**ARTIFICIALINTELLIGENCE-GROUP4**

**EARTHQUAKEPREDICTIONMODELUSINGPYTHON**

**Team members**

**1)KRISHNA RAJAN M**

**2)JOSHUVA M**

**3)AKASH K**

**4)ESAKKIAPPAN E**

**5)LOKESH BHARATHI A**

**EARTHQUAKEPREDICTIONMODELUSINGPYTHON**

# Earthquakepredictionmodelinloadingandpre-processingthedataset:

**Introduction:**

Machine learning has the ability to advance our knowledge ofearthquakes and enable more accurate forecasting and catastropheresponse.It’scrucialtorememberthatdevelopingaccurateanddependable prediction models for earthquakes still needs more studyasitisacomplicated anddifficult topic.

In order to anticipate earthquakes, machine learning may be used toexamineseismicdatatrends.Seismometerscaptureseismicdata,whichmaybeusedtospotchangestotheearth’ssurface,likeseismicwaves brought on by earthquakes. Machine learning algorithms mayutilize these patterns to forecast the risk of an earthquake happeninginacertainregionbystudyingthesepatternsandlearningtorecognizekeytraits thatarelinkedtoseismicactivity.

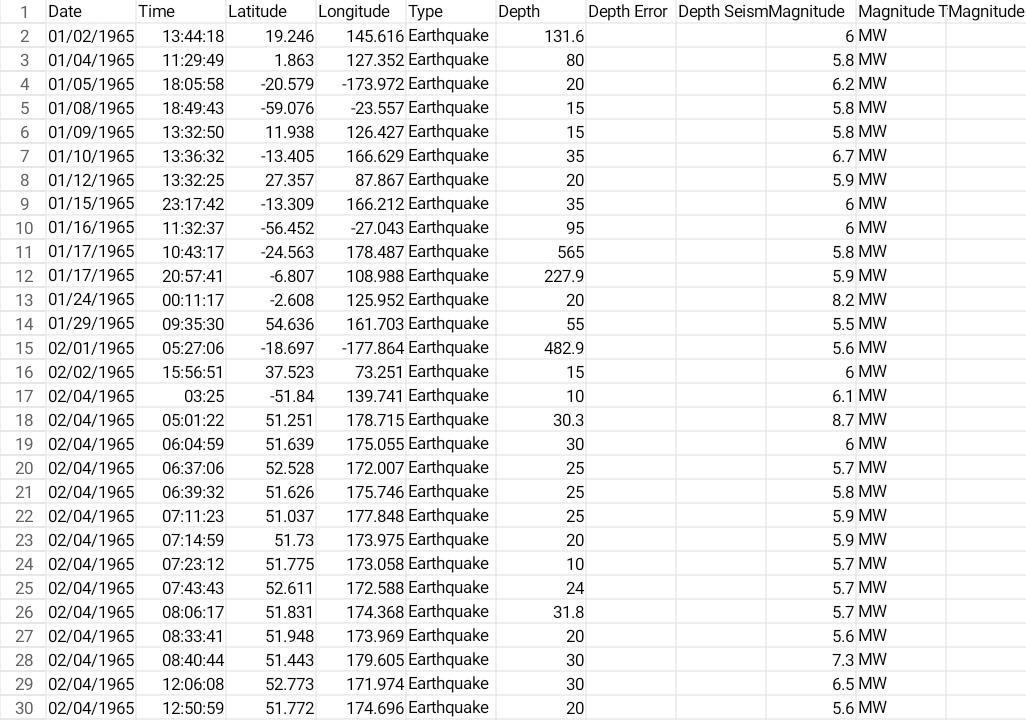
**AboutDataset:**

TheNationalEarthquakeInformationCenter(NEIC)determines the location and size of all significant earthquakes thatoccur worldwide and disseminates this information immediately tonational and international agencies, scientists, critical facilities, andthe general public. The NEIC compiles and provides to scientists andtothepublicanextensiveseismicdatabasethatservesasafoundationforscientificresearchthroughtheoperationofmoderndigitalnationalandglobalseismographnetworksandcooperative

internationalagreements.TheNEICisthenationaldatacenterandarchiveforearthquakeinformation.

