

## PAI ASSIGNMENT 2

**Q1:**

```
import requests

def getLiveScores():

    print("Fetching Live Cricket Scores...")

    with open("api.txt","r") as file:

        apiUrl = file.read()

    try:

        response = requests.get(apiUrl)

        if(response.status_code!=200):

            print(f"HTTP Error:{response.status_code}")

            print(f"Raw Response:{response.text}")

            return

        data = response.json()

        for key , value in data.items():

            if key != "data":

                print(f"{key}:{value}")

        print()

        if data.get("status") != "success":

            reason = data.get("API returned failure status")

            print(f"API Error:{reason}")

            return

        if "data" not in data or not data["data"]:

            print("No matches found")

        matchList = data["data"]

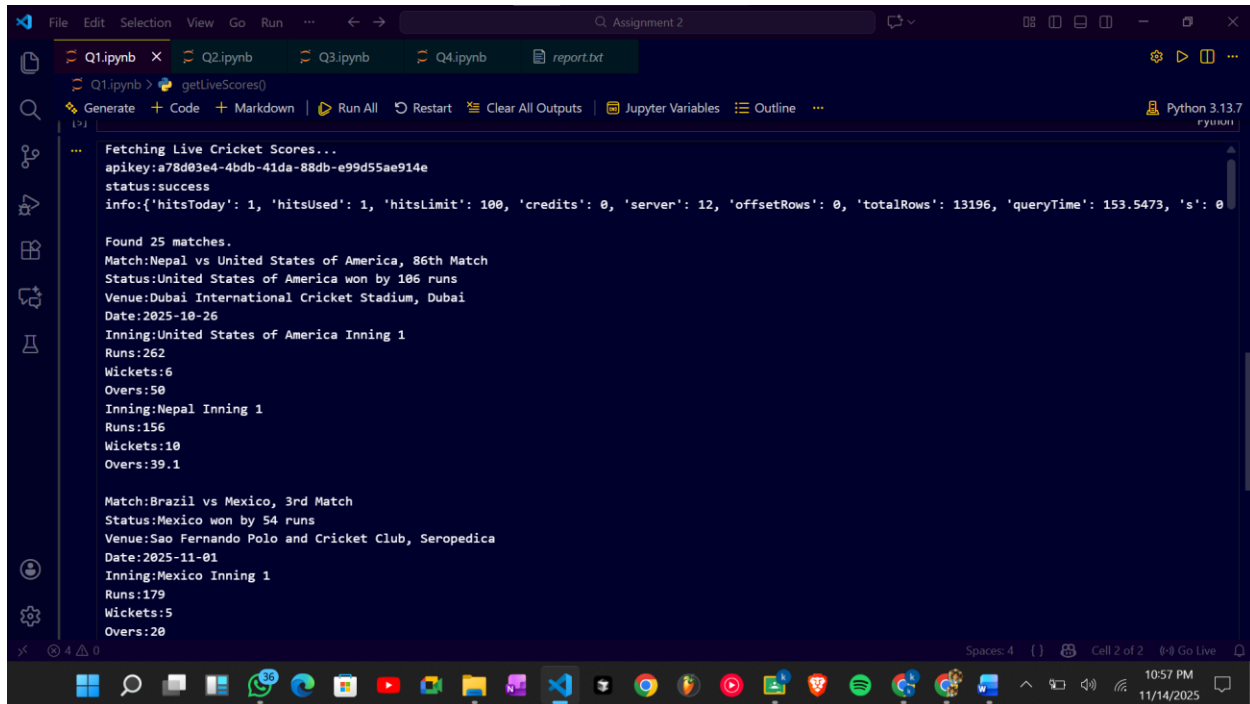
        print(f"Found {len(matchList)} matches.")
```

```
for match in matchList:

    print(f"Match:{match.get('name','N/A')}")
    print(f"Status:{match.get('status','N/A')}")
    print(f"Venue:{match.get('venue','N/A')}")
    print(f>Date:{match.get('date','N/A')}")
    if "score" in match and match["score"]:
        for inning in match["score"]:
            print(f"Inning:{inning.get('inning','N/A')}")
            print(f"Runs:{inning.get('r','N/A')}")
            print(f"Wickets:{inning.get('w','N/A')}")
            print(f>Overs:{inning.get('o','N/A')}")
        else:
            print("No score available.")
        print()
except requests.exceptions.RequestException as e:
    print(f"Error:{e}")

getLiveScores()
```

