Project Report

Project Title

Telecom Call Records Analysis

Project Description

This project focuses on creating an interactive web-based dashboard and a Python-based analysis script for telecom call records.

The dashboard (HTML, CSS, JavaScript, PapaParse, Plotly.js) allows users to upload CSV files, filter customers, and visualize insights like top customers, call duration trends, and call type distributions.

The Python script (using pandas) performs data preprocessing and summarization. It automatically generates:
- Customer summary (total/average call duration, number of calls).
- Call type summary (local vs international).
- Daily summary of call records.

Together, the web dashboard and backend analysis script help telecom companies better understand customer call behaviors and usage trends.

Team Member

NAME : CHETHANA B M

USN : 4GW23CI011

BRANCH : CSE - (AI&ML)

SEMESTER : 5

Index

Detailed Explanation

Overview

The project has two main parts:
1. Web Dashboard – User-friendly interface for interactive analysis.
2. Python Backend – Automated data processing for summaries.

Use Case Explanations

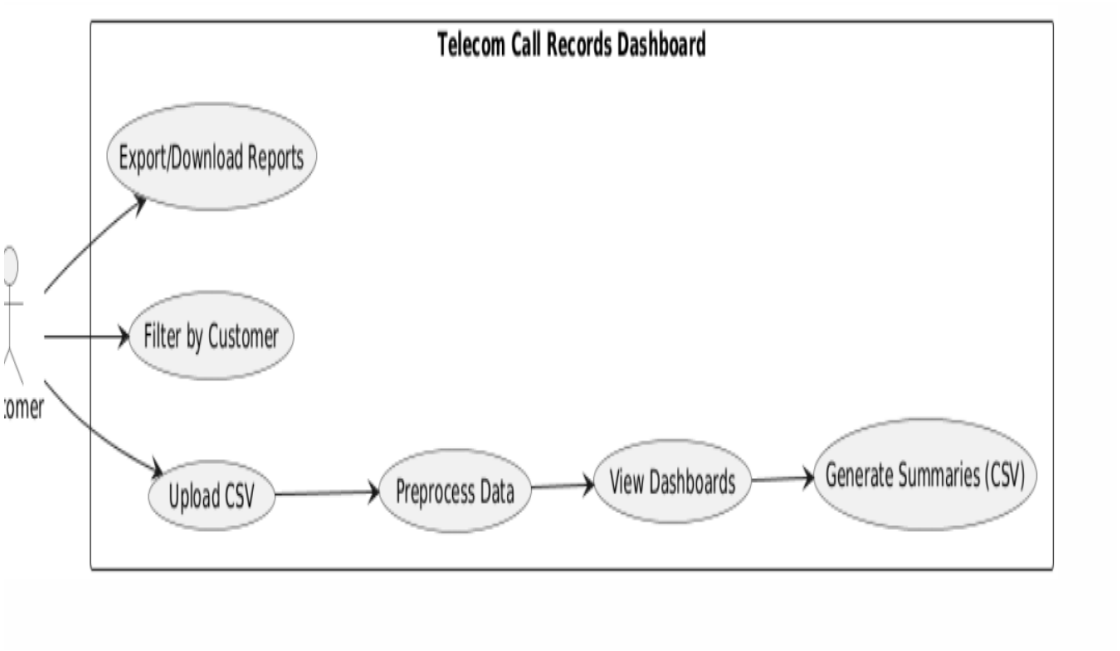
- All Customers Call Duration → Bar chart of total duration.
- Top 5 Customers → Identifies heavy users.
- Calls > 10 Minutes → Detects long-duration calls.
- International Calls > 30 Minutes → Highlights expensive long-distance calls.
- Daily Call Summary → Trend of usage per day.
- Call Type Distribution → Pie chart of local vs international usage.
- Python Summaries → Generates 3 CSV reports: customer, call type, and daily summary.

Algorithms

- Call Duration Calculation = (EndTime - StartTime) / 60
- Categorization Algorithm → Local, International, Other
- Group-by Aggregation → Using pandas.groupby for summaries

UML Diagrams

1. Use Case Diagram.
   - **Purpose:** Shows the interaction between the **user** and the **dashboard system**.
   - **Actors:**
- **User:** The person who uploads CSV files, selects filters, and views charts.
   - **Use Cases:**
- Upload Call Records (CSV)
- View All Customers Call Duration
- View Top 5 Customers
- View Long Calls (>10 Minutes)
- View International Calls (>30 Minutes)
- View Daily Call Summary
- View Call Type Distribution
- Filter by Customers
   - **Explanation:**
     The diagram helps users understand what functionalities the system provides and how they interact with it. Each use case is a **feature the user can perform**.