# Content:

This dataset includes arecordofthe date,time, location, depth,magnitude,andsourceofeveryearthquakewithareportedmagnitude5.5or highersince1965.



# Necessary step to follow:1.ImportLibraries:

Startbyimportingthenecessarylibraries:

# Program:

Import pandas as pdImportnumpyasnp

From sklearn.model\_selection import train\_test\_splitFrom sklearn.preprocessing import StandardScaler**2.LoadtheDataset:**

LoadyourdatasetintoaPandasDataFrame.Youcantypicallyfind

HousepricedatasetsinCSVformat,butyoucanadaptthiscodetoother

Formats as needed.Program:

Df=pd.read\_csv(‘E:\USA\_Housing.csv‘)Pd.read()

# ExploratoryDataAnalysis(EDA):

Perform EDA to understand your data better. This includesChecking for missing values, exploring the data’s statistics, andVisualizingitto identifypatterns.

Program:

# Check for missing valuesPrint(df.isnull().sum())

# Explore statisticsPrint(df.describe())

#Visualizethedata(e.g.,histograms,scatterplots,etc.)

# FeatureEngineering:

Dependingon yourdataset,you mayneedtocreatenewfeaturesor

Transform existing ones. This can involve one-hot encodingcategoricalVariables,handlingdate/timedata,orscalingnumericalfeatures.

Program:

#Example:One-hotencodingforcategorical variables

Df=pd.get\_dummies(df,columns=[‘Avg.AreaIncome‘,‘Avg.AreaHouse Age‘])

# SplittheData:

Split your dataset into training and testing sets. This helps youevaluate

Yourmodel’sperformancelater.

X = df.drop(‘price’, axis=1) # FeaturesY=df[‘price’] #Targetvariable

X\_train,X\_test,y\_train,y\_test=train\_test\_split(X,y,test\_size=0.2,Random\_state=42)

# FeatureScaling:

Apply feature scaling to normalize your data, ensuring that allFeatureshavesimilarscales.Standardization(scalingtomean=0andStd=1)isacommonchoice.

Program:

Scaler=StandardScaler()

X\_train = scaler.fit\_transform(X\_train)X\_test=scaler.transform(X\_test)

# Importanceofloadingandprocessingdatase:

* 1. **QualityIn,QualityOut:**

Theintegrityofyouranalysisormodeldependsonthequalityofyour data. Loading data properly ensures that you’re working withaccurateandreliableinformation.

# UnderstandingYourData:

Before diving into analysis or model building, you need a clearunderstandingofyourdata.Loadingitallowsyoutoexploreitsstructure,identifypatterns,andgaininsightsintoitscharacteristics.

# DataCleaning:

Real-world data is often messy. Loading data lets you identifymissingvalues,outliers,anderrorsthatneedtobeaddressedthroughcleaningprocesses.Thisisessentialformeaningfulandaccurateresults.

# FeatureEngineering:

Loading data is the first step in feature engineering, where youtransformandenhanceyourvariablestoimprovemodelperformance.Thiscaninvolvecreatingnewfeatures,scaling,orencodingcategorical variables.

# Compatibility:

Different models or analysis tools may require data in specificformats. Loading data allows you to transform it into a compatiblestructure,ensuringasmoothworkflow.

# Efficiency:

Processing data efficiently can have a significant impact on thespeed and performance of your analysis or model. This includes taskssuchasindexing,sorting,andaggregatingdatatostreamlinesubsequentoperations.

# DataExploration:

Loadingdataallowsyoutovisualizeandexploreitthroughvariousstatisticalandgraphicalmethods.Thisexplorationiscrucialforformulatinghypotheses,identifyingtrends,andmakinginformeddecisionsabout furtheranalysis.

# ModelTraining:

Ifyou’rebuildingamachinelearningmodel,loadingandprocessingdataisaprerequisitefortraining.Themodel’sperformanceheavilydependsonthequalityandcharacteristicsofthetrainingdata.

# IterativeProcess:

Dataloadingandprocessingareofteniterativeprocesses.Asyouanalyzeormodel,youmaydiscovertheneedforadditionalpreprocessing or adjustments. Having a well-organized and flexibledataprocessingpipelinemakesiteasiertoadapt.

