Flood Prediction Using IBM Watson

1.INTRODUCTION

1.1 OVERVIEW

Floods are inevitable, but with timely alerts, their effects can be minimized. There are a number of people who die every year due to devastating floods, the number of people become homeless and a number of people die due to lack of proper help after a flood. The lack of timely alerts has always been an issue concerning it. Delay in alerts in flood-prone areas is the biggest loophole of an economy. Conventional systems run a little low in forecasting floods at the right time so that proper actions could be taken before any disaster.

By using machine learning we can predict floods or forecast floods with better accuracy. This project aims at building predictive modeling based on the historical weather data of particular areas in order to predict the occurrence of floods. The predictive model is built on different machine learning algorithms. The concerned authority monitor this flood prediction system through a web application.

In this project, we will be using classification algorithms such as Decision tree, Random forest, KNN, and xgboost. We will train and test the data with these algorithms. From this best model is selected and saved in pkl format. We will be doing flask integration and IBM deployment.

1.2 PURPOSE

By using machine learning we can predict floods or forecast floods with better accuracy. This project aims at building predictive modeling based on the historical weather data of particular areas in order to predict the occurrence of floods. The predictive model is built on different machine learning algorithms. The concerned authority monitor this flood prediction system through a web application.

2 LITERATURE SURVEY

2.1 EXISTING PROBLEM

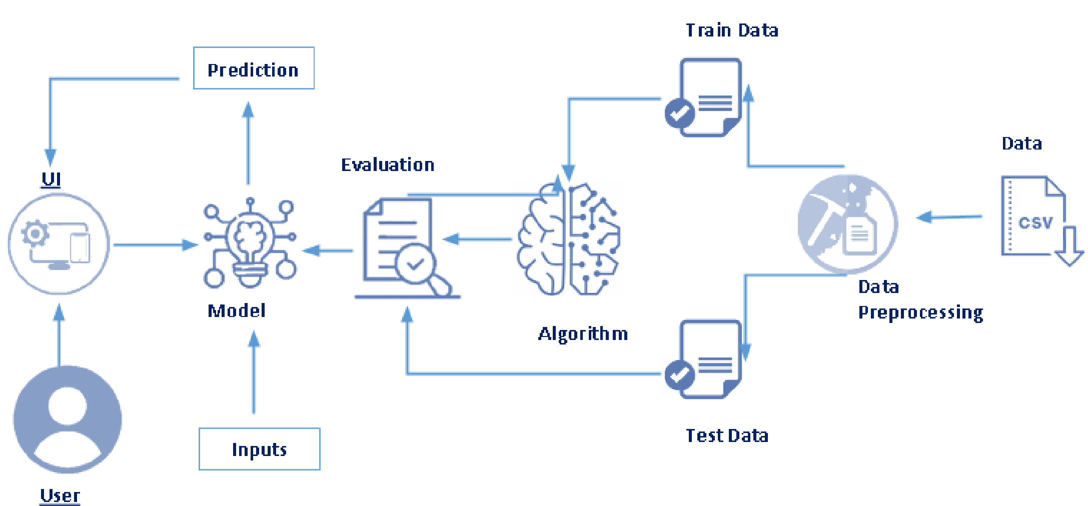
Floods are inevitable, but with timely alerts, their effects can be minimized. There are a number of people who die every year due to devastating floods, the number of people become homeless and a number of people die due to lack of proper help after a flood. The lack of timely alerts has always been an issue concerning it. Delay in alerts in flood-prone areas is the biggest loophole of an economy. Conventional systems run a little low in forecasting floods at the right time so that proper actions could be taken before any disaster.

2.2 PROPOSED SYSTEM

By using machine learning in this project we can predict floods or forecast floods with better accuracy. This project aims at building predictive modeling based on the historical weather data of particular areas in order to predict the occurrence of floods. The predictive model is built on different machine learning algorithms. The concerned authority monitor this flood prediction system through a web application.