## Output:



```
... Fetching Live Cricket Scores...
apikey:a78d03e4-4bdb-41da-88db-e99d55ae914e
status:success
info:{'hitsToday': 1, 'hitsUsed': 1, 'hitsLimit': 100, 'credits': 0, 'server': 12, 'offsetRows': 0, 'totalRows': 13196, 'queryTime': 153.5473, 's': 0}

Found 25 matches.
Match:Nepal vs United States of America, 86th Match
Status:United States of America won by 106 runs
Venue:Dubai International Cricket Stadium, Dubai
Date:2025-10-26
Inning:United States of America Inning 1
Runs:262
Wickets:6
Overs:50
Inning:Nepal Inning 1
Runs:156
Wickets:10
Overs:39.1

Match:Brazil vs Mexico, 3rd Match
Status:Mexico won by 54 runs
Venue:Sao Fernando Polo and Cricket Club, Seropedica
Date:2025-11-01
Inning:Mexico Inning 1
Runs:179
Wickets:5
Overs:20
```

## Q2:

```
import numpy as np
```

```
my_list = []
```

```
with open("sensor_data.csv","r") as file:
```

```
    for item in file:
```

```
        row = item.strip().split(",")
```

```
        rowFloat = [float(value) for value in row]
```

```
        my_list.append(rowFloat)
```

```
Array = np.array(my_list)
```

```
Array[Array == -999] = np.nan
```

```
Array[(Array<0) | (Array>100)] = np.nan
```

```
columnsMean = np.nanmean(Array,axis=0)
```

```
rowMedian = np.nanmedian(Array,axis=1)
```

```
nanCount = np.sum(np.isnan(Array),axis=0)
```

```

sensorMostInvalid = np.argmax(nanCount)

maxInvalidCount = nanCount[sensorMostInvalid]

maxVal = np.nanmax(Array)

minVal = np.nanmin(Array)

dataNormalised = (Array - minVal) / (maxVal - minVal)

np.savetxt("sensor_data_normalized.csv",dataNormalised,delimiter=";",fmt='%.6f')