# Reproducibility:

Properlyloadingandprocessingdatacontributetothereproducibility of your work. If someone else needs to replicate youranalysisormodel,clearandwell-documenteddataloadingandprocessing stepsareessential.

# Challengesinvolvedinloadingandpreprocessingaearthquakepredictiondatasets:

**DataVolumeandSize:**

Earthquakedatasetscanbemassive,especiallyiftheycoverlongtime periods and include detailed information. Managing and loadinglargedatasetscanstraincomputationalresourcesandrequireefficientstoragesolutions.

# DataVariety:

Earthquakedataoftencomesinvariousformats,includingtime-series data, geospatial data, and categorical information. Integratingandpreprocessingthesediversedatatypescanbechallenging,requiringspecializedtechniquesfor each.

# DataQuality:

Earthquake datasets may have missing or inaccurate values,outliers, or inconsistencies. Cleaning and validating the data is crucialtoensuretheaccuracyofpredictions.Incompleteorincorrectinformation can significantly impact the performance of predictionmodels.

# TemporalandSpatialDependencies:

Earthquakesexhibittemporalandspatialdependencies.Preprocessingmustconsiderthetimeintervalsbetweenseismicevents and the geographical relationships between data points. Thismight involve creating features that capture trends and patterns overtimeandspace.

# ImbalancedClasses:

Theoccurrenceofsignificantearthquakesisrarecomparedtosmallerseismicactivities.Thisclassimbalancecanposechallengesformachinelearningmodels,whichmightstruggletolearnpatternsassociated with infrequent events. Techniques such as oversampling,undersampling, or using appropriate evaluation metrics need to beconsidered.

# FeatureEngineering:

Extractingmeaningfulfeaturesfromseismicdatarequiresdomain expertise. Transforming raw sensor readings into relevantfeatures,suchasfrequencycomponents,amplitude,andspectralcharacteristics,isacrucialpreprocessingstepforearthquakeprediction.

# NormalizationandScaling:

Differentsensorsandmeasurementunitsmaybeusedinearthquakedatasets.Normalizingandscalingfeaturesareessentialtoensure that the model interprets all variables on a consistent scale,preventingcertainfeaturesfromdominating thelearningprocess.

# HandlingTimeSeriesData:

Earthquakedataofteninvolvestimeseriesinformation.Dealingwithtime-dependentpatterns,seasonality,andtrendsrequiresspecializedpreprocessingtechniquessuchastime-seriesdecomposition,lagfeatures,orrollingstatistics.

# ComputationalIntensity:

Earthquakepredictionmodels,especiallythosebasedonmachinelearningalgorithms,canbecomputationallyintensive.Preprocessingstepsneedtobeoptimizedforefficiency,andconsideration should be given to parallel processing or distributedcomputingforlargedatasets.

# Domain-SpecificChallenges:

Understanding the geological and seismological context iscrucial. Domain-specific knowledge is needed for meaningful featureselection, identifying relevant patterns, and interpreting the results.Collaboratingwithdomain expertsisoftennecessary.

# LOADINGTHEDATASET:

Loading the dataset using machine learning is the process ofbringingthedataintothemachinelearningenvironmentsothatitcanbe usedtotrain andevaluateamodel.

ThespecificstepsinvolvedinloadingthedatasetwillvarydependingOnthemachinelearninglibraryorframeworkthatisbeingused.However, there are some general steps that are common tomostMachinelearning frameworks.

# Identifythedataset:

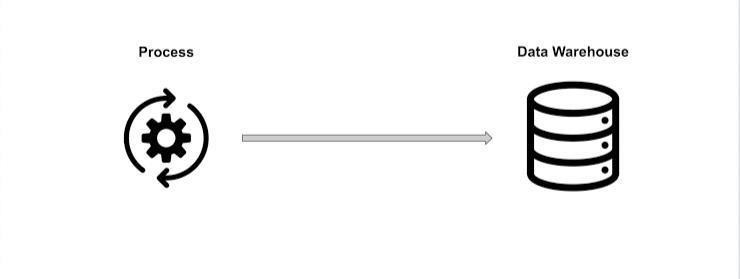
Thefirststep istoidentifythedatasetthatyouwanttoload.This

Datasetmaybestoredinalocalfile,inadatabase,orinacloudstorageService.