3 THEORITICAL ANALYSIS

3.1 BLOCK DIAGRAM



3.2 HARDWARE/SOFTWARE DESIGNING

Software requirements:

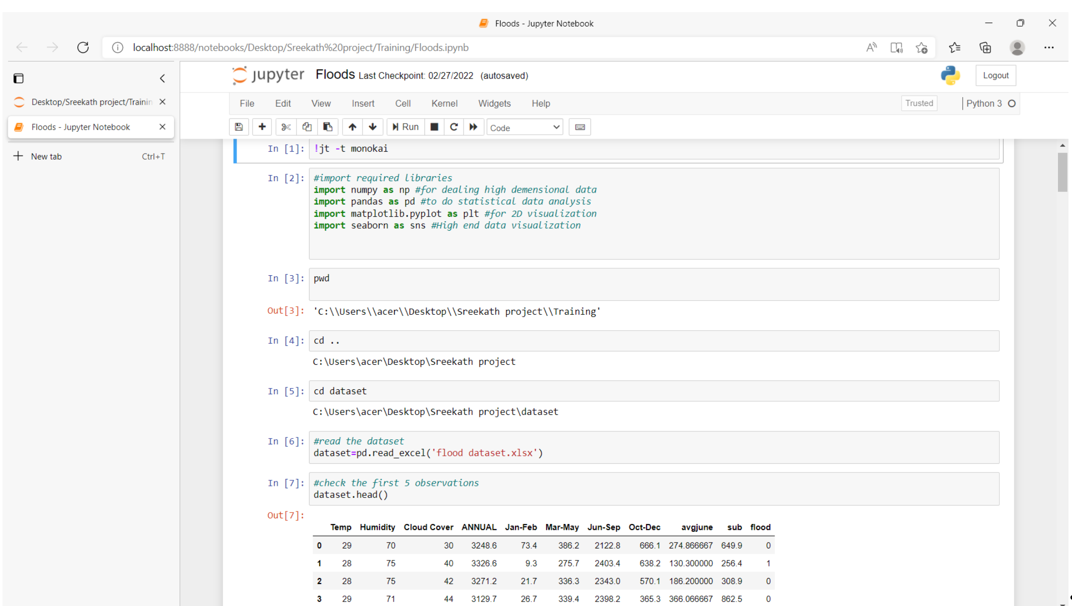
- Ananconda navigator
- Python packages
- IBM watson studio

4 EXPERIMENTAL INVESTIGATION

- Shape of inputs plays a major role in the correctness of the model.
- IBM Cloud helps to deploy machine learning models and test the correctness of our model.
- Integrating Flask with the machine learning model involves a lot of data preprocessing to make the predictions correctly.

5 RESULT

SOURCE CODE:



Floods - Jupyter Notebook

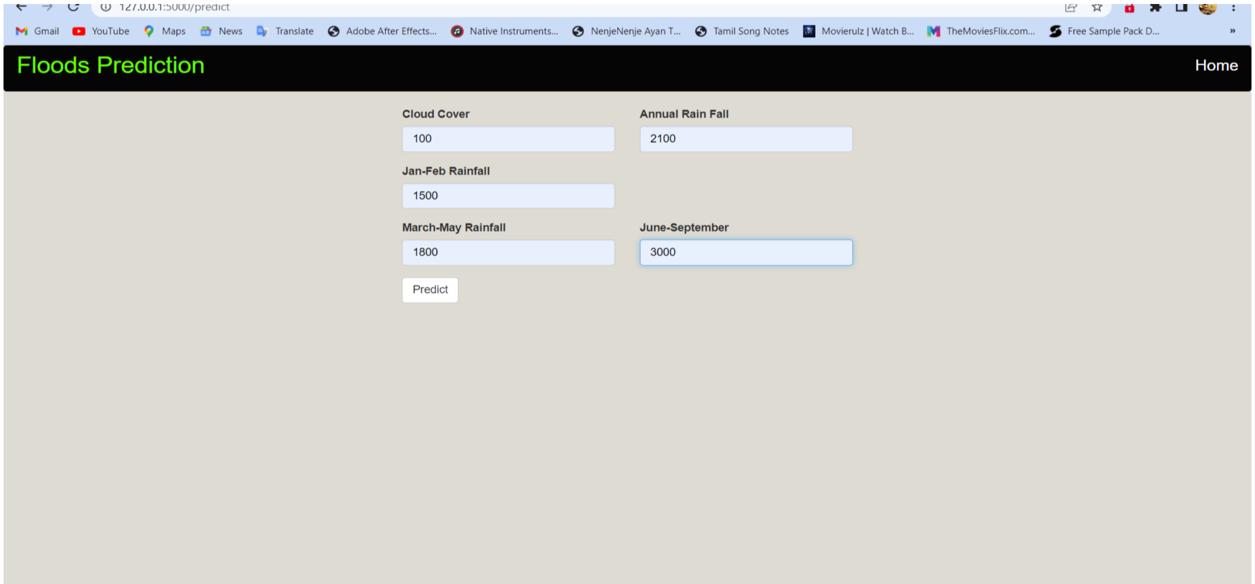
localhost:8888/notebooks/Desktop/Sreekath%20project/Training/Floods.ipynb