2. Activity Diagram

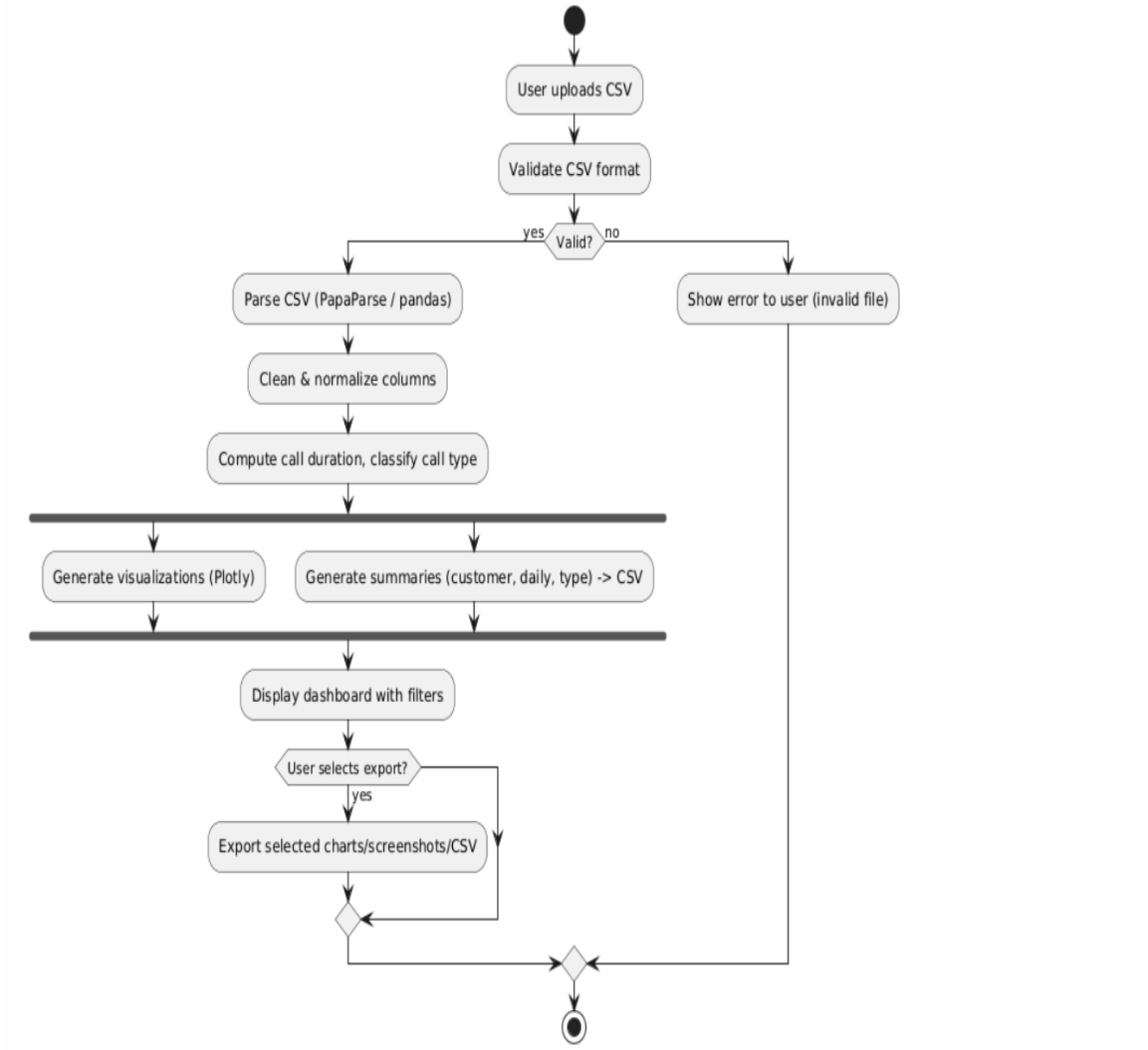
   Purpose: Represents the **workflow of the dashboard** from start to finish.

   Key Activities:

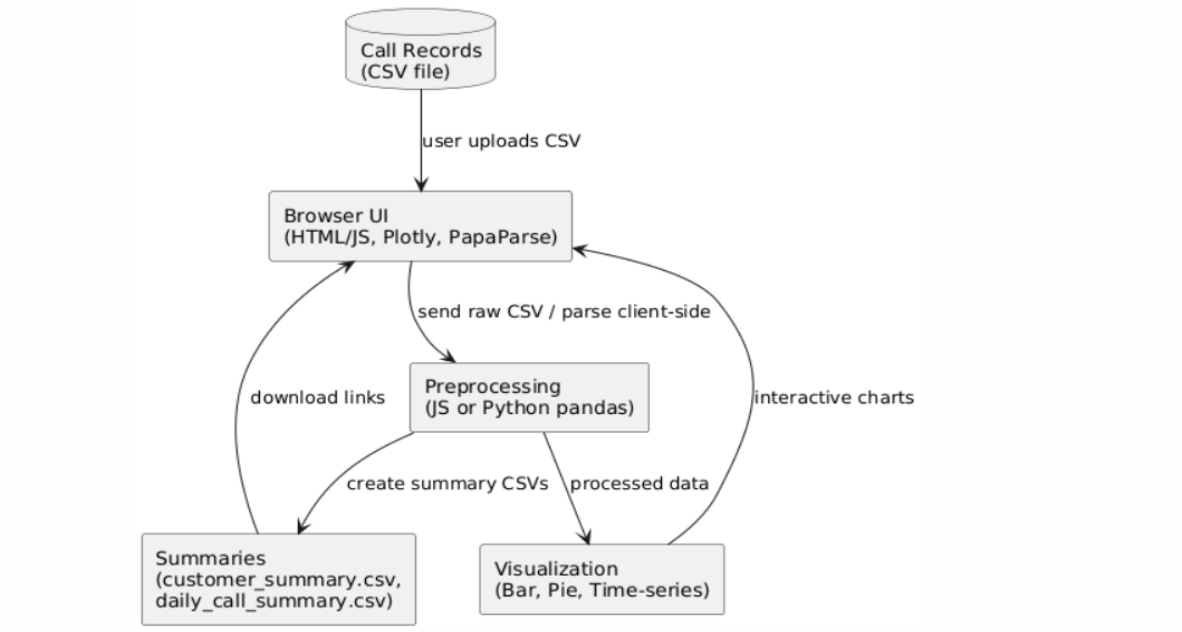
1. User uploads CSV file
2. System parses and preprocesses data
3. System generates unique customer list for filtering
4. User selects filter options (optional)
5. User clicks a button to view a chart
6. System computes necessary summary (total duration, top customers, etc.)
7. System generates and displays chart
8. User can repeat steps with different filters or charts