# Loadthedataset:

Once you have identified the dataset, you need to load it into theMachine learning environment. This may involve using a built-inFunctioninthemachinelearninglibrary,oritmayinvolvewritingyourOwncode.

# Preprocessthedataset:



Oncethedatasetisloadedintothemachinelearningenvironment,Youmayneedtopreprocessitbeforeyoucanstart

training and Evaluating your model. This may involve cleaning thedata, transforming the data into a suitable format, and splitting thedataintotrainingandTestsets.

# Input dataPandas:

This library helps to load the data frame in a 2D array format andhasmultiplefunctionstoperformanalysistasksinonego.

# Matplotlib/Seaborn:

Thislibraryisusedtodrawvisualizations.

Importpandasaspd

Import matplotlib.pyplot as pltImportseabornassb

Importwarnings

Warnings.filterwarnings(‘ignore’)

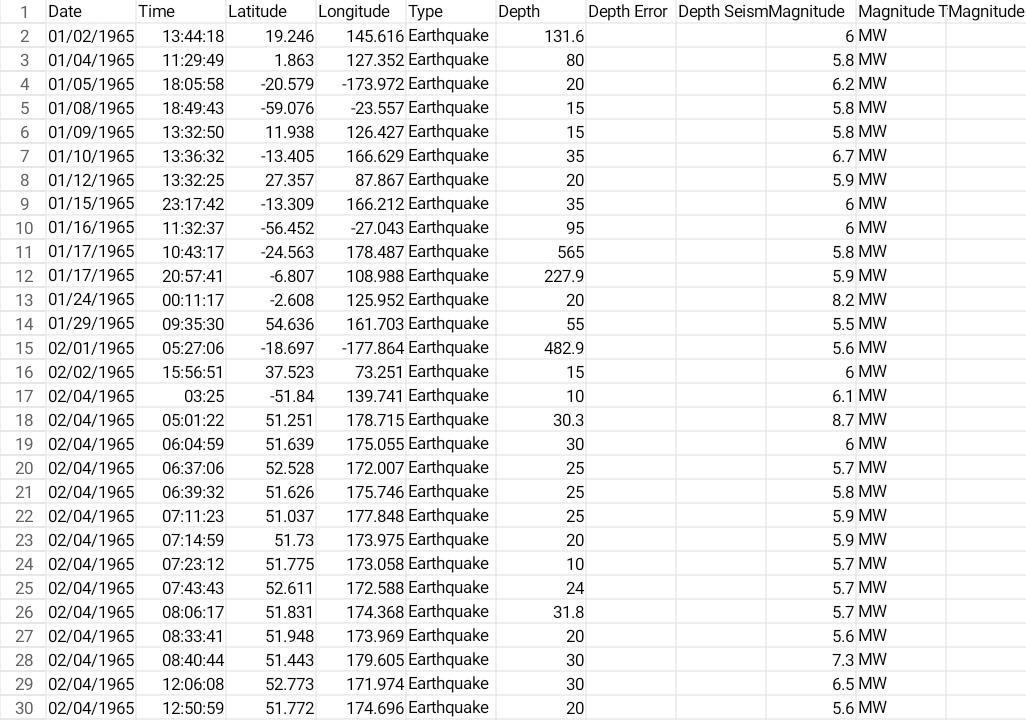
Df=pd.read\_csv(‘dataset.csv’)Df.head()

# Data exploration:Dataset

**Output:**

Thedatasetweareusingherecontainsdataforthefollowingcolumns:

* OrigintimeoftheEarthquake
* Latitudeandthelongitudeofthelocation.
* Depth–Thismeanshowmuchdepthbelowtheearth’sleveltheearthquakestarted.
* Themagnitudeoftheearthquake
* Location



# PREPROCESSINGDATASETS

Datapreprocessingistheprocessofcleaning,transforming,andintegratingdatainordertomake itreadyforanalysis.

This may involve removing errors and inconsistencies, handlingMissingvalues,transformingthedataintoaconsistentformat,andScalingthedatatoasuitablerange.

