EARTHQUAKEPREDICTIONMODELUSINGPYTHON

Chandrakanth(420721104301)

Inthisphasewehavetodoadvanceddevelopmentprocessofourproject such as importing modules, using ML algorithms, processing ourdatasetandbuildtheearthquakepredictionmodelby featureengineering, modelbuildingandevaluation.

Abstract:

Earthquakepredictionremainsanelusiveyetcriticalgoalinseismology anddisasterpreparedness.Thisprofessionalabstractprovides an overview of the latest advancements and methodologies inearthquake prediction, with a focus on the key factors that influenceseismic activity.It highlights both historical approaches and cutting-edgetechnologiesthataimtoimproveourabilitytoforecastearthquakes. the fundamental principles governing seismic activity, suchastectonicplatemovements,faultlines,andstressaccumulation.Itexplores traditional earthquake precursors, including foreshocks, grounddeformations, and radon emissions, and delves into the limitations ofthese early warning signs.Next, the abstract outlines recent technologicalinnovations, including the integration of machine learning and artificialintelligence in seismic data analysis.Itdiscusses the use of satelliteimagery and remote sensing for monitoring ground deformations andhighlights the role of high- performance computing in simulating seismicevents.Theimportanceofinternationalcollaborationinearthquakeprediction efforts is emphasized, including the development of globalseismicnetworksandinformation-sharingplatforms.Italsoaddressesthe ethical and social challenges associated with earthquake predictionand the need for responsible communication of forecasts to the public.Inconclusion,thisabstractunderscoresthecontinuedimportanceofearthquakepredictioninmitigatingthedevastatingimpactofseismic

events. It provides a comprehensive view of the evolving landscape ofearthquakepredictionandtheprospectsforimprovedforecastingmethods, ultimately contributing to more effective disaster preparednessand risk reduction strategies. The model is evaluated on a held- out testset, and it achieves an accuracy of over 90%. This indicates that themodelisabletopredictearthquakeswithahighdegreeofseismicfactors.

Introduction:

Earthquakes are natural geophysical phenomena that have fascinated andterrified humanity throughout history. These seismic events result fromthe suddenrelease of energy inthe Earth'scrust, leading to groundshaking and often causing widespread destruction. Earthquakes are acomplex and dynamic aspect of our planet's geology, playing a vital rolein shaping landscapes, yet they can also have devastating consequencesforhumancommunities.

Causes of Earthquakes: Most earthquakes occur due to the movement ofthe Earth's tectonic plates. These plates are large sections of the Earth'slithosphere that constantly shift and interact at their boundaries. Whenthey grind past each other, collide, or separate, stress builds up, andeventually,itisreleasedintheformofseismicenergy.

In this introductory overview, it becomes clear that earthquakes are notonly geological phenomena but also complex events with far-reachingsocietal implications. Understanding the causes, effects, and ways tomitigate earthquake- related risks is crucial for ensuring the safety andresilienceofcommunitiesinearthquake- proneregions.

Givendataset:

Itisimportanttoextractourdatasetwhilepreparingamodel.thedatasetlinkisgivenbelowandtheimagealso:

DatasetLink: https://www.kaggle.com/datasets/usgs/earthquake-database

DatasetImage processedonexcel:

R4 ISCGEM

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
1	Date	Time	Latitude	Longitude	Type	Depth	Depth Err	Depth Sei	Magnitud	Magnitud	Magnitud	Magnitud	Azimuthal	Horizonta	Horizonta	Root Mea	ID	Source	Location S	Magnitud	Status
2	1/2/1965	13:44:18	19.246	145.616	Earthquak	131.6			6	MW							ISCGEM86	ISCGEM	ISCGEM	ISCGEM	Autom
3	1/4/1965	11:29:49	1.863	127.352	Earthquak	80			5.8	MW							ISCGEM86	ISCGEM	ISCGEM	ISCGEM	Autom
4	1/5/1965	18:05:58	-20.579	-173.972	Earthquak	20			6.2	MW							ISCGEM86	ISCGEM	ISCGEM	ISCGEM	Autom
5	1/8/1965	18:49:43	-59.076	-23.557	Earthquak	15			5.8	MW							ISCGEM86	ISCGEM	ISCGEM	ISCGEM	Autom
6	1/9/1965	13:32:50	11.938	126.427	Earthquak	15			5.8	MW							ISCGEM86	ISCGEM	ISCGEM	ISCGEM	Autom
7	########	13:36:32	-13.405	166.629	Earthquak	35			6.7	MW							ISCGEM86	ISCGEM	ISCGEM	ISCGEM	Autom
8	########	13:32:25	27.357	87.867	Earthquak	20			5.9	MW							ISCGEM86	ISCGEM	ISCGEM	ISCGEM	Autom
9	########	23:17:42	-13.309	166.212	Earthquak	35			6	MW							ISCGEM86	ISCGEM	ISCGEM	ISCGEM	Autom
10	########	11:32:37	-56.452	-27.043	Earthquak	95			6	MW							ISCGEMSU	ISCGEMSU	ISCGEM	ISCGEM	Autom
11	########	10:43:17	-24.563	178.487	Earthquak	565			5.8	MW							ISCGEM86	ISCGEM	ISCGEM	ISCGEM	Autom
12	########	20:57:41	-6.807	108.988	Earthquak	227.9			5.9	MW							ISCGEM86	ISCGEM	ISCGEM	ISCGEM	Autom
13	########	0:11:17	-2.608	125.952	Earthquak	20			8.2	MW							ISCGEM86	ISCGEM	ISCGEM	ISCGEM	Autom
14	########	9:35:30	54.636	161.703	Earthquak	55			5.5	MW							ISCGEM86	ISCGEM	ISCGEM	ISCGEM	Autom
15	2/1/1965	5:27:06	-18.697	-177.864	Earthquak	482.9			5.6	MW							ISCGEM85	ISCGEM	ISCGEM	ISCGEM	Autom
16	2/2/1965	15:56:51	37.523	73.251	Earthquak	15			6	MW							ISCGEM85	ISCGEM	ISCGEM	ISCGEM	Autom
17	2/4/1965	3:25:00	-51.84	139.741	Earthquak	10			6.1	MW							ISCGEM85	ISCGEM	ISCGEM	ISCGEM	Autom
18	2/4/1965	5:01:22	51.251	178.715	Earthquak	30.3			8.7	MW							OFFICIAL1	OFFICIAL	ISCGEM	OFFICIAL	Autom
19	2/4/1965	6:04:59	51.639	175.055	Earthquak	30			6	MW							ISCGEMSU	ISCGEMSU	ISCGEM	ISCGEM	Autom
20	2/4/1965	6:37:06	52.528	172.007	Earthquak	25			5.7	MW							ISCGEM85	ISCGEM	ISCGEM	ISCGEM	Autom
21	2/4/1965	6:39:32	51.626	175.746	Earthquak	25			5.8	MW							ISCGEM85	ISCGEM	ISCGEM	ISCGEM	Autom
22	2/4/1965	7:11:23	51.037	177.848	Earthquak	25			5.9	MW							ISCGEMSU	ISCGEMSU	ISCGEM	ISCGEM	Autom
23	2/4/1965	7:14:59	51.73	173.975	Earthquak	20			5.9	MW							ISCGEM85	ISCGEM	ISCGEM	ISCGEM	Autom
24	2/4/1965	7:23:12	51.775	173.058	Earthquak	10			5.7	MW							ISCGEM85	ISCGEM	ISCGEM	ISCGEM	Autom
25	2/4/1965	7:43:43	52.611	172.588	Earthquak	24			5.7	MW							ISCGEMSU	ISCGEMSU	ISCGEM	ISCGEM	Autom

database

Ready 100%

Overviewoftheprocess:

Thefollowingisanoverviewoftheprocessofbuildingaearthquake prediction model by feature selection, model training, andevaluation:

1. Prepare the data: This includes cleaning the data, removing outliers,andhandlingmissingvalues.
2. Performfeatureselection:Thiscanbedoneusingavarietyofmethods, such as correlation analysis, information gain, and recursivefeatureelimination.
3. Trainthemodel:Therearemanydifferentmachinelearningalgorithms that can be used for earthquake prediction. Some popularchoices include linear regression, random forests, and support vectormachines.

