Earthquake Prediction in a Region Using Regression Algorithms and Big Data Infrastructure

Ashish Kumar
axk171531@utdallas.edu

Varun Mehrotra
vxm170830@utdallas.edu

Purna Rama Gopal Vetukuri
pxv172130@utdallas.edu

Siddharth Swarup Panda
ssp171730@utdallas.edu

Abstract – An earthquake is the shaking of the surface of Earth caused due to movable plate boundary interactions, resulting from sudden release of energy that creates seismic waves. Earthquakes are measured using remarks from seismometers with Richter magnitude scale. Earthquake prediction is a branch of the science of seismology concerned with specification of time (reported in milliseconds), location (lat/lng) and magnitude of future earthquakes. We are particularly interested in the determination of the parameters for the next strong earthquake to occur in a region.

Earthquake magnitude prediction has been studied widely in the last decades. Statistical, geophysical and machine learning approaches can be found in literature, with no particularly satisfactory results. In the recent years, huge datasets are analysed using powerful computational big data techniques. California is one of the regions with highest seismic activity in the world with lots of data available to carry out analysis. This report aims at providing the preliminary regression techniques and approaches that we will be applying to do Earthquake prediction in California.

I. INTRODUCTION

The modern world is susceptible to natural calamities and every now and then there is huge economical as well as human impact due to natural disasters. So, there is a demand of proper preparation for reducing their impact. Among natural disasters like earthquakes, tsunamis, hurricanes, floods, etc, earthquake has the most devastating effects on human and economic losses. Since, it arrives suddenly without any prior hint and can damage the whole region in fraction of seconds leading to huge property and human damages, social and economic harm, a successful earthquake prediction model can be a great contribution towards mankind in reducing its impact.

Big data analytics is a very powerful technique which can be utilized to analyse massive datasets and extract fruitful results. In today's world, we have plethora of datasets and machine learning algorithms can be implemented on them along with big data models. These techniques can make use of the huge amount of earthquake data available to predict future earthquake magnitudes.

For our project, we have considered various machines learning techniques such as Random Forests, Extreme Gradient Boosting, Pipelines, Principal Component Analysis (PCA), etc. Random Forests is an ensemble learning technique that is based on decision trees and we create new datasets by varying the instances from original dataset(cross validation). PCA is a mathematical tool from applied linear algebra used for extracting relevant information from confusing datasets by reducing complex data to lower dimensions. Pipelines simplify the ML process by modularizing various phases which consist of a series of operations that are run sequentially. ML Pipelines consist of Transformers and Estimators components.

We chose "**Earthquake parameters catalog of the Northern California Seismic Network**" as the project dataset for performing analysis. The dataset is available in EHP CSV format.