# Visualization:

Frommpl\_toolkits.basemapimportBasemap

M= Basemap(projection=’mill’,llcrnrlat=-80,urcrnrlat=80,llcrnrlon=-180,urcrnrlon=180,lat\_ts=20,resolution=’c’)

Longitudes = data[“Longitude”].tolist()Latitudes=data[“Latitude”].tolist()

#m = Basemap(width=12000000,height=9000000,projection=’lcc’,#resolution=None,lat\_1=80.,lat\_2=55,lat\_0=80,lon\_0=-107.)

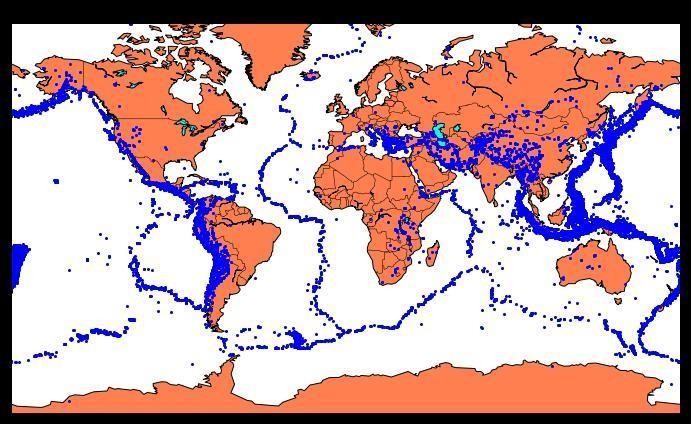
X,y=m(longitudes,latitudes)

Fig=plt.figure(figsize=(12,10))Plt.title(“All affectedareas”)

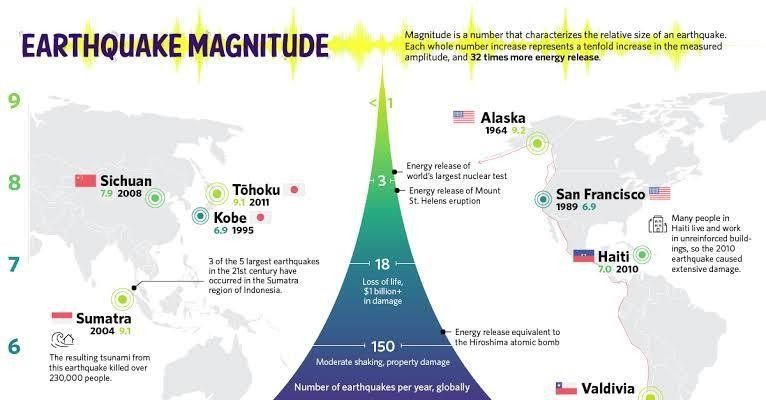
m.plot(x,y,“o”,markersize=2,color=‘blue’)m.drawcoastlines()

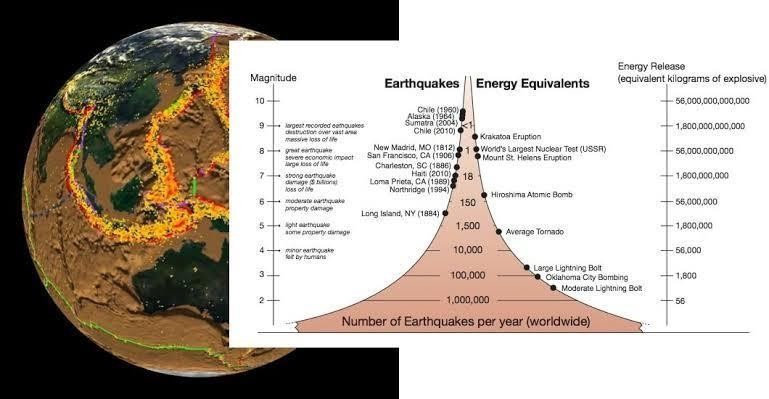
m.fillcontinents(color=’coral’,lake\_color=’aqua’)m.drawmapboundary()

m.drawcountries()plt.show()



Wehavelocatedontheworldmapwhereearthquakeshappenedinthelastfewyears.