   Explanation:

   The activity diagram shows the **step-by-step process** of how data flows through the dashboard and how the user interacts with the system to get insights.


User uploads CSV
Validate CSV format
Valid?
yes
no
Parse CSV (PapaParse / pandas)
Show error to user (invalid file)
Clean & normalize columns
Compute call duration, classify call type
Generate visualizations (Plotly)
Generate summaries (customer, daily, type) -> CSV
Display dashboard with filters
User selects export?
yes
Export selected charts/screenshots/CSV

3. Data Flow Diagram (DFD) → Data source (CSV) → Processing (Python + JS) → Visualization & Reports.



 Purpose: Shows how **data moves through the system**.

 Key Components:

- **Input:** CSV file containing raw call records
- **Processing:**
    - Data cleaning
    - Call duration calculation
    - Call type categorization
    - Customer and daily aggregation
- **Output:** Interactive charts (bar charts, pie charts, time series)

 Explanation:
This diagram makes it clear how **raw data transforms into meaningful visual outputs** through the system's processing steps.

Front-end Design

The front-end is created using:
- HTML5 – Structure
- CSS – Styling and layout
- JavaScript (Plotly.js) – Interactive graphs
- PapaParse.js – CSV reading in browser

Code

Python (Data Processing)

```
import pandas as pd

# Load original call records
df = pd.read_csv("call_records.csv")

# Ensure column names are consistent
df.columns = [c.strip() for c in df.columns]

# Convert call times to datetime
df['CallStartTime'] = pd.to_datetime(df['CallStartTime'], dayfirst=True)
df['CallEndTime'] = pd.to_datetime(df['CallEndTime'], dayfirst=True)

# Calculate call duration in minutes (override if needed)
df['CallDuration'] = (df['CallEndTime'] - df['CallStartTime']).dt.total_seconds() / 60

# ---------- Customer Summary ----------
customer_summary = df.groupby("CustomerID").agg(
    NumberOfCalls=("CallID", "count"),
    TotalDuration=("CallDuration", "sum"),
    AverageDuration=("CallDuration", "mean")
).reset_index()

customer_summary.to_csv("customer_summary.csv", index=False)

# ---------- Call Type Summary ----------
def categorize_call(call_type):
    call_type = str(call_type).strip().lower()
    if call_type in ["local", "domestic"]:
        return "Local"
    elif call_type in ["std", "isd", "international"]:
        return "International"
    else:
```

```
    return "Other"

df['CallTypeCategory'] = df['CallType'].apply(categorize_call)

call_type_summary = df.groupby("CallTypeCategory").agg(
  NumberOfCalls=("CallID", "count"),
  TotalDuration=("CallDuration", "sum")
).reset_index()

call_type_summary.to_csv("call_type_summary.csv", index=False)

# ---------- Daily Summary ----------
df['CallDate'] = df['CallStartTime'].dt.date

daily_summary = df.groupby("CallDate").agg(
  NumberOfCalls=("CallID", "count"),
  TotalDuration=("CallDuration", "sum")
).reset_index()

daily_summary.to_csv("daily_call_summary.csv", index=False)

print("All summary CSVs generated successfully!")
```