```

## Output:

```

1 0.140125,0.710750,0.471625,0.907000,0.141625,nan,0.259125,0.431875,0.363625,0.878750,0.069625,0.369250,0.337125,nan,0.914625,0.263750,nan,0
2 0.465500,0.771250,0.730500,0.017750,0.108500,0.693250,nan,0.990375,0.348125,0.813000,0.844625,0.603500,0.099750,0.310875,0.083250,0.571875,
3 0.363375,0.148125,0.273250,0.536000,0.912750,0.195125,0.082750,0.462250,0.179875,0.861500,0.868125,0.105625,0.742750,0.252000,0.096750,0.70
4 0.574875,0.840000,0.453625,0.040875,0.177250,0.942125,0.038000,0.841375,0.307125,nan,0.503500,0.887375,0.090375,0.587000,0.101125,0.746250,
5 0.834250,0.776375,nan,0.214500,0.568625,0.793750,0.882250,0.752250,0.722000,0.476375,0.968500,0.002750,0.981250,0.844875,nan,0.413375,nan,0
6 0.986000,0.262625,nan,0.805875,0.057000,nan,0.558250,0.107125,nan,0.976125,0.530250,nan,0.959750,0.179250,0.247375,0.633500,0.671625,0.9775
7 0.279875,0.442500,0.097500,0.173000,0.158000,0.849000,0.549375,0.114750,0.907125,0.108625,0.049250,0.702375,0.288625,0.153125,0.523875,0.09
8 0.846375,0.280875,nan,0.884875,0.984750,0.172875,0.513250,0.511500,0.467000,0.009125,0.691750,0.109875,0.162750,0.622625,0.000375,0.719500,
9 0.634750,0.261000,0.425750,0.170375,0.956625,0.893750,0.558000,0.636500,0.157250,0.102750,0.262000,0.351375,0.998750,0.942500,0.600000,0.71
10 0.708000,0.534625,0.688875,0.332875,0.188000,0.699125,0.568000,0.553375,0.902750,0.472875,0.496875,0.497500,0.980125,0.066250,0.454875,0.77
11 0.896250,0.228375,0.511000,0.538500,0.172500,0.399375,0.977500,0.772000,0.818750,0.051750,0.792375,0.544750,0.043375,0.205000,0.951750,0.39
12 0.093250,0.313000,nan,0.703375,0.871000,0.479375,0.886250,0.225750,0.616000,0.716875,0.042625,0.162875,0.200000,0.418875,0.768750,0.821000,
13 nan,0.048500,0.396500,0.035000,0.652875,0.167250,0.467875,0.995250,0.553625,0.896125,0.442750,0.453500,0.365500,0.864875,0.508500,0.644375,
14 0.893000,0.252375,nan,0.353875,0.858000,0.491000,0.105375,nan,0.422000,0.258750,0.875375,0.762000,0.970875,0.007125,0.274875,0.496000,0.375
15 0.281500,0.413750,0.148625,0.489125,0.872625,0.831125,0.776750,0.188250,0.504375,0.056375,0.016250,0.606125,0.897000,0.478125,0.491000,nan,
16 0.169625,0.534000,0.854750,0.063625,nan,0.347250,nan,0.113250,0.091625,0.603500,0.798750,0.627625,0.028125,0.259750,0.742250,0.746500,0.577
17 0.628375,0.853625,0.359750,0.542500,0.224125,0.106875,0.603500,0.451875,0.677250,0.195125,0.580875,0.381000,0.959000,0.900000,nan,0.133125,
18 nan,0.317375,0.423000,0.888375,0.791500,0.622625,0.742500,0.021375,nan,0.810875,0.910750,0.420125,0.090250,0.942875,0.576625,0.365750,0.859
19 0.228125,0.421750,0.184750,0.834750,0.675625,0.153500,0.234625,0.732125,0.730250,0.258625,0.122750,0.156625,0.120250,0.886625,0.786625,nan,
20 0.692375,0.091500,0.691000,0.926125,0.361000,0.284500,0.795125,0.917500,0.254625,0.548250,0.350375,0.934000,0.987000,0.895250,0.75
21 0.997625,0.094250,0.920000,nan,0.423500,0.559375,0.795125,0.228250,0.231625,0.816875,0.119375,0.631375,0.741125,0.771875,0.190000,0.192750,
22 0.596250,0.071125,nan,0.052250,0.848250,0.039750,0.758000,0.941625,0.894500,nan,0.513625,0.966875,0.034125,0.334750,0.415000,0.718750,0.109
23 nan,0.732125,0.009625,0.111750,0.168750,0.176875,0.080125,0.480750,0.837000,0.778625,0.075250,nan,0.574875,0.869125,0.757625,0.610500,0.363
24 0.613125,0.254875,0.631375,0.638250,0.483125,0.886000,0.950875,0.975375,0.797875,0.136500,0.760375,0.895375,0.104875,0.432500,0.948375,0.53
25 0.853125,0.224625,0.512750,nan,0.322250,0.856250,0.427875,0.368500,0.335375,0.231125,0.918625,0.494625,0.157750,0.981750,0.101250,0.370375,
26 0.336250,0.664625,0.020625,0.695625,0.954625,0.664375,0.478500,0.762625,0.292000,0.384750,0.458875,0.045875,nan,0.647125,0.461875,0.770625,
27 0.363500,0.432125,nan,0.731875,0.909250,0.929000,0.535250,0.323625,nan,0.621375,0.955625,0.561500,0.186500,0.562250,0.133750,0.991250,0.708
28 0.103500,0.005750,0.494000,0.787750,0.376750,0.148875,0.314375,0.035500,0.934125,0.650000,0.135000,0.544500,0.177000,0.561625,0.313000,0.20
29 0.637750,0.166000,0.342125,0.142125,0.321625,0.649875,0.776000,0.424875,0.217250,0.294125,0.870750,0.269125,0.703750,0.573875,0.249500,0.11

```

## Q3:

```

import pandas as pd

df = pd.read_csv("Titanic-Dataset.csv")

reportLines = []

for column in df.columns:

    dataType = df[column].dtype

    missingPart = df[column].isna().mean() * 100

    reportLines.append(f'{column}: {dataType}, missing:{missingPart}')