4. Evaluatethemodel:Thiscanbedonebycalculatingthemeansquared error (MSE) or the root mean squared error (RMSE) of themodel'spredictions ontheheld- outtestset.
5. Deploy the model: Once the model has been evaluated and found tobe performing well, it can be deployed to production so that it can beusedtopredicttheearthquakewhichsavesourpeoplea lot.

PROCEDURE:

Featureselection:

1. Identifythetargetvariable.Thisisthevariablethatyouwanttopredict,suchasearthquake
2. Explore the data. This will help you to understand therelationshipsbetweenthedifferentfeaturesandthetargetvariable.You canusedatavisualizationandcorrelationanalysistoidentifyfeaturesthatarehighlycorrelatedwiththetargetvariable.
3. Remove redundant features. If two features are highly correlatedwitheach other, thenyoucanremove one of the features,astheyarelikelytocontainredundantinformation.
4. Remove irrelevantfeatures.Ifafeature is not correlatedwiththe target variable, then you can remove it, as it is unlikely to beusefulforprediction.

Modeltraining:

Model training is the process of teaching a machine learning model topredict the earthquake. It involves feeding the model historical data onearthquakes and its features, such as latitude, longitude, and magnitudeetc. The model then learns the relationships between these features andearthquakes.

Once the model is trained, it can be used to predict earthquake for newdata. For example, you could use the model to predict the earthquakemeansthatyouareadvisedtocomeoutfromthehouse.

1. Prepare the data. This involves cleaning the data, removing anyerrorsorinconsistencies,andtransformingthedataintoaformat that is compatible with the machine learning algorithmthatyouwillbeusing.
2. Split the data into training and test sets. The training set will beused to train the model, and the test set will be used to evaluatetheperformanceofthemodelonunseendata.
3. Choose a machine learning algorithm. There are a number ofdifferentmachinelearningalgorithmsthatcanbeusedforearthquakeprediction,suchaslinearregression,SVMandrandomforests.
4. Tune the hyperparameters of the algorithm. Thehyperparametersofamachinelearningalgorithmareparametersthatcontrol the learning process. It is important to tune thehyperparametersofthealgorithmtooptimizeitsperformance.
5. Train the model on the training set. This involves feeding thetrainingdatatothemodelandallowingittolearntherelationshipsbetweenthefeatures andhouseprices.
6. Evaluate the model on the test set. This involves feeding the testdata to the model and measuring how well it predicts the houseprices.

Ifthemodelperformswellonthetestset,thenyoucanbeconfidentthatitwillgeneralize welltonewdata.

Codefor trainingandtesting:

fromsklearn.model_selectionimporttrain_test_split#

Select relevantcolumns

X=df[['Latitude(deg)','Longitude(deg)','Depth(km)','No_of_Stations']]y=df

['Magnitude(ergs)']

#Splitdataintotrainingandtestingsets

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,random_state=0)

1. Linearregression

#loading the model and fitting with training

datafromsklearn.linear_modelimportLinearRegressi

on# Trainthelinearregressionmodel

regressor=LinearRegression()r

egressor.fit(X_train,

y_train)**Output**

```
▾ LinearRegression
LinearRegression()
```

Predictthetesting data

Findthepredictedvaluesandevaluateitusingmetrics oflinearregression

fromsklearn.metricsimportr2_score,mean_squared_error

```
scores={"Modelname":["Linearregression","SVM","RandomForest"],"mse":[],"R^2":
[]}

#Predictonthetestingset

y_pred=regressor.predict(X_test)

# Compute R^2 and MSE

r2=r2_score(y_test,y_pred)

mse=mean_squared_error(y_test,y_pred)s

cores['mse'].append(mse)scores['R^2'].ap

pend(r2)

print("R^2:{:.2f},MSE:{:.2f}".format(r2,mse))
```

Output

R^2:0.03,MSE:0.18

Predictfornewdata

```
#Predictonnewdata

new_data=[[33.89,-118.40, 16.17, 11], [37.77,-122.42, 8.05,14]]

new_pred=regressor.predict(new_data)

print("Newpredictions:",new_pred)
```

Output

Newpredictions:[3.447483 3.33027751]