INDEX.html code

```
<!DOCTYPE html>

<html lang="en">

<head>

  <meta charset="UTF-8">

  <title>Telecom Call Records Dashboard</title>

  <script
src="https://cdn.jsdelivr.net/npm/papaparse@5.4.1/papaparse.min.js"></script>

  <script src="https://cdn.plot.ly/plotly-latest.min.js"></script>

  <style>

    body { font-family: Arial, sans-serif; margin: 20px; background-color: #f4f4f9; }
```

```
    .header { text-align: center; margin-bottom: 20px; }
    .header img { width: 100px; height: auto; border-radius: 10px; box-shadow: 0 4px 8px rgba(0,0,0,0.2); margin-bottom: 10px; }
    h1 { text-align: center; margin-bottom: 20px; }
    .buttons { text-align: center; margin-bottom: 20px; }
    .filter { text-align: center; margin-bottom: 20px; }
    button { margin: 5px; padding: 10px 15px; border: none; background-color: #007bff; color: white; border-radius: 5px; cursor: pointer; }
    button:hover { background-color: #0056b3; }
    #chart { width: 100%; max-width: 900px; margin: 0 auto; }
    .checkbox-container { display: inline-block; text-align: left; margin: 5px 10px; }
    #csvFileInput { margin: 10px 0; padding: 5px; }
  </style>
</head>
<body>

<div class="header">
  <img src="https://cdn-icons-png.flaticon.com/512/15/15874.png" alt="Telecom Logo">
  <h1>Telecom Call Record Dashboard</h1>
</div>

<div class="buttons">
  <input type="file" id="csvFileInput" accept=".csv"><br>
  <button onclick="showAllCustomers()">All Customers Call Duration</button>
  <button onclick="showTopCustomers()">Top 5 Customers</button>
  <button onclick="showLongCalls()">Calls > 10 Minutes</button>
  <button onclick="showIntlLongCalls()">International Calls > 30 Min</button>
```

```
  <button onclick="showDailyCallSummary()">Daily Call Duration</button>
  <button onclick="showCallTypePie()">Call Type Distribution</button>
</div>

<div class="filter">
  <h3>Select Customers to Filter:</h3>
  <div id="customerCheckboxes"></div>
  <button onclick="plotFilteredTop5()">Top 5 Customers (Filtered)</button>
  <button onclick="plotFilteredLongCalls()">Calls > 10 Min (Filtered)</button>
  <button onclick="plotFilteredIntlCalls()">Intl > 30 Min (Filtered)</button>
</div>

<div id="chart"></div>

<script>
  let data = [];

  // Load CSV
  document.getElementById('csvFileInput').addEventListener('change', function(e){
    const file = e.target.files[0];
    if (!file) return;

    Papa.parse(file, {
      header: true,
      dynamicTyping: true,
      skipEmptyLines: true,
```

```
        complete: function(results){
            data = results.data;
            preprocessData();
            populateCustomerCheckboxes();
            alert("CSV loaded! Use buttons or filters to plot graphs.");
        }
    });
});

// Preprocess CSV
function preprocessData(){
    data.forEach(row => {
        row.CustomerID = row.CustomerID?.toString().trim();
        row.CallStartTime = row.CallStartTime?.toString().trim();
        row.CallEndTime = row.CallEndTime?.toString().trim();
        row.CallType = row.CallType?.toString().trim();

        row.callstarttime = new Date(row.CallStartTime);
        row.callendtime = new Date(row.CallEndTime);
        row.call_duration_minutes = (row.callendtime - row.callstarttime)/60000;

        const type = row.CallType.toLowerCase();
        if(['local','domestic'].includes(type)) row.call_type_category = 'Domestic';
        else if(['std','isd','international'].includes(type)) row.call_type_category =
'International';
        else row.call_type_category = 'Other';
```

```
        row.call_date = row.callstarttime.toISOString().slice(0,10);
      });
    }

    // Populate checkboxes for filtering
    function populateCustomerCheckboxes(){
      const container = document.getElementById('customerCheckboxes');
      container.innerHTML = '';
      const uniqueIDs = [...new Set(data.map(r => r.CustomerID))];
      uniqueIDs.forEach(id => {
        const div = document.createElement('div');
        div.className = 'checkbox-container';
        div.innerHTML = `<input type="checkbox" class="customerFilter" value="${id}"
id="cust_${id}">
                    <label for="cust_${id}">${id}</label>`;
        container.appendChild(div);
      });
    }

    function getSelectedCustomers(){
      const checkboxes = document.querySelectorAll('.customerFilter:checked');
      return Array.from(checkboxes).map(cb => cb.value);
    }

    // ===== Modified Buttons to Respect Checkboxes =====
```

```
  function showAllCustomers(){
    const selected = getSelectedCustomers();
    const summary = {};
    data.forEach(r => {
      if(selected.length===0 || selected.includes(r.CustomerID)) {
