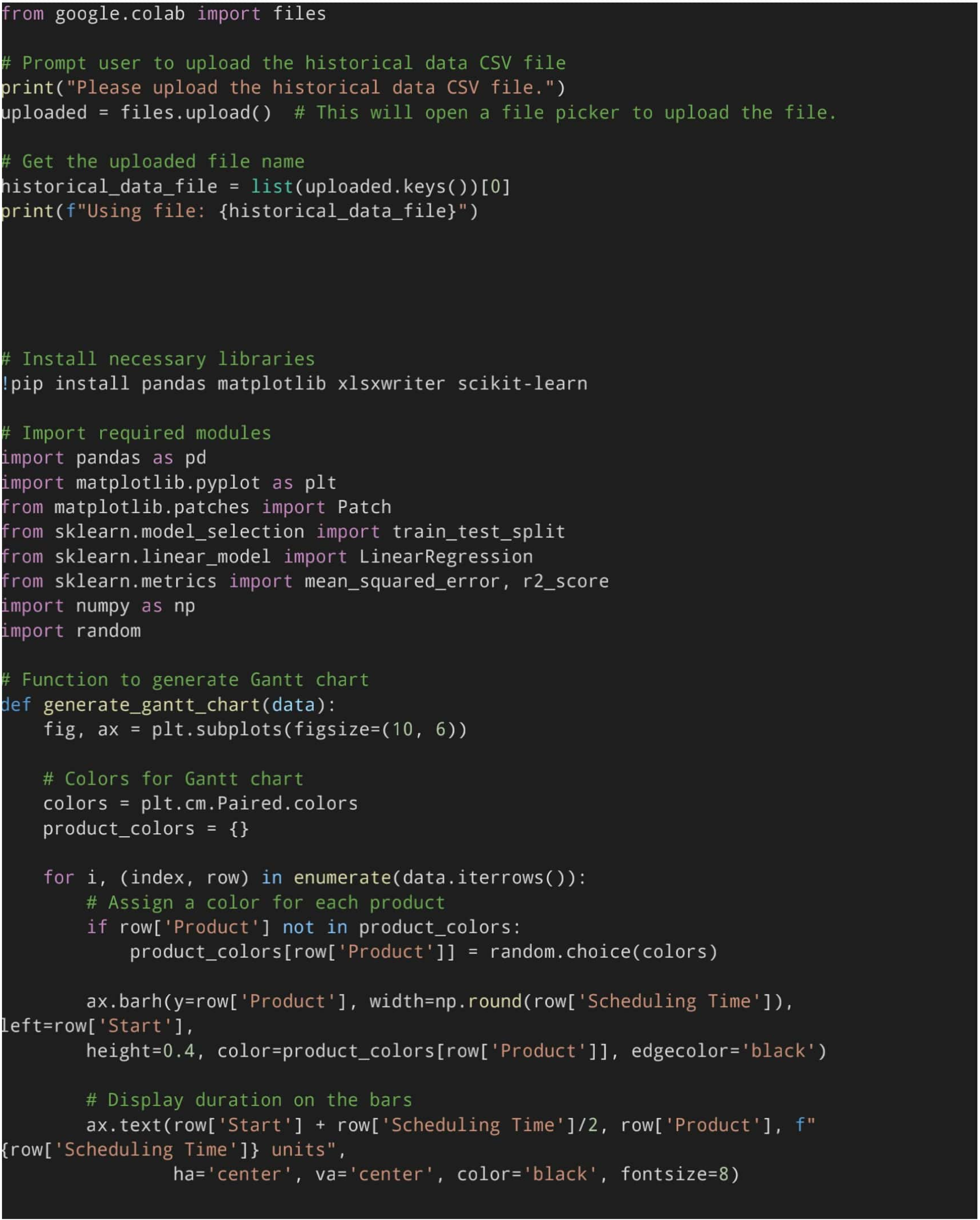

```
from google.colab import files

# Prompt user to upload the historical data CSV file
print("Please upload the historical data CSV file.")
uploaded = files.upload()  # This will open a file picker to upload the file.