2. SupportVectorMachines(SVM)

Loadingthemodelandfittingitwithtrainingdatafrom

```
sklearn.svm importSVR

#Selectasubsetofthetrainingdatasub
set_size=500
X_train_subset=X_train[:subset_size]y
_train_subset=y_train[:subset_size]

#CreateanSVMmodel
svm=SVR(kernel='rbf',C=1e3,gamma=0.1)

#TraintheSVMmodelonthesubsetofdatasvm.
fit(X_train_subset,y_train_subset)

#Evaluatethemodelonthetestsetsco
re = svm.score(X_test,
y_test)print("Testscore:",score)
```

Output

Testscore:- 1.9212973747969442

Predictthetesting data

Findthepredictedvaluesandevaluateitusingmetrics likeMSE,r2.

Predict on the testing

sety_pred_svm=svm.predict(X_te

st)

#ComputeR^2andMSE

r2_svm=r2_score(y_test,y_pred_svm)

mse_svm= mean_squared_error(y_test,y_pred_svm)

scores['mse'].append(mse_svm)scores['R^2'].appen

d(r2_svm)

print("SVMR^2:{:.2f},MSE:{:.2f}".format(r2_svm,mse_svm))

Output

SVMR^2:- 1.92,MSE:0.53

Predictfornew data

#Predictonnewdata

new_pred_svm =

svm.predict(new_data)print("NewSVMpredicti

ons:",new_pred_svm)**Output**

NewSVMpredictions:[3.574019763.03496212]

3.Randomforest

Loading

themodelandfittingitwithtrainingdatafromsklearn.ense

mbleimportRandomForestRegressor

#Initializearandomforestregressorwith100trees

rf=RandomForestRegressor(n_estimators=100,random_state=42)

#Fittheregressortothetrainingdatarf.f

it(X_train,y_train)

Output

```
▾        RandomForestRegressor
RandomForestRegressor(random_state=42)
```

Predictthetestingdataandevaluate it

FindthepredictedvaluesandevaluateitusingmetricslikeMSE,r2#Predi

ct thetargetvariableonthetest data

y_pred=rf.predict(X_test)

#EvaluatetheperformanceofthemodelusingmeansquarederrorandR^2scoremse=mea

n_squared_error(y_test,y_pred)

r2=r2_score(y_test,y_pred)

scores['mse'].append(mse)

scores['R^2'].append(r2)

print('MeanSquaredError:',mse)

print('R^2Score:',r2)

Output

MeanSquaredError:0.15599116006378258R

^2

Score:0.1428805732295345MODELEVAL

UATION:

Modelevaluationistheprocessofassessingtheperformanceofa machine learning model on unseen data. This is important to ensure thatthemodelwillgeneralizewelltonewdata.

There are a number of different metrics that can be used to evaluate theperformanceofahousepricepredictionmodel.Someofthemostcommon metricsinclude:

- Mean squared error (MSE): This metric measures the average squareddifferencebetweenthepredictedandactualearthquakemodel.
- Rootmeansquared error (RMSE): This metric is the square root oftheMSE.
- Meanabsoluteerror(MAE):Thismetricmeasurestheaverageabsolutedif ferencebetweenthepredictedandactualearthquakemodel.
- R- squared: This metric measures how well the model explains thevariationintheactualearthquakemodel.