        summary[r.CustomerID] = (summary[r.CustomerID] || 0) + r.call_duration_minutes;
      }
    });
    const trace = { x:Object.keys(summary),
y:Object.values(summary).map(v=>v.toFixed(2)), type:'bar', marker:{color:'orange'} };
    Plotly.newPlot('chart',[trace], {title:'Total Call Duration for Selected Customers'});
  }

  function showTopCustomers(){
    const selected = getSelectedCustomers();
    const summary = {};
    data.forEach(r=>{
      if(selected.length===0 || selected.includes(r.CustomerID))
        summary[r.CustomerID] = (summary[r.CustomerID] || 0) + r.call_duration_minutes;
    });
    const sorted = Object.entries(summary).sort((a,b)=>b[1]-a[1]).slice(0,5);
    const trace = {x:sorted.map(e=>e[0]), y:sorted.map(e=>e[1].toFixed(2)), type:'bar',
marker:{color:'orange'}};
    Plotly.newPlot('chart',[trace],{title:'Top 5 Customers'});
  }

  function showLongCalls(){
```

```
    const selected = getSelectedCustomers();
    const filtered = data.filter(r=>r.call_duration_minutes>10 && (selected.length===0 ||
selected.includes(r.CustomerID)));
    const counts = {};
    filtered.forEach(r=>counts[r.CustomerID]=(counts[r.CustomerID]||0)+1);
    const trace = {x:Object.keys(counts), y:Object.values(counts), type:'bar',
marker:{color:'green'}};
    Plotly.newPlot('chart',[trace],{title:'Calls > 10 Minutes'});
  }

  function showIntlLongCalls(){
    const selected = getSelectedCustomers();
    const filtered = data.filter(r=>r.call_type_category==='International' &&
r.call_duration_minutes>30 && (selected.length===0 || selected.includes(r.CustomerID)));
    const counts = {};
    filtered.forEach(r=>counts[r.CustomerID]=(counts[r.CustomerID]||0)+1);
    const trace =
{x:Object.keys(counts),y:Object.values(counts),type:'bar',marker:{color:'red'}};
    Plotly.newPlot('chart',[trace],{title:'International Calls > 30 Min'});
  }

  function showDailyCallSummary(){
    const selected = getSelectedCustomers();
    const daily = {};
    data.forEach(r=>{
      if(selected.length===0 || selected.includes(r.CustomerID)){
        daily[r.call_date] = (daily[r.call_date] || 0) + r.call_duration_minutes;
      }
```

```
    });
    const trace = {x:Object.keys(daily),y:Object.values(daily),type:'bar',marker:{color:'purple'}};
    Plotly.newPlot('chart',[trace],{title:'Daily Call Duration'});
  }

  function showCallTypePie(){
    const selected = getSelectedCustomers();
    const typeCounts = {};
    data.forEach(r=>{
      if(selected.length===0 || selected.includes(r.CustomerID)){
        typeCounts[r.call_type_category] = (typeCounts[r.call_type_category] || 0) + 1;
      }
    });
    const trace = {labels:Object.keys(typeCounts),values:Object.values(typeCounts),type:'pie',textinfo:"label+percent"};
    Plotly.newPlot('chart',[trace],{title:'Call Type Distribution'});
  }

  // Filtered Graphs (Checkbox buttons)
  function plotFilteredTop5(){
    const selected = getSelectedCustomers();
    if(selected.length===0){ alert("Select at least one customer."); return; }
    const summary = {};
    data.forEach(r=>{ if(selected.includes(r.CustomerID)) summary[r.CustomerID]=(summary[r.CustomerID]||0)+r.call_duration_minutes; });
    const sorted = Object.entries(summary).sort((a,b)=>b[1]-a[1]).slice(0,5);
```

```
    const trace =
{x:sorted.map(e=>e[0]),y:sorted.map(e=>e[1].toFixed(2)),type:'bar',marker:{color:'orange'
}};

    Plotly.newPlot('chart',[trace],{title:'Top 5 Customers (Filtered)'});

  }


  function plotFilteredLongCalls(){

    const selected = getSelectedCustomers();

    if(selected.length===0){ alert("Select at least one customer."); return; }

    const filtered = data.filter(r=>r.call_duration_minutes>10 &&
selected.includes(r.CustomerID));

    const counts = {};

    filtered.forEach(r=>counts[r.CustomerID]=(counts[r.CustomerID]||0)+1);

    const trace =
{x:Object.keys(counts),y:Object.values(counts),type:'bar',marker:{color:'green'}};

    Plotly.newPlot('chart',[trace],{title:'Calls > 10 Min (Filtered)'});

  }


  function plotFilteredIntlCalls(){

    const selected = getSelectedCustomers();

    if(selected.length===0){ alert("Select at least one customer."); return; }

    const filtered = data.filter(r=>r.call_type_category==='International' &&
r.call_duration_minutes>30 && selected.includes(r.CustomerID));

    const counts = {};

    filtered.forEach(r=>counts[r.CustomerID]=(counts[r.CustomerID]||0)+1);

    const trace =
{x:Object.keys(counts),y:Object.values(counts),type:'bar',marker:{color:'red'}};

    Plotly.newPlot('chart',[trace],{title:'Intl Calls > 30 Min (Filtered)'});
```

```
  }


</script>

</body>

</html>
```