```

with open("inspect\_report.txt","w") as file:

for line in reportLines:

file.write(line+"\n")

```
df["Age"] = df.groupby(['Pclass','Sex'])["Age"].transform(lambda g : g.fillna(g.median()))
```

```
embarkedMode = df["Embarked"].mode()[0]
```

```
df["Embarked"] = df["Embarked"].fillna(embarkedMode)
```

```
if("Cabin" in df.columns):
```

```
df = df.drop(columns=["Cabin"])
```

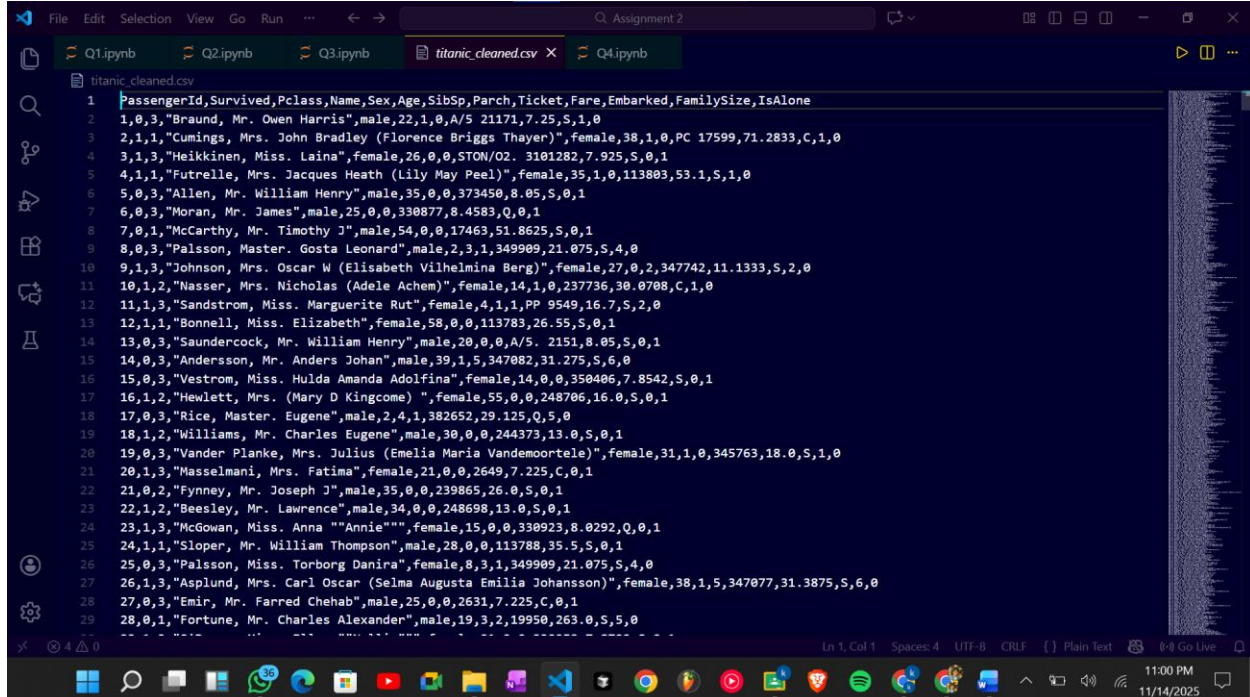
```
df['FamilySize'] = df["SibSp"] + df["Parch"]
```

```
df["IsAlone"] = (df["FamilySize"] == 0).astype(int)
```

```
df["Age"] = df["Age"].astype('int64')
```

```
df.to_csv("titanic_cleaned.csv",index=False)
```