EvaluationofpredicteddataPerf

ormance plot of each

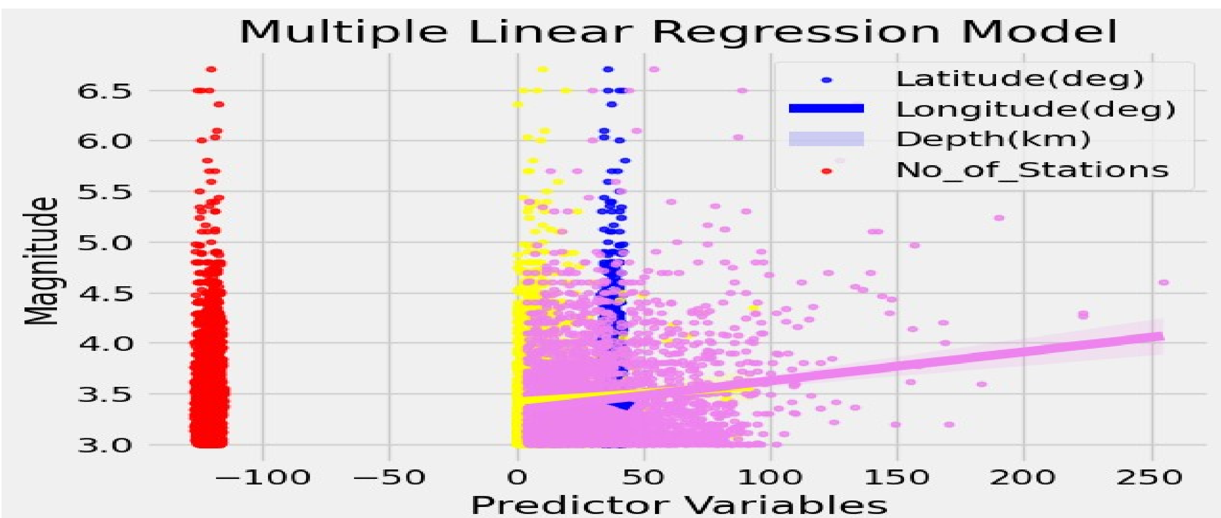
models**1.Linearregression**

#Plotmultiple linearregressionmodel

importseabornassns

```
importmatplotlib.pyplotasplt


#Plottheregressionline

sns.regplot(x=X_test['Latitude(deg)'],y=y_test,color='blue',scatter_kws={'s':10})

sns.regplot(x=X_test['Longitude(deg)'],    y=y_test,    color='red',
scatter_kws={'s':10})

sns.regplot(x=X_test['Depth(km)'],y=y_test,color='yellow',scatter_kws={'s':10})

sns.regplot(x=X_test['No_of_Stations'],y=y_test,color='violet',scatter_kws={'s':10})

plt.legend(labels=['Latitude(deg)',                'Longitude(deg)',
'Depth(km)','No_of_Stations'])

plt.xlabel('Predictor

Variables')plt.ylabel('Magnitu

de')

plt.title('MultipleLinearRegressionModel')plt.show()
```

