Data Collection (Epic 1)

There are many popular open sources for collecting the data. Eg: kaggle.com, UCI repository, etc.

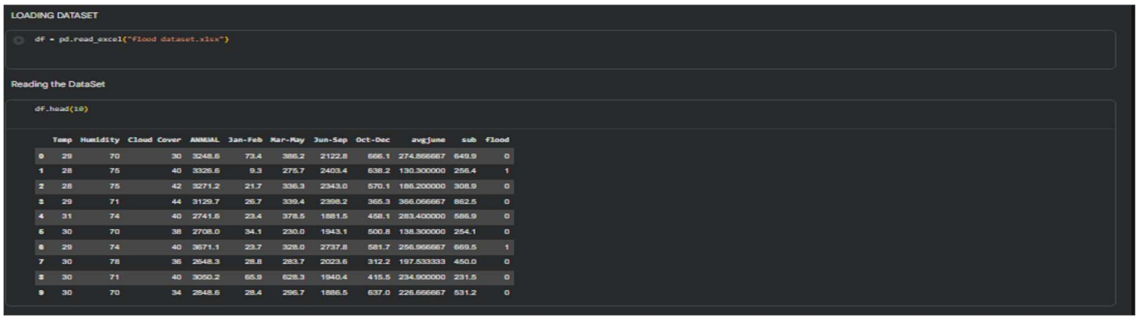
In this project we have used Floods_Data.csv data. This data is downloaded from kaggle.com. Please refer to the link given below to download the dataset.

Link: https://www.kaggle.com/datasets/arbethi/rainfall-dataset

Loading Data(USN2) 1

Our dataset format might be in .csv, excel files, .txt, .json, etc. We can read the dataset with the help of pandas.

In pandas we have a function called read_csv() to read the dataset. As a parameter we have to give the directory of csv file.



	Temp	Humidity	Cloud Cover	ANNUAL	Jan-Feb	Mar-May	Jun-Sep	Oct-Dec	avgjune	sub	flood
0	29	70	30	3248.6	73.4	386.2	2122.8	666.1	274.866667	649.9	0
1	28	75	40	3326.6	9.3	275.7	2403.4	638.2	130.300000	256.4	1
2	28	75	42	3271.2	21.7	336.3	2343.0	570.1	186.200000	308.9	0
3	29	71	44	3129.7	26.7	339.4	2398.2	365.3	366.066667	862.5	0
4	31	74	40	2741.6	23.4	378.5	1881.5	458.1	283.400000	586.9	0
5	30	70	38	2708.0	34.1	230.0	1943.1	500.8	138.300000	254.1	0
6	29	74	40	3671.1	23.7	328.0	2737.8	581.7	256.966667	669.5	1
7	30	78	36	2648.3	28.8	283.7	2023.6	312.2	197.533333	450.0	0
8	30	71	40	3050.2	65.9	628.3	1940.4	415.5	234.900000	231.5	0
9	30	70	34	2848.6	28.4	296.7	1886.5	637.0	226.666667	531.2	0

Data Preparation (Epic 2)

Handling Missing Values(USN3) 3

- For checking the null values, df.isnull() function is used. To sum those null values we use .sum() function to it. From the below image we found that there are no null values present in our dataset. So we can skip handling of missing values step.

```
#checking null values
dataset.isnull().any()
```