**Output:**



```
1 PassengerId, Survived, Pclass, Name, Sex, Age, SibSp, Parch, Ticket, Fare, Embarked, FamilySize, IsAlone
2 1,0,3,"Braund, Mr. Owen Harris",male,22,1,0,A/5 21171,7.25,S,1,0
3 2,1,1,"Cumings, Mrs. John Bradley (Florence Briggs Thayer)",female,38,1,0,PC 17599,71.2833,C,1,0
4 3,1,3,"Heikkinen, Miss. Laina",female,26,0,0,STON/O2. 3101282,7.925,S,0,1
5 4,1,1,"Futrelle, Mrs. Jacques Heath (Lily May Peel)",female,35,1,0,113803,53.1,S,1,0
6 5,0,3,"Allen, Mr. William Henry",male,35,0,0,373450,8.05,S,0,1
7 6,0,3,"Moran, Mr. James",male,25,0,0,330877,8.4583,Q,0,1
8 7,0,1,"McCarthy, Mr. Timothy J",male,54,0,0,17463,51.8625,S,0,1
9 8,0,3,"Palsson, Master. Gosta Leonard",male,2,3,1,349909,21.075,S,4,0
10 9,1,3,"Johnson, Mrs. Oscar W (Elisabeth Vilhelmina Berg)",female,27,0,2,347742,11.1333,S,2,0
11 10,1,2,"Nasser, Mrs. Nicholas (Adele Achem)",female,14,1,0,237736,30.0708,C,1,0
12 11,1,3,"Sandstrom, Miss. Marguerite Rut",female,4,1,1,PP 9549,16.7,S,2,0
13 12,1,1,"Bonnell, Miss. Elizabeth",female,58,0,0,113783,26.55,S,0,1
14 13,0,3,"Saunderscock, Mr. William Henry",male,20,0,0,A/5 2151,8.05,S,0,1
15 14,0,3,"Andersson, Mr. Anders Johan",male,39,1,5,347082,31.275,S,6,0
16 15,0,3,"Vestrom, Miss. Hulda Amanda Adolfina",female,14,0,0,350406,7.8542,S,0,1
17 16,1,2,"Hewlett, Mrs. (Mary D Kingcome) ",female,55,0,0,248706,16.0,S,0,1
18 17,0,3,"Rice, Master. Eugene",male,2,4,1,382652,29.125,Q,5,0
19 18,1,2,"Williams, Mr. Charles Eugene",male,30,0,0,244373,13.0,S,0,1
20 19,0,3,"Vander Planke, Mrs. Julius (Emelia Maria Vandemoortele)",female,31,1,0,345763,18.0,S,1,0
21 20,1,3,"Masseimani, Mrs. Fatima",female,21,0,0,2649,7.225,C,0,1
22 21,0,2,"Fynney, Mr. Joseph J",male,35,0,0,239865,26.0,S,0,1
23 22,1,2,"Beesley, Mr. Lawrence",male,34,0,0,248698,13.0,S,0,1
24 23,1,3,"McGowan, Miss. Anna ""Annie""",female,15,0,0,330923,8.0292,Q,0,1
25 24,1,1,"Sloper, Mr. William Thompson",male,28,0,0,113788,35.5,S,0,1
26 25,0,3,"Palsson, Miss. Torborg Danira",female,8,3,1,349909,21.075,S,4,0
27 26,1,3,"Asplund, Mrs. Carl Oscar (Selma Augusta Emilia Johansson)",female,38,1,5,347077,31.3875,S,6,0
28 27,0,3,"Emir, Mr. Farred Chehab",male,25,0,0,2631,7.225,C,0,1
29 28,0,1,"Fortune, Mr. Charles Alexander",male,19,3,2,19950,263.0,S,5,0
```