# SplittingTheDataset:

X=final\_data[[‘Timestamp’,‘Latitude’,‘Longitude’]]Y=final\_data[[‘Magnitude’, ‘Depth’]]

Fromsklearn.cross\_validationimporttrain\_test\_split

X\_train,X\_test,y\_train,y\_test=train\_test\_split(X,y,test\_size=0.2,random\_state=42)

Print(X\_train.shape,X\_test.shape,y\_train.shape,X\_test.shape)

# Output:



WewillbeusingtheRandomForestRegressormodeltopredicttheearthquake,herewilllookforits accuracy.

Reg = RandomForestRegressor(random\_state=42)Reg.fit(X\_train, y\_train)

Reg.predict(X\_test)Reg.score(X\_test,y\_test)



# NeuralNetworkModel:

Aneuralnetworkmodelcanbeemployedtoforecastearthquakes by examining diverse elements and trends in seismicdata. This model harnesses the capabilities of neural networks, whichdraw inspiration from the neural connections of the human brain, toanalyzeintricatedataandrevealhiddenrelationshipsandpatterns.Bytrainingtheneuralnetworkonhistoricalearthquakedata,itcanacquiretheabilitytoidentifyprecursorsignalsandpatternsthatindicatetheprobabilityofanupcomingearthquake.

From keras.models import SequentialFromkeras.layersimportDense

Def create\_model(neurons, activation, optimizer, loss):Model=Sequential()

Model.add(Dense(neurons, activation=activation,input\_shape=(3,)))

Model.add(Dense(neurons, activation=activation))Model.add(Dense(2,activation=’softmax’))

Model.compile(optimizer=optimizer, loss=loss,metrics=[‘accuracy’])

Returnmodel

Fromkeras.wrappers.scikit\_learnimportKerasClassifier

Model=KerasClassifier(build\_fn=create\_model,verbose=0)

#neurons=[16,64,128,256]

Neurons=[16]

#batch\_size=[10,20,50,100]

Batch\_size=[10]

Epochs=[10]

# activation = [‘relu’, ‘tanh’, ‘sigmoid’, ‘hard\_sigmoid’, ‘linear’,‘exponential’]

Activation=[‘sigmoid’,‘relu’]

# optimizer=[‘SGD’,‘RMSprop’,‘Adagrad’,‘Adadelta’,‘Adam’,‘Adamax’,‘Nadam’]

Optimizer = [‘SGD’, ‘Adadelta’]Loss=[‘squared\_hinge’]

Param\_grid=dict(neurons=neurons,batch\_size=batch\_size,epochs=epochs,activation=activation,optimizer=optimizer,loss=loss)

Grid=GridSearchCV(estimator=model,param\_grid=param\_grid,n\_jobs=-1)

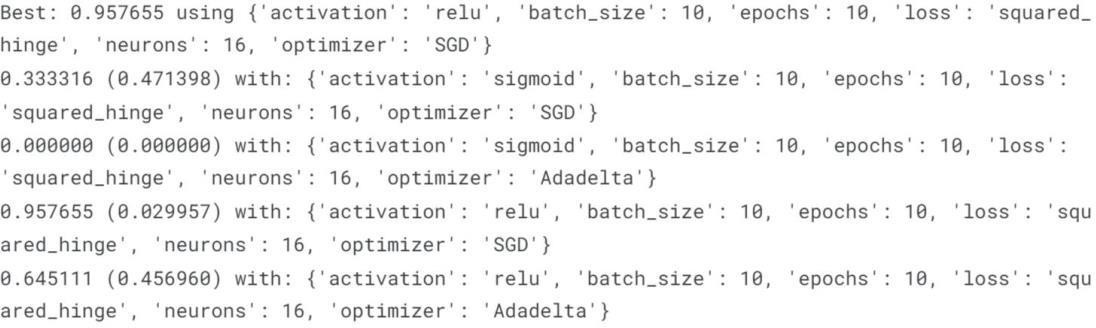
Grid\_result=grid.fit(X\_train,y\_train)

Print(“Best:%fusing%s”%(grid\_result.best\_score\_,grid\_result.best\_params\_))

Means = grid\_result.cv\_results\_[‘mean\_test\_score’]Stds = grid\_result.cv\_results\_[‘std\_test\_score’]Params=grid\_result.cv\_results\_[‘params’]

Formean,stdev,paraminzip(means,stds,params):Print(“%f(%f)with:%r”%(mean,stdev,param))