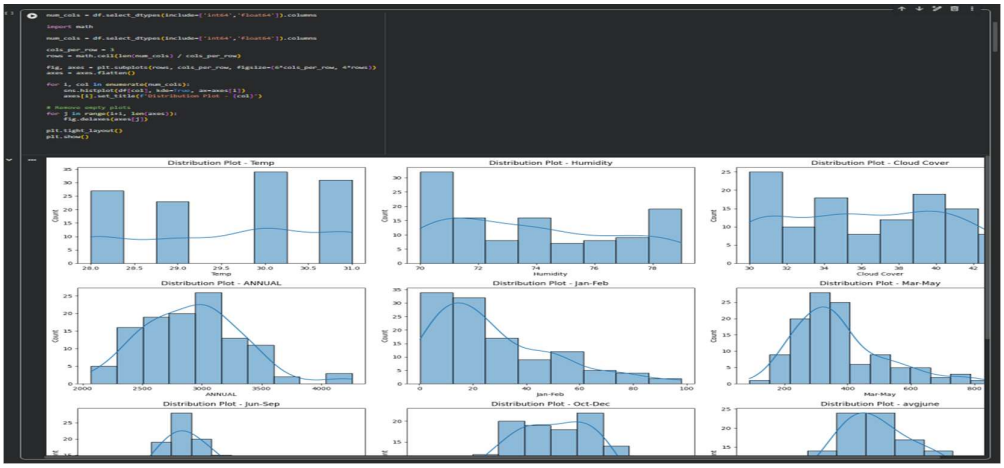
```
Temp           False
Humidity       False
Cloud Cover    False
ANNUAL         False
Jan-Feb        False
Mar-May        False
Jun-Sep        False
Oct-Dec        False
avgjune        False
sub            False
flood          False
dtype: bool
```

From the above code of analysis, we can infer that columns such as newbalanceOrg, oldbalnceOrig,isFraud are having the missing values, we need to treat them in a required way.

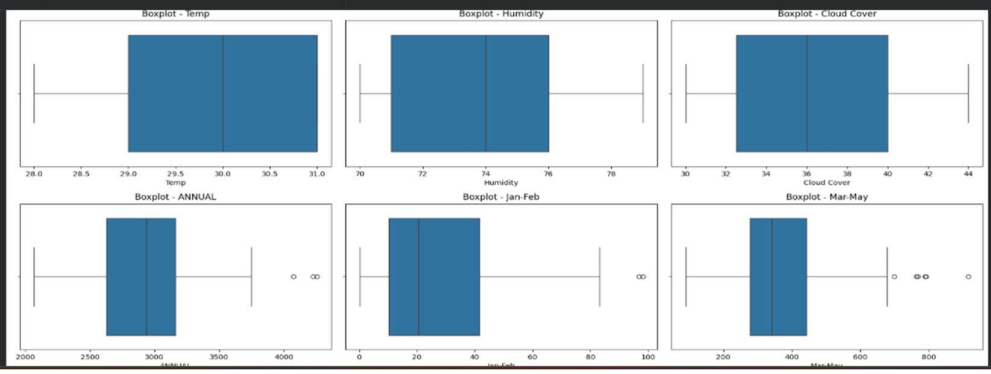
Total Story Point in Sprint 1= 2+1+3+3+3=12

Sprint 2

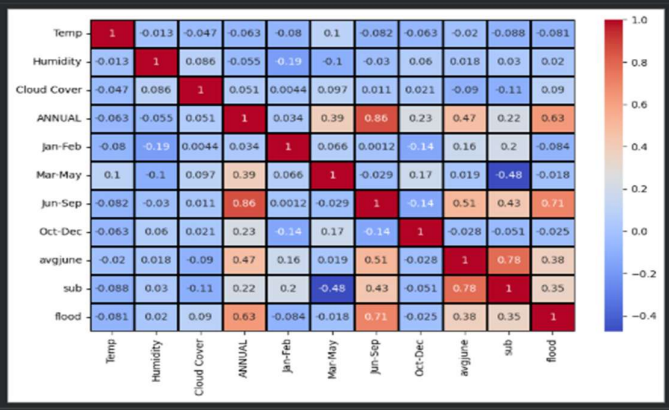
Data Visualization (Epic 3)