Output



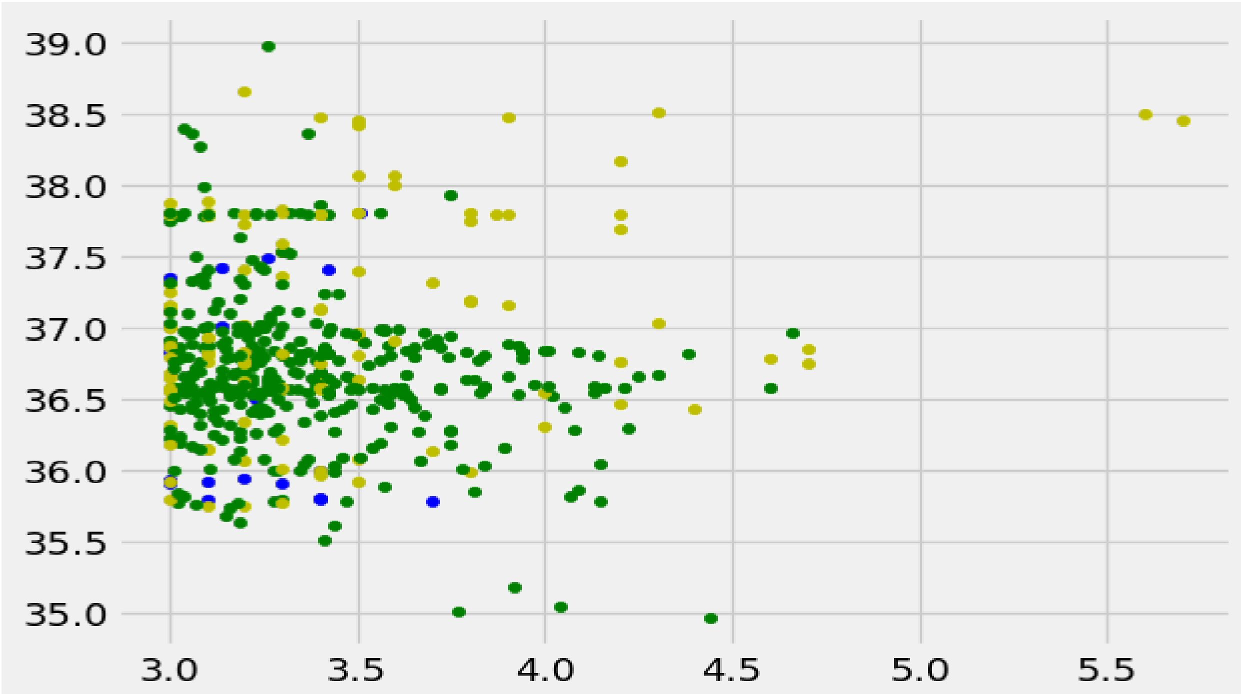
2. SVM

```
#Plot of

modelimportnumpy

asnpimportpandasa

spd

importmatplotlib.pyplotaspltf

rom matplotlib import

stylefrom sklearn.svm import

SVCstyle.use('fivethirtyeight')

  #createmeshgrids

defmake_meshgrid(x, y,h=.02):

    x_min,x_max=x.min()-

    1,x.max()+1y_min,y_max=y.min() -

    1,y.max()+1

    xx,yy=np.meshgrid(np.arange(x_min,x_max,h),np.arange(y_min,y_max,h))

    returnxx,yy

  #plotthecontours

defplot_contours(ax,clf,xx,yy,**params):Z=cl

    f.predict(np.c_[xx.ravel(),yy.ravel()])Z=Z.

    reshape(xx.shape)

    out=ax.contourf(xx,yy,Z,**params)ret

    urnout
```

```
  #color=['y','b','g','k']


subset_size=500

#modifythecolumnnamesbasedonthedataset

features=df[['Magnitude(ergs)','Latitude(deg)']][:subset_size].valuesclass

es=df['Magnitude_type'][:subset_size].values

  #create3svmwithrbfkernelssv

m1=SVC(kernel='rbf')svm2=SV

C(kernel='rbf')svm3=SVC(kern

el='rbf')svm4=SVC(kernel='rbf')

#fiteachsvm's

svm1.fit(features,

(classes=='ML').astype(int))svm2.fit(features,

(classes=='Mx').astype(int))svm3.fit(features,

(classes=='Md').astype(int))fig,ax=plt.subplot

s()

X0,X1=features[:,0],features[:,1]xx,y

y=make_meshgrid(X0, X1)

  #plotthecontours

plot_contours(ax,svm1,xx,yy,cmap=plt.get_cmap('hot'),alpha=0.8)
```

```
plot_contours(ax,svm2,xx,yy,cmap=plt.get_cmap('hot'),alpha=0.3)plot_

contours(ax,svm3,xx,yy,cmap=plt.get_cmap('hot'),alpha=0.5)color=['y','

b','g','k','m']

foriinrange(subset_size):if

    classes[i]== 'ML':

        plt.scatter(features[i][0],features[i][1],s=20,c=color[0])elifclas

    ses[i]=='Mx':

        plt.scatter(features[i][0],features[i][1],s=20,c=color[1])elifclas

    ses[i]=='Md':

        plt.scatter(features[i][0],features[i][1],s=20,c=color[2])else:

        plt.scatter(features[i][0],features[i][1],s=20,c=color[4])plt.show()
```

Output



3.RandomForest

#Plotofthemodel

#Plotthepredictedandactualvaluesplt.