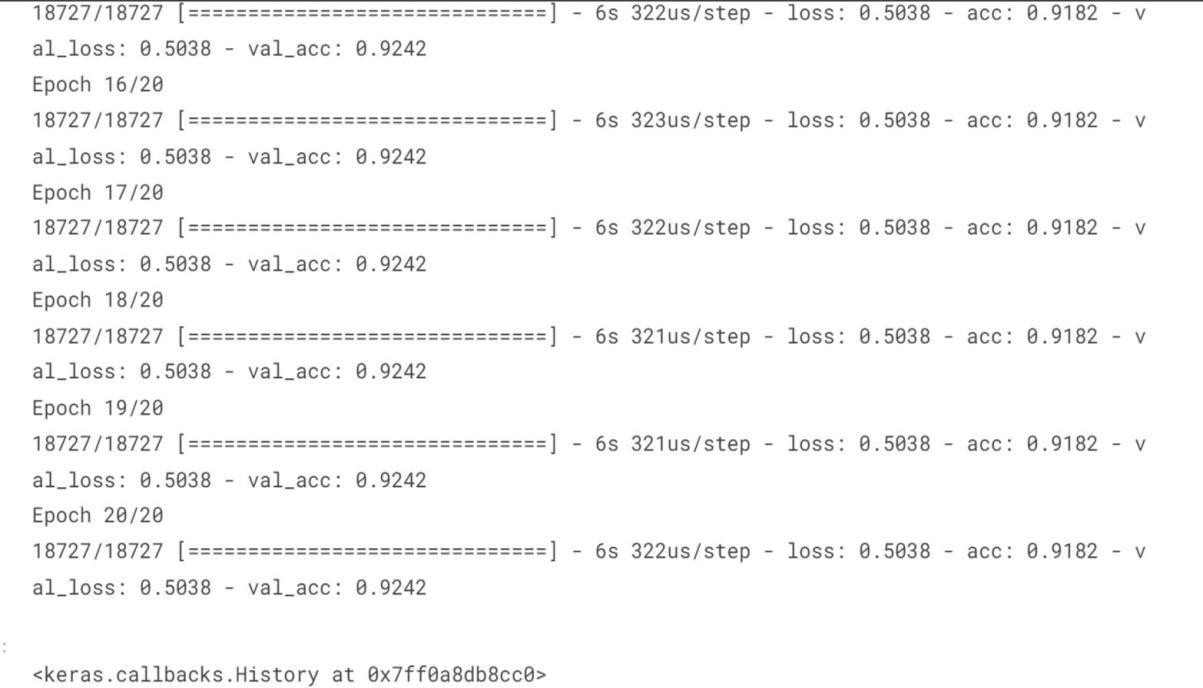
# Output:



Model=Sequential()

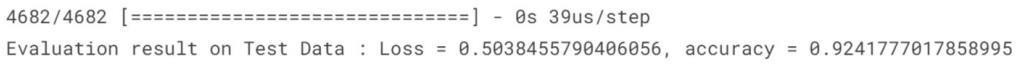
Model.add(Dense(16, activation=’relu’, input\_shape=(3,)))Model.add(Dense(16,activation=’relu’))Model.add(Dense(2,activation=’softmax’))

Model.compile(optimizer=’SGD’, loss=’squared\_hinge’,metrics=[‘accuracy’])



Model.Fit(X\_train,y\_train,batch\_size=10,epochs=20,verbose=1,validation\_data=(X\_test,y\_test))

# Output:



Isn’titamazing thatwegotanaccuracyof92%.

Wecansaytheneuralnetworkisoneofthebestmodelstopredictearthquakesthatcan beusedinfuture.

# Conclusion:

Understanding earthquakes and effectively responding to themremainsacomplexandchallengingtask,evenwiththelatesttechnological advancements. However, leveraging the capabilities ofmachine learning can greatly enhance our comprehension of seismicevents. By employing machine learning techniques to analyze seismicdata,wecanuncovervaluableinsightsandpatternsthatcontributetoadeeperunderstandingofearthquakes.Theseinsightscansubsequently inform more effective strategies for mitigating risks andrespondingtoseismicevents.

As we head towards the future, we might see new technologies thatwill precisely predict the place and time of the earthquake that willhappen.