jupyter Floods Last Checkpoint: 02/27/2022 (autosaved)

```
In [1]: !jt -t monokai

In [2]: #import required libraries
        import numpy as np #for dealing high demensional data
        import pandas as pd #to do statistical data analysis
        import matplotlib.pyplot as plt #for 2D visualization
        import seaborn as sns #High end data visualization

In [3]: pwd

Out[3]: 'C:\\Users\\acer\\Desktop\\Sreekath project\\Training'

In [4]: cd ..

        C:\Users\acer\Desktop\Sreekath project

In [5]: cd dataset

        C:\Users\acer\Desktop\Sreekath project\dataset

In [6]: #read the dataset
        dataset=pd.read_excel('flood dataset.xlsx')

In [7]: #check the first 5 observations
        dataset.head()

Out[7]:
```

	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

AFTER ENTERING VALUES:



Here we got the prediction result as "No possibility of severe flood"

6 ADVANTAGES AND DISADVANTAGES

6.1 ADVANTAGES

1.Advancement of flood prediction models is essential for risk reduction, policy suggestion, minimization of the loss of human life, rapid flood mapping, and reduction of property damage associated with floods.

2.By predictind the flood we can ensure ,Even if the floods are small, small businesses and farmers are most affected. So we do flood prediction so that they can take precautionary measures and get help in advance

6.2 DISADVANTAGES:

1. If the value given is not accurate, it may lead to false prediction. Due to this false prediction it may mislead people and they may get panic.
2. Its completely based on previous datasets. It won't consider new environmental changes.

7 APPLICATION

1.In Research centers
2.Wheather forcasting
3.National hydrological services

8 CONCLUSION

Floods are without doubt the most devastating natural disasters, striking numerous regions in the world each year. During the last decades the trend in flood damages has been growing exponentially. This is a consequence of the increasing frequency of heavy rain, changes in upstream land-use and a continuously increasing concentration of population and assets in flood prone areas. In general, less developed countries are the most vulnerable to floods, causing damages that significantly affect the national GDP. At country and community levels important initiatives have and are being devoted to implement appropriate countermeasures, both structural and non-structural, aiming to alleviate the persistent threats of water-related disasters. Flood Forecasting (FF) forms an important tool in reducing vulnerabilities and flood risk and form an important ingredient of the strategy to "live with floods", thereby contributing to national sustainable development.

9 FUTURE SCOPE

For future work, conducting a survey on spatial flood prediction using machine learning models is highly encouraged. This important aspect of flood prediction was excluded from our project due to the nature of modeling methodologies and the datasets used in predicting the location of floods. Nevertheless, the recent advancements in machine learning models for spatial flood analysis revolutionized this particular realm of flood forecasting, which requires separate investigation.

10 BIBLIOGRAPHY

https://www.mdpi.com/2073-4441/10/11/1536

https://www.youtube.com/watch?v=1lTkzWGrrWY

https://www.google.co.in/books/edition/Flood_Forecasting_Using_Machine_Learning/mFeNDwAAQBAJ?hl=en&gbpv=1&printsec=frontcover

https://un-spider.org/links-and-resources/data-sources/daotm-floods-ml#:~:text=Advantages,-ML%20models%20have&text=Advancement%20of%20flood%20prediction%20models,property%20damage%20associated%20with%20floods