#### Q4:

```
import pandas as pd

titanicDf = pd.read_csv("titanic_cleaned.csv")

faresDf = pd.read_csv("ticket_fares.csv")

df = pd.merge(titanicDf,faresDf,on="Ticket",how="left")

bins = [0,12,19,30,61,120]

labels = ['Child','Teenager','Young Adult','Adult','Senior']

df['AgeGroup'] = pd.cut(df['Age'],bins = bins,labels = labels,include_lowest=True)

survivedGenderAge =
df.groupby(['Age','AgeGroup'],observed=False)['Survived'].mean().reset_index()

survivedPClass = df.groupby("Pclass")["Survived"].mean().reset_index()

df["FareBin"] = pd.qcut(df['Fare_x'], q = 4 , labels = ["Low","Medium","High","VeryHigh"])

survivedFareBin = df.groupby("FareBin",observed=False)["Survived"].mean().reset_index()

paragraph1 = """Survival rates grouped by gender and age indicate that women and
children had higher chances of survival compared to men, adults, and seniors.

This supports the idea that women and children were given priority during evacuation."""

paragraph2 = """Passengers in 1st Class had significantly higher survival rates than those in
2nd and 3rd Class.

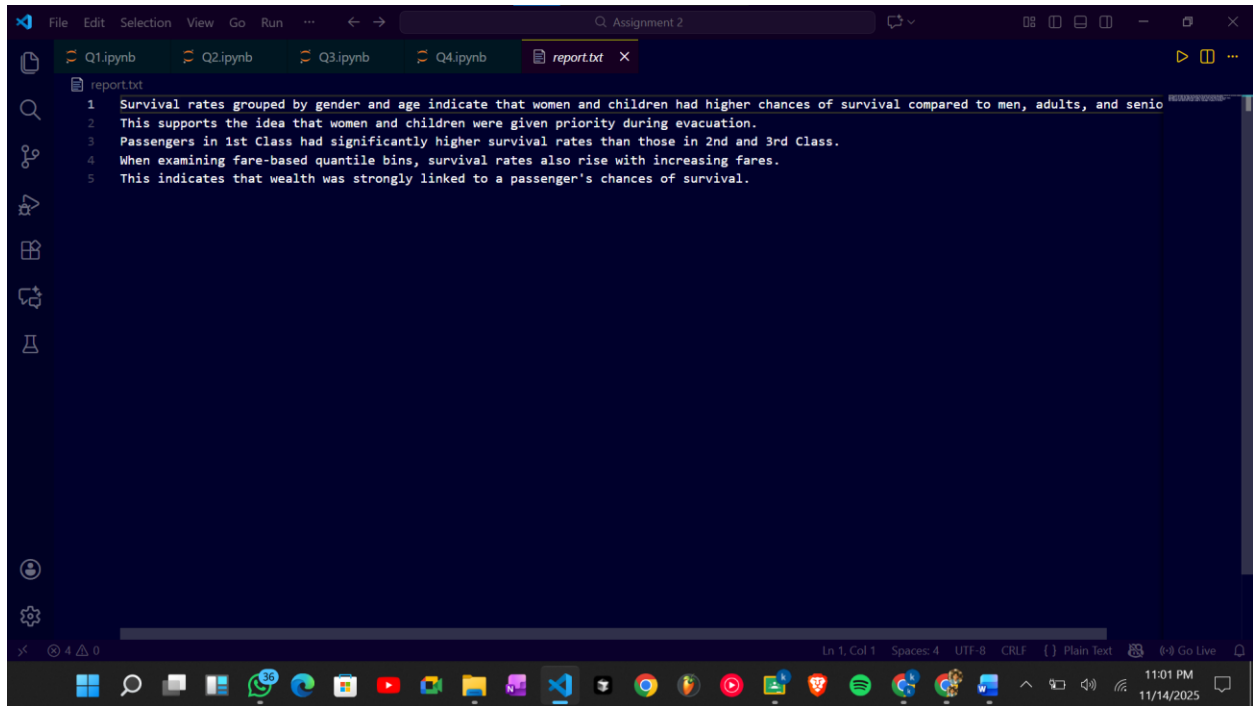
When examining fare-based quantile bins, survival rates also rise with increasing fares.

This indicates that wealth was strongly linked to a passenger's chances of survival."""

with open('report.txt','w') as file:

    file.write(paragraph1.strip() + "\n" + paragraph2.strip())
```

## Output:



The image shows a screenshot of a Visual Studio Code editor window. The title bar at the top reads "Assignment 2". The editor has several tabs open: "Q1.ipynb", "Q2.ipynb", "Q3.ipynb", "Q4.ipynb", and "report.txt". The "report.txt" tab is active, showing a file named "report.txt" with the following content:

```
1 Survival rates grouped by gender and age indicate that women and children had higher chances of survival compared to men, adults, and senio
2 This supports the idea that women and children were given priority during evacuation.
3 Passengers in 1st Class had significantly higher survival rates than those in 2nd and 3rd Class.
4 When examining fare-based quantile bins, survival rates also rise with increasing fares.
5 This indicates that wealth was strongly linked to a passenger's chances of survival.
```

The status bar at the bottom indicates "Ln 1, Col 1", "Spaces: 4", "UTF-8", "CRLF", and "Plain Text". The system tray at the bottom shows the date and time as "11:01 PM 11/14/2025".