	A	B	C	D	E	F	G	H	I	J	K	L	M
1	time	latitude	longitude	depth	mag	magType	nst	gap	dmin	rms	net	id	updated
2	2013-04-11T	20.7915	122.226	8.29	4.6	mb	46	115	2.28	1.21	us	usb000g50m	2013-04-11T
3	2013-04-11T	-17.3579	175.0663	9.88	5.3	mb		50	7.802	1.44	us	usb000g4z6	2013-04-11T
4	2013-04-11T	-17.4508	-178.7735	535.92	4.9	mb	21	128	8.56	0.71	us	usb000g4xf	2013-04-11T
5	2013-04-11T	-16.9546	-179.1921	528.46	4.5	mb	45	75	9.06	0.84	us	usb000g4ug	2013-04-11T
6	2013-04-11T	-10.6708	166.0755	167.21	4.7	mb	42	102	6.16	0.8	us	usb000g4su	2013-04-11T
7	2013-04-11T	2.8643	125.4971	64.96	4.6	mb		72	2.794	0.78	us	usb000g4rg	2013-04-11T
8	2013-04-11T	20.9199	122.1061	12.94	4.5	mb	32	130	2.12	0.69	us	usb000g4qu	2013-04-11T
9	2013-04-11T	-2.7939	148.1628	9.39	4.6	mb	23	147	1.09	1.12	us	usb000g4q9	2013-04-11T
10	2013-04-11T	19.2629	95.6948	10.07	5.2	Mwp	75	43	2.14	0.98	us	usb000g4nw	2013-04-11T
11	2013-04-11T	41.6165	141.9924	55.27	4.6	mb	43	123	0.96	1.09	us	usb000g4ni	2013-04-11T
12	2013-04-11T	28.5074	51.6758	10.07	4.8	mb		64	10.83	1.19	us	usb000g4mt	2013-04-11T
13	2013-04-10T	18.854	97.5096	8.27	4.7	mb	29	75	0.63	0.6	us	usb000g4i5	2013-04-11T
14	2013-04-10T	20.8187	122.1203	4.2	5.8	Mww	115	31	2.21	1.28	us	usb000g4ca	2013-04-10T
15	2013-04-10T	2.6017	127.2174	66.02	5	mb	60	105	1.82	1.03	us	usb000g4br	2013-04-10T
16	2013-04-10T	15.5366	-87.228	10	5.5	mb		37	1.471	0.85	us	usb000g4a2	2013-04-11T
17	2013-04-10T	-10.7302	-75.2622	99.62	5.2	mb		72	1.99	0.77	us	usb000g43v	2013-04-10T
18	2013-04-10T	-17.7569	167.7868	10	4.6	mb	20	160	3.71171	0.87	us	us2013nvap	2013-04-10T
19	2013-04-10T	28.5135	51.5523	9.93	4.6			94	10.877	0.97	us	usb000g3y3	2013-04-10T
20	2013-04-10T	28.438	51.738	9.87	5.2	mb	76	58	10.87	1.03	us	usb000g3ts	2013-04-10T
21	2013-04-10T	28.309	51.7514	10.06	4.8	mb		75	10.974	0.95	us	usb000g3t2	2013-04-10T
22	2013-04-10T	37.4728	142.0723	27.79	4.6	mb	40	132	3.23	1.17	us	usb000g3qe	2013-04-10T
23	2013-04-10T	28.45	51.6075	10.02	5.6	mb	76	25	10.91	1.14	us	usb000g3p7	2013-04-10T
24	2013-04-10T	-2.9729	139.0662	55.19	4.8	mb	33	61	6.81	1.6	us	usb000g3ns	2013-04-10T
25	2013-04-10T	28.4814	51.604	10	4.9	mb		139	10.883	0.83	us	usb000g3nn	2013-04-10T
26	2013-04-10T	-2.0824	-79.5666	103.34	4.5	mb	35	113	2.56	0.5	us	usb000g3ng	2013-04-11T
27	2013-04-09T	-22.7541	69.1376	10.2	4.6	mb	17	110	11.11	0.55	us	usb000g3ls	2013-04-09T
28	2013-04-09T	28.2759	51.6754	9.88	4.8	mb		86	11.034	0.68	us	usb000g3ft	2013-04-09T
29	2013-04-09T	5.6129	93.3101	31.21	4.7	mb		139	3.641	0.61	us	usb000g3fe	2013-04-09T
30	2013-04-09T	28.4201	51.6408	19.93	4.6	mb	94	94	10.92	0.72	us	usb000g3dn	2013-04-09T

Snapshot of dataset

There were 22 columns in the original dataset with many

rows with null entries. We then pre-processed the data by dropping unwanted columns and removing null entries. More details about the dataset and pre-processing techniques is discussed in the following sections.

II. DATASET DESCRIPTION

The dataset chosen for the project is taken from ftp://www.ncedc.org/pub/catalogs/NCSS-catalogs/. The dataset contains a snapshot of earthquake parameters catalogue of the Northern California Seismic Network from 1966 to present. The format of the data is EHP CSV (comma separated values) ASCII text file.

It has the following 22 attributes:
Time(time reported in milliseconds when the event occurred)**, latitude**(Decimal degrees latitude. Negative values for southern latitudes.)**, longitude**(Decimal degrees longitude. Negative values for western latitudes.)**, depth(**Depth of even in kilometres**), mag (**Magnitude of the event**) , magType**(algorithm used to calculate the preferred magnitude for the event), **nst**(total number of seismic stations used to determine earthquake location), **gap**(largest azimuthal gap between azimuthally adjacent stations (in degrees)) , **dmin**(Horizontal distance from the epicentre to the nearest station (in degrees)), **rms**(root-mean-square (RMS) travel time residual, in sec, using all weights), **net**(ID of a data contributor), **id**(a unique identifier for the event), **updated**(time when the event was most recently updated)**, place**(Textual description of named geographic region near to the event), **type**(type of seismic event)**, horizontal Error**(Uncertainty of reported location of the event in kilometres)**, depth Error, magError, magNst**(total number of seismic stations used to calculate the magnitude for this earthquake)**, status**(Indicates whether the event has been reviewed by a human)**, Location Source**(the network that originally authored the reported location of this event)**, MagSource**(network that originally authored the reported magnitude for this event).