Explanation of the Code

- Data Cleaning: Removes inconsistencies, trims column names.
- Datetime Conversion: Ensures correct parsing of call start/end times.
- Call Duration: Calculates duration in minutes.
- Customer Summary: Groups by CustomerID to get total/average usage.
- Call Type Summary: Categorizes call type into Local/International/Other.
- Daily Summary: Groups data by CallDate for daily call analytics.
- CSV Exports: Three summary reports are saved for further use in dashboard/analysis.

1. Data Input

- The system starts by **loading a CSV file** containing telecom call records.
- Each record includes information such as:
    - **Call ID**
    - **Customer ID**
    - **Call Type** (Local, STD, ISD, International)
    - **Call Start and End Time**
    - **Call Duration**
- The CSV is parsed and converted into a **structured format** that the dashboard can use.

---

2. Data Preprocessing

- The code **cleans the data**: removes extra spaces, fixes missing or inconsistent values.
- Converts string timestamps into **actual date-time objects** to calculate call durations accurately.
- Calculates **call duration in minutes** for each record.
- Categorizes calls into **Domestic**, **International**, or **Other**.
- Extracts the **call date** for daily summaries.

Purpose: Prepare data for accurate analysis and visualization.

---

3. Customer Selection & Filtering

- The code identifies **unique customers** and generates **checkboxes** for filtering.
- Users can select one or more customers to focus the analysis.
- This makes the dashboard **interactive and customizable**.

---

4. Visualization

- Uses **Plotly.js** to create interactive charts:
    1. **Bar Charts** – Total call duration per customer, Top 5 customers, long calls.
    2. **Pie Charts** – Distribution of call types.
    3. **Time Series / Daily Summary** – Total calls or durations per day.
- Charts are updated **dynamically** based on:
    - Button clicks (All Customers, Top 5, Long Calls, etc.)
    - Customer selection (filtered charts)

Purpose: Convert raw data into **easy-to-understand visual insights**.

---

5. Filtered Analysis

- The code allows **filtering data by customer selection**:
    - Top 5 customers among selected users

    - Long calls among selected users
    - International calls among selected users
- Ensures the dashboard can show **targeted insights**.

---

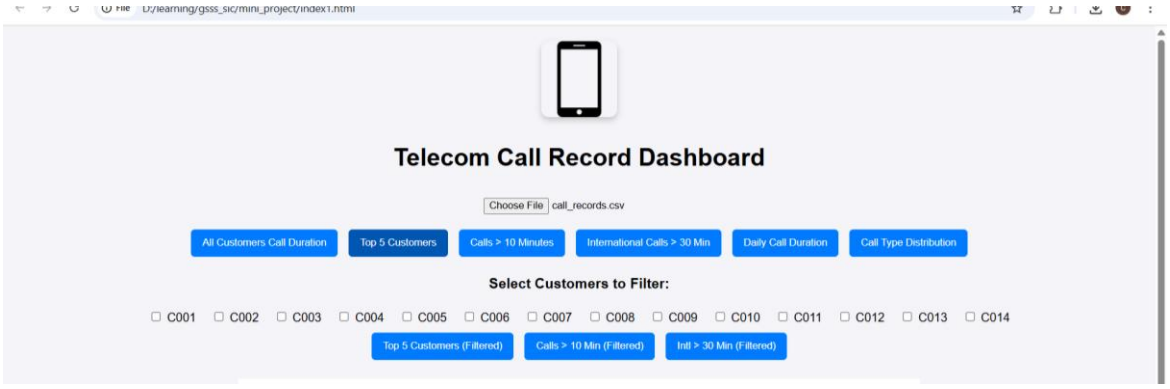
6. Summary

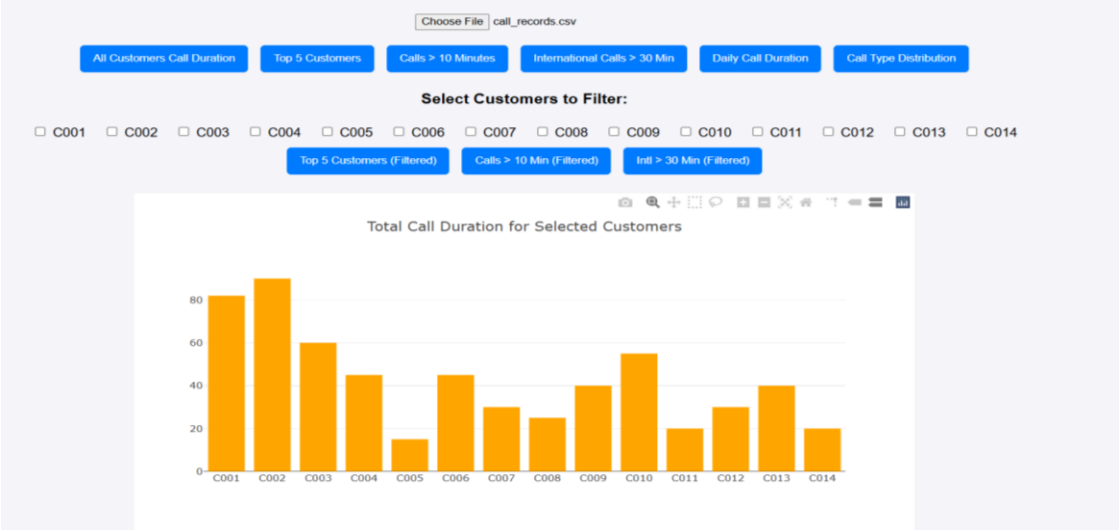
- The dashboard code combines:
    - **Data reading & cleaning** → ensures reliable data.
    - **Calculations & categorization** → prepares metrics (total calls, durations, call type).
    - **Interactive visualization** → charts and graphs that update based on user inputs.
    - **Filtering** → makes the tool user-friendly and adaptable for analysis.