# Get the uploaded file name
historical_data_file = list(uploaded.keys())[0]
print(f"Using file: {historical_data_file}")





# Install necessary libraries
!pip install pandas matplotlib xlsxwriter scikit-learn

# Import required modules
import pandas as pd
import matplotlib.pyplot as plt
from matplotlib.patches import Patch
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error, r2_score
import numpy as np
import random

# Function to generate Gantt chart
def generate_gantt_chart(data):
    fig, ax = plt.subplots(figsize=(10, 6))

    # Colors for Gantt chart
    colors = plt.cm.Paired.colors
    product_colors = {}

    for i, (index, row) in enumerate(data.iterrows()):
        # Assign a color for each product
        if row['Product'] not in product_colors:
            product_colors[row['Product']] = random.choice(colors)

        ax.barh(y=row['Product'], width=np.round(row['Scheduling Time']),
left=row['Start'],
        height=0.4, color=product_colors[row['Product']], edgecolor='black')

        # Display duration on the bars
        ax.text(row['Start'] + row['Scheduling Time']/2, row['Product'], f"
{row['Scheduling Time']} units",
                ha='center', va='center', color='black', fontsize=8)
```

```
    # Set labels and title
    ax.set_xlabel("Time Units")
    ax.set_ylabel("Products")
    ax.set_title("Production Scheduling Gantt Chart")
    plt.grid(True, which='both', linestyle='--', linewidth=0.5)

    # Create legend
    legend_elements = [Patch(facecolor=color, edgecolor='black', label=product)
                       for product, color in product_colors.items()]
    ax.legend(handles=legend_elements, title="Products")

    plt.show()

# Function to export data to Excel
def export_to_excel(data, file_name="production_schedule.xlsx"):
    with pd.ExcelWriter(file_name, engine='xlsxwriter') as writer:
        data.to_excel(writer, sheet_name='Schedule Data', index=False)

        # Access the workbook and worksheet for formatting
        workbook = writer.book
        worksheet = writer.sheets['Schedule Data']

        # Add Gantt Chart in Excel
        chart = workbook.add_chart({'type': 'bar'})

        for i, product in enumerate(data['Product']):
            chart.add_series({
                'name':       ["Schedule Data", 0, 0],
                'categories': ["Schedule Data", 1, 0, len(data), 0],
                'values':     ["Schedule Data", 1, 3, len(data), 3],
                'fill':       {'color': random.choice(['#FF5733', '#33FF57', '#3357FF'])}
            })

        chart.set_title({'name': 'Gantt Chart'})
        chart.set_x_axis({'name': 'Time Units'})
        chart.set_y_axis({'name': 'Products'})
        worksheet.insert_chart("G10", chart)

    print(f"Data and Gantt chart exported to {file_name}")

# Function to train and predict using Linear Regression
def train_and_predict(historical_data_file, input_data):
    # Load historical data
    historical_data = pd.read_csv(historical_data_file)

    # Features and target
    X = historical_data[['Available', 'Sold']]
    y = historical_data['Scheduling Time']

    # Split the data
    X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,
random_state=42)

    # Train the model
    model = LinearRegression()
    model.fit(X_train, y_train)

    # Evaluate the model
    y_pred = model.predict(X_test)
    print(f"Model Performance:")
    print(f"Mean Squared Error: {mean_squared_error(y_test, y_pred):.2f}")
    print(f"R2 Score: {r2_score(y_test, y_pred):.2f}")

    # Predict scheduling times for input data
    input_features = input_data[['Available', 'Sold']]
    input_data['Scheduling Time'] = np.round(model.predict(input_features))
    return input_data

# Main Program
if __name__ == "__main__":
    # Load historical data and new input data
   # Use uploaded file from Colab
    historical_data_file = list(uploaded.keys())[0]
    num_products = int(input("Enter number of products: "))

    product_data = []
    for i in range(num_products):
        product = input(f"Enter name of product {i+1}: ")
        quantity_available = int(input(f"Enter number of products available for {product}: "))
        quantity_sold = int(input(f"Enter number of products sold for {product}: "))
        product_data.append([product, quantity_available, quantity_sold])

    # Convert input data to DataFrame
    input_data = pd.DataFrame(product_data, columns=['Product', 'Available', 'Sold'])

    # Predict scheduling times using the trained model
    input_data = train_and_predict(historical_data_file, input_data)

    # Add start times
    start_time = 0
    start_times = []
    for time in input_data['Scheduling Time']:
        start_times.append(start_time)
        start_time += time
    input_data['Start'] = start_times

    # Generate Gantt Chart
    generate_gantt_chart(input_data)

    # Export to Excel
    export_to_excel(input_data)

    # Display Scheduling Table
    print("\nProduction Schedule Data:\n")
    print(input_data)
```

[Quoted text hidden]