III. PREPROCESSING TECHNIQUES

The originally downloaded dataset contained 22 attributes in total, including data other than earthquake data and null entries. We plotted the different attributes to see their distribution as shown in figure 1 to figure 4. We found out from figure 1 that there are two types of seismic event and we require only require earthquake type. So, we filtered out other type and keep only earthquake type.

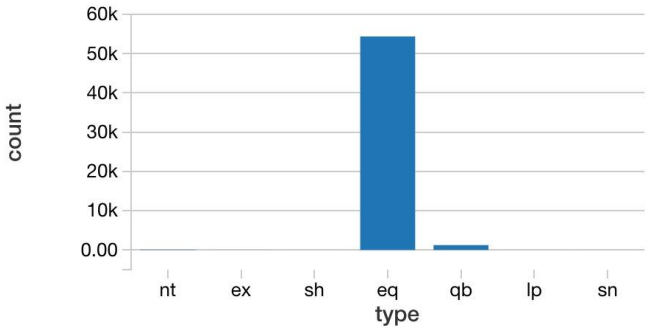

Fig 1: count vs. type of seismic event

We also plotted rms vs magnitude which to see how much noise in the data and we see majority of the data is useful as rms is less.

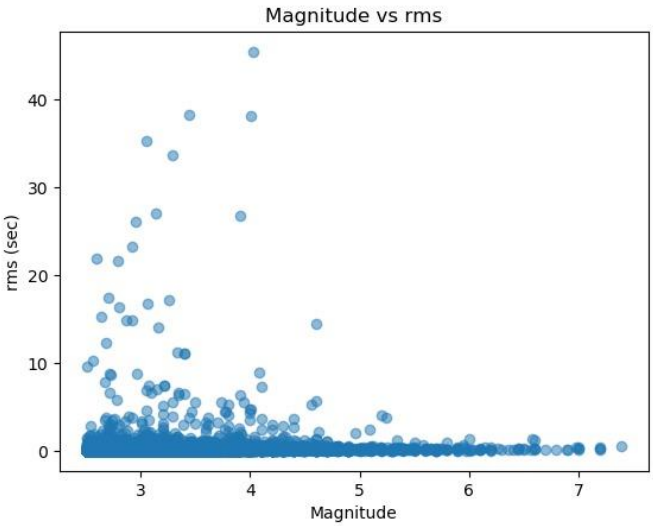

Fig 2: Magnitude vs rms

We started with filtering the data, keeping entries only where the "mag" values is greater than 2.5 as earthquake magnitude less than or equal to 2.5 are usually not felt but are still recorded by seismograph. As it can be inferred from figure 3, magnitude vs frequency plot, most of the earthquake are 2.5 to 3.25, which kind of makes sense as earthquake of smaller magnitude are much more frequent.

Then, as the analysis is for earthquake prediction, data other than earthquake info are removed. The rows containing any null or NaN values ar dropped. Also, the columns locationSource, magSource, status, magNst, nst, time, updated, net, place, id doesn't add value to

our analysis and in predicting the future earthquake parameters. So above mentioned columns are dropped as well. We filtered out such unwanted data and narrowed down the number of attributes to 11 attributes by dropping 11 columns.