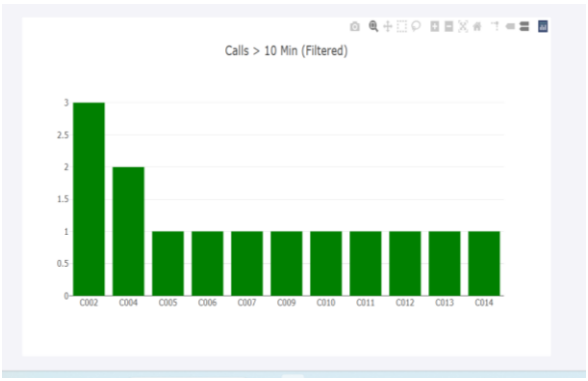
Key Idea: The code transforms **raw call records** into **insights for decision-making** by organizing, summarizing, and visualizing the data effectively.

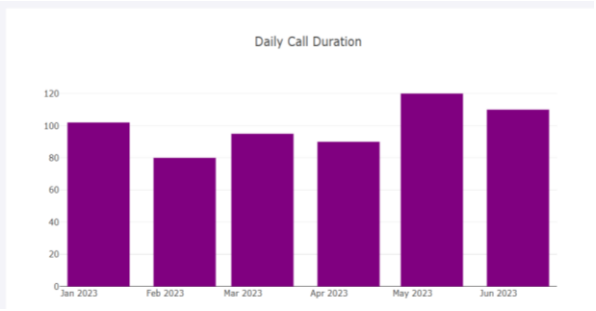
Output Screenshots with Explanation

1.Dashboard – Customer filters + Graphs.

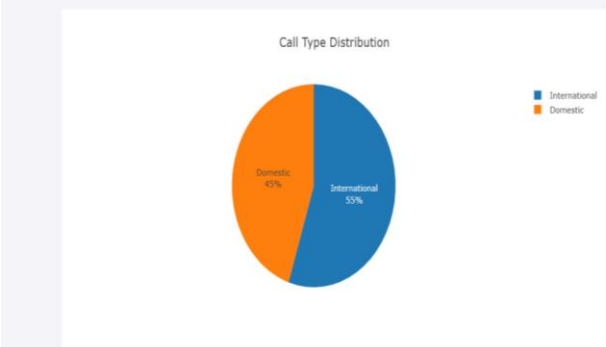


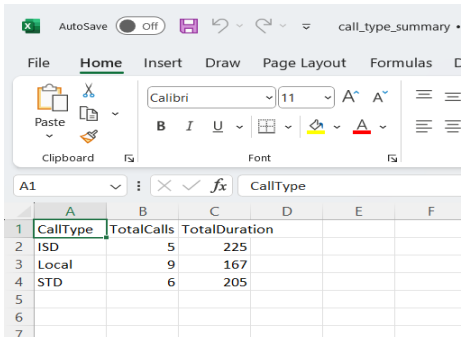

Choose File
call_records.csv
All Customers Call Duration
Top 5 Customers
Calls > 10 Minutes
International Calls > 30 Min
Daily Call Duration
Call Type Distribution
Select Customers to Filter:
C001 C002 C003 C004 C005 C006 C007 C008 C009 C010 C011 C012 C013 C014
Top 5 Customers (Filtered)
Calls > 10 Min (Filtered)
Intl > 30 Min (Filtered)
Total Call Duration for Selected Customers
0
20
40
60
80
C001 C002 C003 C004 C005 C006 C007 C008 C009 C010 C011 C012 C013 C014


Calls > 10 Min (Filtered)
0
0.5
1
1.5
2
2.5
3
C002 C004 C005 C006 C007 C009 C010 C011 C012 C013 C014


Daily Call Duration
0
20
40
60
80
100
120
Jan 2023
Feb 2023
Mar 2023
Apr 2023
May 2023
Jun 2023