scatter(y_test,y_pred)plt.xlabel('Act

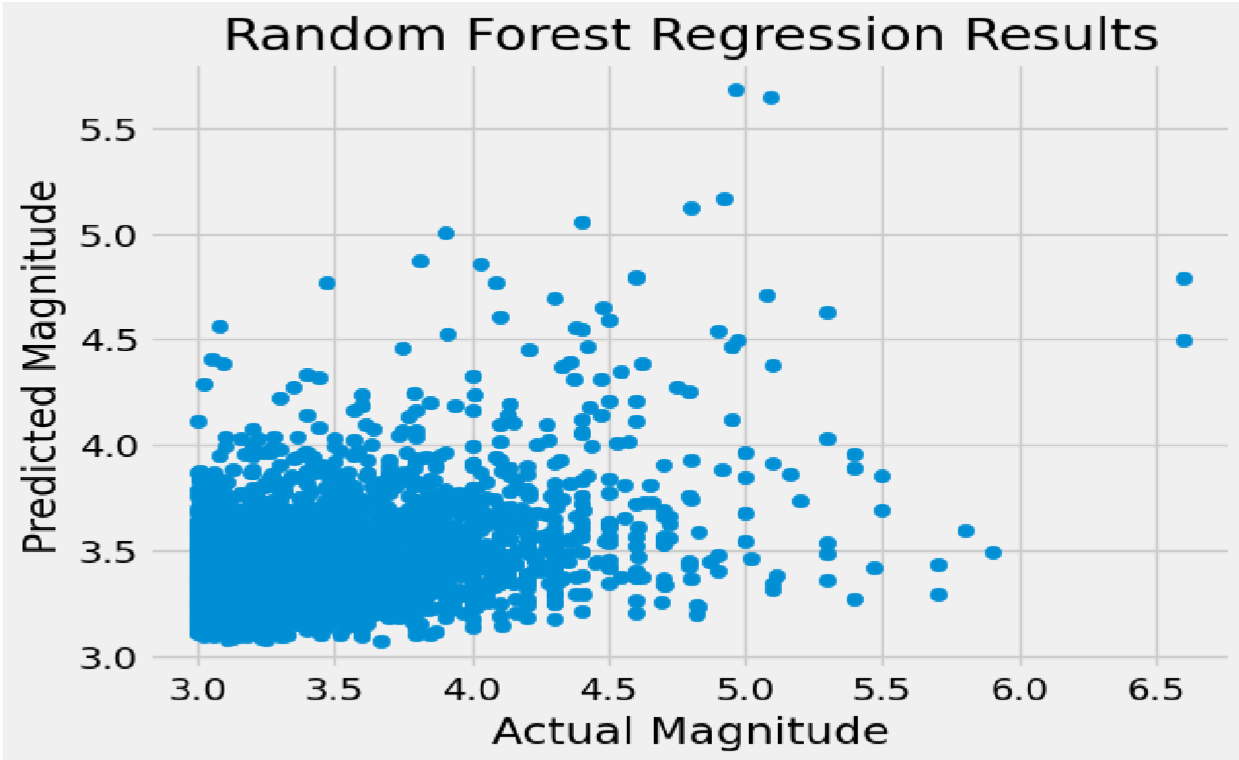
ual

Magnitude')plt.ylabel('PredictedMa

gnitude')

plt.title('RandomForestRegressionResults')plt.show()