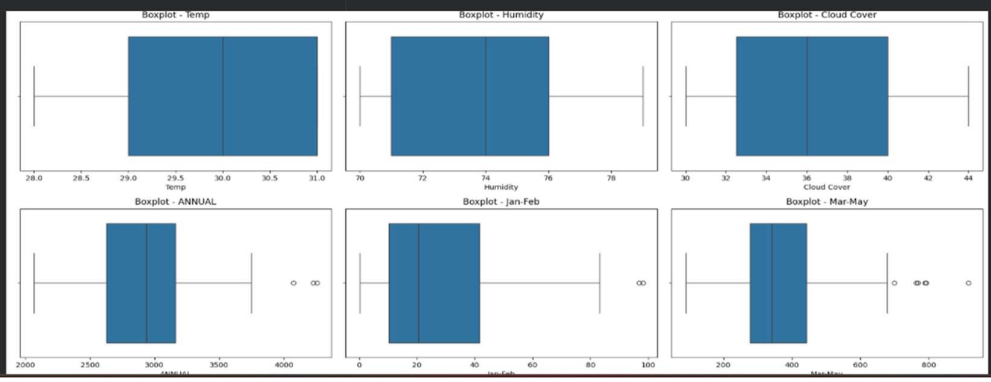
```
import math

num_cols = df.select_dtypes(include=['int64','float64']).columns

cols_per_row = 3
rows = math.ceil(len(num_cols) / cols_per_row)

fig, axes = plt.subplots(rows, cols_per_row, figsize=(6*cols_per_row, 4*rows))
axes = axes.flatten()

for i, col in enumerate(num_cols):
    sns.boxplot(y=df[col], ax=axes[i])
    axes[i].set_title(f'Boxplot - {col}')

# Remove unused axes
for j in range(i+1, len(axes)):
    fig.delaxes(axes[j])

plt.tight_layout()
plt.show()
```



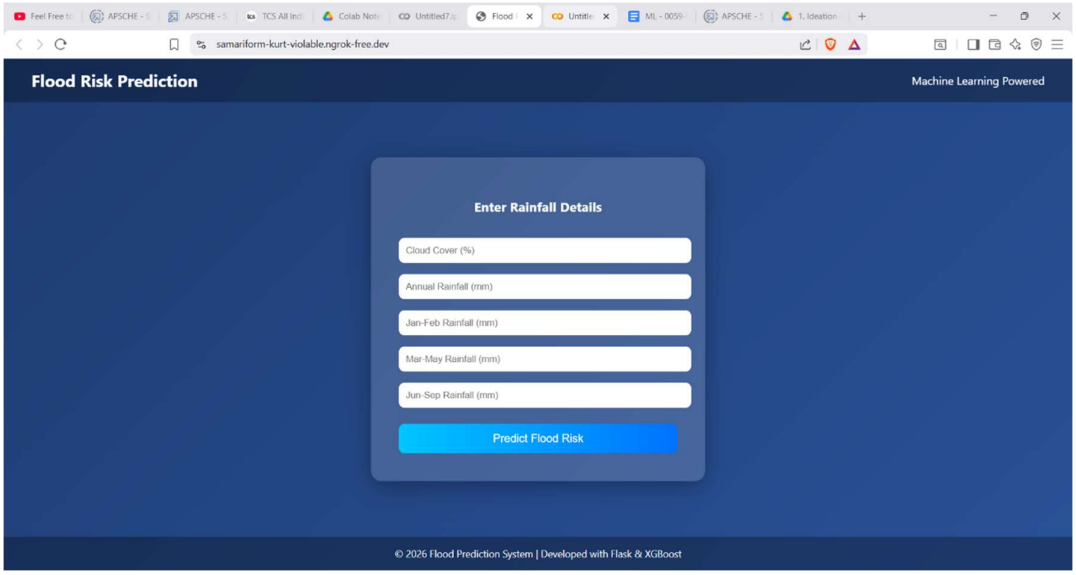

```
plt.figure(figsize=(10,6))
sns.heatmap(df.corr(), annot=True, cmap="coolwarm",
            linewidths=1, linecolor='k')
plt.show()
```




```
import math

num_cols = df.select_dtypes(include=['int64','float64']).columns

cols_per_row = 3
rows = math.ceil(len(num_cols) / cols_per_row)

fig, axes = plt.subplots(rows, cols_per_row, figsize=(6*cols_per_row, 4*rows))
axes = axes.flatten()

for i, col in enumerate(num_cols):
    sns.boxplot(y=df[col], ax=axes[i])
    axes[i].set_title(f'Boxplot - {col}')

# Remove unused axes
for j in range(i+1, len(axes)):
    fig.delaxes(axes[j])

plt.tight_layout()
plt.show()
```

output (Epic 4)



Story (Epic 5)