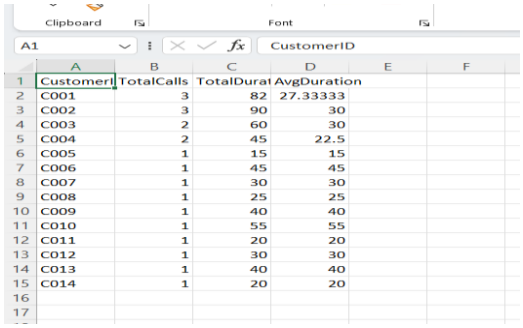

Top 5 Customers
0
20
40
60
80
C002 C001 C003 C010 C004


Call Type Distribution
International
Domestic
Domestic
45%
International
55%



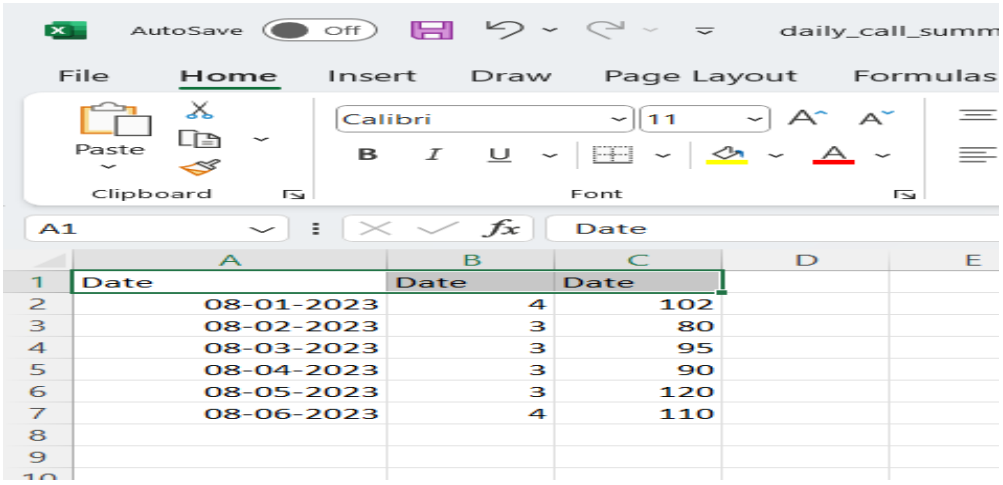
CallType	TotalCalls	TotalDuration
ISD	5	225
Local	9	167
STD	6	205

Call Type Summary CSV.



CustomerID	TotalCalls	TotalDura	AvgDuration
C001	3	82	27.33333
C002	3	90	30
C003	2	60	30
C004	2	45	22.5
C005	1	15	15
C006	1	45	45
C007	1	30	30
C008	1	25	25
C009	1	40	40
C010	1	55	55
C011	1	20	20
C012	1	30	30
C013	1	40	40
C014	1	20	20

Customer Summary CSV



Date	Date	Date
08-01-2023	4	102
08-02-2023	3	80
08-03-2023	3	95
08-04-2023	3	90
08-05-2023	3	120
08-06-2023	4	110

Daily Summary CSV

Output Explanation – Telecom Call Records Dashboard

The dashboard produces **interactive visual outputs** that help users analyze call data efficiently. Each chart or visualization represents different aspects of telecom usage.

---

1. All Customers Call Duration

- **Chart Type:** Bar chart
- **X-axis:** Customer IDs
- **Y-axis:** Total call duration (minutes)
- **Purpose:** Shows how much each customer has used the telecom service in terms of total call time.
- **Interpretation:** Taller bars indicate customers with higher call usage.

---

2. Top 5 Customers

- **Chart Type:** Bar chart
- **X-axis:** Top 5 Customer IDs
- **Y-axis:** Total call duration
- **Purpose:** Quickly identifies the most active customers.
- **Interpretation:** Useful for targeting high-value customers or understanding usage patterns.

---

3. Calls > 10 Minutes

- **Chart Type:** Bar chart
- **X-axis:** Customer IDs
- **Y-axis:** Number of long-duration calls (>10 minutes)
- **Purpose:** Highlights customers who make long conversations.
- **Interpretation:** Can indicate frequent heavy users or potential international/STD callers.

---

4. International Calls > 30 Minutes

- **Chart Type:** Bar chart
- **X-axis:** Customer IDs
- **Y-axis:** Number of international calls longer than 30 minutes
- **Purpose:** Shows users making long international calls.
- **Interpretation:** Helps telecom companies monitor high international usage.

---

5. Daily Call Summary

- **Chart Type:** Bar chart (time series)
- **X-axis:** Dates
- **Y-axis:** Total call duration for all customers
- **Purpose:** Displays trends in daily call activity.
- **Interpretation:** Peaks show days of high activity; helps in planning network capacity.

---

6. Call Type Distribution

- **Chart Type:** Pie chart
- **Segments:** Domestic, International, Other
- **Purpose:** Shows the percentage of calls by type.
- **Interpretation:** Gives a quick overview of the call type mix across all customers.

---

7. Filtered Charts (Customer Selection)

- Users can select **specific customers** to view:
    - Top 5 among selected customers
    - Long calls among selected customers

    - International calls > 30 min among selected customers
- **Purpose:** Enables **focused analysis** on individual or group customer behavior.

---

8. Interactive Features

- Hovering over bars/pie slices shows exact **numbers and durations**.
- Filtering dynamically updates charts without reloading the page.
- Provides **quick insights** into call patterns, usage trends, and customer behavior.

Summary:
The output of the dashboard converts raw CSV data into **visual insights** — bar charts, pie charts, and time series — making it easier for users to **analyze call patterns, monitor high-value customers, and make data-driven decisions**.

Conclusion

This project demonstrates how data visualization and analysis can be integrated for telecom record analysis. The dashboard provides interactivity, while the Python script automates preprocessing and summaries. Together, they give a complete solution for telecom data insights.

Bibliography

- Pandas Documentation: https://pandas.pydata.org/
- Plotly.js Documentation: https://plotly.com/javascript/
- PapaParse Documentation: https://www.papaparse.com/
- W3Schools & MDN Web Docs for HTML/CSS/JavaScript