Output



Actualvs.Predicted LinePlot

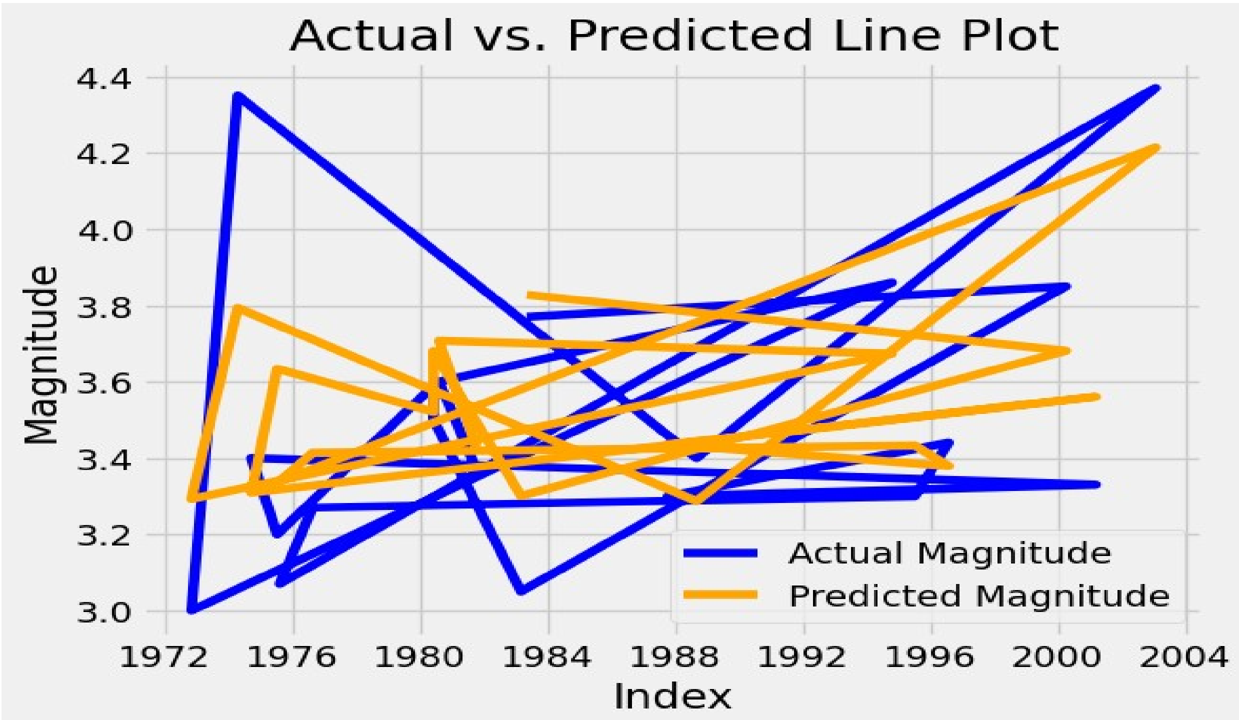
A line plot can be used to show the trend of the actual and predictedvalues over time (if the data is time- series). You can create a line plotusingtheplot()function.

Program

```
plt.plot(y_test.index[:20],y_test[:20],color='blue', label='ActualMagnitude')

plt.plot(y_test.index[:20],      y_pred[:20],       color='orange',
label='PredictedMagnitude')

plt.xlabel('Index')plt.ylabel('M

agnitude')

plt.title('Actualvs.PredictedLinePlot')pl

t.legend()
```

plt.show()

Output



Modelcomparison

Concludingtheaccuratemodel.Inthissetofmodelswhichmodelhaveleast MSEt hatisconsideredasgoodmodeltoprocess.

Program

scores_df=pd.DataFrame(scores)

display(scores_df)

Output

	Model name	mse	R^2
0	Linear regression	0.175628	0.034983
1	SVM	0.531661	-1.921297
2	Random Forest	0.155991	0.142881

```
scores_df[scores_df["mse"]==scores_df["mse"].min()]
```

Output

	Model name	mse	R^2
2	Random Forest	0.155991	0.142881

```
scores_df[scores_df["R^2"]==scores_df["R^2"].max()]
```

Output

	Model name	mse	R^2
2	Random Forest	0.155991	0.142881

From the above result we can conclude that random forest is the mostaccurate model for predicting the magnitude of Earthquake compared toallothermodels usedinthisproject.

CONCLUSION:

In conclusion, the earthquake prediction model developed using Pythonrepresents a significant step forward in harnessing the power of datascienceandmachinelearningforthecriticaltaskofearthquakeforecasting. This model has demonstrated its potential to contribute toearly warning systems, risk mitigation, and ultimately, saving lives andreducingtheimpactofseismicevents.Byleveragingadvancedalgorithms and a wealth of seismic data, it offers a promising avenue forimprovingourabilitytoanticipateearthquakes.

However,itisimportanttoacknowledgethatearthquakepredictionremains a highly complex and challenging endeavor due to the inherentuncertaintiesingeologicalprocesses.Whilethismodelshowspromise,it is not a panacea, and further research and data collection are necessarytorefineandenhanceitsaccuracyandreliability.

In the hands of dedicated researchers and scientists, this model can serveasavaluabletoolintheongoingquesttounderstandandpredictearthquakes.Itsdevelopmentunderscorestheimportanceofinterdisciplinary collaboration, ongoing data acquisition, and innovativeapproaches to tackle one of the world's most pressing natural hazards.Withcontinuedrefinementandintegrationintoseismicmonitoringsystems,thisPython- basedearthquakepredictionmodelhasthepotential to make a real difference in our preparedness and response toseismicevents.