12 APPENDIX

```
!jt -t monokai
#import required libraries
import numpy as np #for dealing high demensional data
import pandas as pd #to do statistical data analysis
import matplotlib.pyplot as plt #for 2D visualization
import seaborn as sns #High end data visualization
pwd
cd ..
cd dataset
#read the dataset
dataset=pd.read_excel('flood dataset.xlsx')
#check the first 5 observations
dataset.head()
print(dataset.shape)
print(dataset.info())
dataset.columns
dataset.describe().T
#checking null values
dataset.isnull().any()
#Correlation
dataset.corr()
import seaborn as sns
fig=plt.gcf()
fig.set_size_inches(15,15)
fig=sns.heatmap(dataset.corr(),annot=True,cmap='summer',
            linewidths=1,linecolor='k',square=True,
            mask=False, vmin=-1, vmax=1,
            cbar_kws={"orientation": "vertical"},cbar=True)
dataset.drop(["Oct-Dec"],axis=1,inplace=True)
dataset.head()
dataset['flood'].value_counts()
#independent features
x=dataset.iloc[:,2:7].values
#dependent feature
y=dataset.iloc[:,9:].values
x.shape
y.shape
#split the data into train and test set from our x and y
```

```
#import train_test_split fucntion
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test = train_test_split(x,y,test_size=0.25,random_state=10)
#checking the shape of our 4 variables
print(x_train.shape)
print(x_test.shape)
print(y_train.shape)
print(y_test.shape)
#import Standardscaler
from sklearn.preprocessing import StandardScaler
#create object to Standardscaler class
sc=StandardScaler()
x_train=sc.fit_transform(x_train)
x_test=sc.fit_transform(x_test)
#import dump class from joblib
from joblib import dump
dump(sc,"transform.save")
#pip install xgboost
#import xgboost
import xgboost as xgb
#hyper parameter tuning to xgboost
xg_cla = xgb.XGBClassifier(objective ='reg:linear',learning_rate = 0.1,
                max_depth = 5, n_estimators = 10)
#fit the model
xg_cla.fit(x_train,y_train)
#predictions with unseen data by model
y_pred_xgb = xg_cla.predict(x_test)
y_pred_xgb
#checking the accuracy score
from sklearn.metrics import accuracy_score,confusion_matrix
acc=accuracy_score(y_test,y_pred_xgb)
acc
#summary of predictions
cm2=confusion_matrix(y_test,y_pred_xgb)
cm2
y_pred_xgb.shape
from sklearn.metrics import mean_squared_error
rmse = np.sqrt(mean_squared_error(y_test, y_pred_xgb))
print("RMSE: %f" % (rmse))
dataset.head()
rand_pred = xg_cla.predict(sc.transform([[30,3248.6,73.4,386.2,2122.8]]))
rand_pred
rand_pred1 = xg_cla.predict(sc.transform([[40,3326.6,9.3,275.7,2403.4]]))
rand_pred1
data_dmatrix = xgb.DMatrix(data=x,label=y)
```

```
params = {"objective":"reg:linear",'colsample_bytree': 0.3,'learning_rate': 0.1,
            'max_depth': 5, 'alpha': 10}

cv_results = xgb.cv(dtrain=data_dmatrix, params=params, nfold=3,
                num_boost_round=50,early_stopping_rounds=10,metrics="rmse", as_pandas=True, seed=123)
cv_results.head()
xg_cla_model = xgb.train(params=params, dtrain=data_dmatrix, num_boost_round=10)

#import matplotlib.pyplot as plt

#xgb.plot_tree(xg_cla_model,num_trees=0)
#plt.rcParams['figure.figsize'] = [15,15]
#plt.show()

from sklearn.tree import DecisionTreeClassifier
dtc = DecisionTreeClassifier()
dtc.fit(x_train,y_train)

y_predict = dtc.predict(x_test)
y_predict

from sklearn.metrics import accuracy_score,confusion_matrix

acc=accuracy_score(y_test,y_predict)
acc

acc=accuracy_score(y_test,y_pred_xgb)
acc

cm1=confusion_matrix(y_test,y_predict)
cm1

cm2=confusion_matrix(y_test,y_pred_xgb)
cm2

#saving the file
from joblib import dump
dump(xg_cla,'floods.save')
```