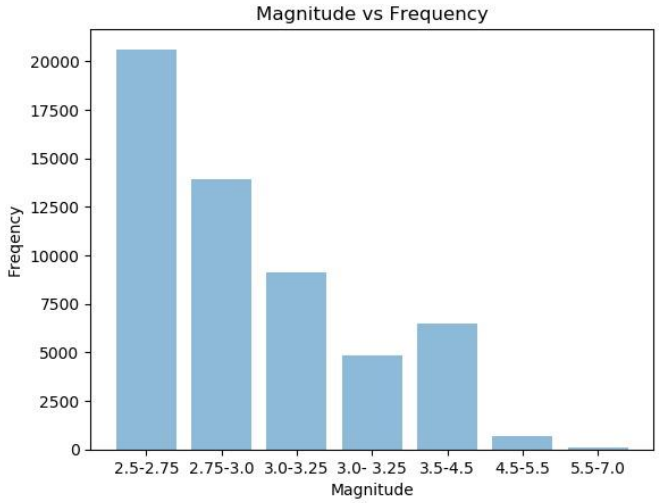

Fig 3: Magnitude vs Frequency

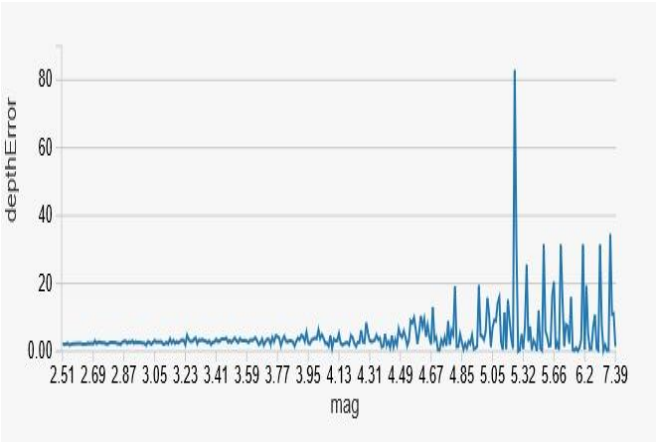

Fig 4: Magnitude vs depth error

The magType column contains method or algorithm used to calculate the preferred magnitude for the event in String format, so these data were parsed to column of label indices using String Indexer. Vector assembler was used to combine the columns latitude, longitude, depth, gap, dmin, rms, horizontal Error, depth Error, mag Error, newMagType into a single vector column to be considered as the features. Vector Assembler is a transformer that is useful in combining multiple raw features into a single feature vector for training the machine learning models.

Feature vector is scaled using the normalization techniques such as Standard Scaler and Normalizer. The StandardScaler standardizes features by scaling to unit variance and/or removing the mean using column summary statistics on the samples in the training set. Normalizer scales individual samples to have unit L^p norm. It implements VectorTransformer which apply the normalization on a vector to produce transformed vector.

Primary Component Analysis (PCA) is used for dimensionality reduction. It is a method to find a rotation such that the first coordinate has largest variance possible. *spark.mllib* supports PCA for tall and skinny matrices stored in row-oriented format. The columns if the roatition matrix are called principal components. PCA can be computed y eigen value decomposition or a data covariance matrix or SVD of a data matrix.

IV. PROPOSED SOLUTION

This project aims at predicting the earthquake magnitude in a California region using set of features. These features are extracted from the seismic data . Since the task is related to predicting the magnitude scale of earthquake, each feature in the feature vector has its own characteristic so a regression model is used to learn the important characteristics in the feature vector. We have used a set of 4 machine learning based regressors for prediction. The four methods are Linear Regressor, Decision Tree Regressor, Gradient Boosted Tree Regressor and Random Forest Regressor.

Linear Regressor is a linear approach to modelling the relationship between a scalar response and one or more explanatory variables. In linear regression, the relationships are modelled using linear predictor functions whose unknown model parameters are estimated from the data. Such models are called linear models. These models are often fitted using the least squares approach, but they may also be fitted in other ways, like by minimizing the "lack of fit" in some other norm or by minimizing a penalized version of the least squares cost function.

Decision tree regressor constructs a tree structure by breaking down the dataset into smaller subsets and ends with leaf node containing corresponding regressor value. It is a greedy algorithm which do a recursive binary partitioning of the feature space. The tree predicts the same label for each bottommost (leaf) partition. Each partition is chosen greedily by selecting the best split

from a set of possible splits, in order to maximize the information gain at a tree node. Decision trees are are easy to interpret, handle categorical features, extend to the multiclass classification setting. They do not require feature scaling, and can capture non-linearities and feature interactions, so they are widely used.

Gradient Boosted Tree (GBT) regressor are ensembles of decision trees. GBTs minimize a loss function by training decision trees iteratively. It builds the model in a stage-wise fashion like other boosting methods do. It fits an additive model (ensemble) in a forward stage-wise manner. GBT algorithms optimize a cost function over function space by iteratively choosing a function (weak hypothesis) that points in the negative gradient direction.

Random Forest regressor combines multiple decision trees in order to reduce overfitting by combining all the trees. It fits several classifying decision trees on various sub-samples of the dataset and to improve the predictive accuracy and control over-fitting, it uses averaging. By default, it leads to very large fully grown and unpruned trees on some data sets for the default parameter values that controls the size of the trees, e.g. max_depth, min_samples_leaf, etc. The features are always randomly permuted at each split. Hence, the best split may differ, even with the same training data, max_features = n_features and bootstrap = False, if the improvement of the criterion is identical for several splits enumerated during the search of the best split.

The spark.ml implementation supports linear regressor, decision trees, GBT, random forest for regression, using both continuous and categorical features.

V. RESULTS

All the models are evaluated with different types of error metrics.

The different types of regression metrics are:

RMSE (Root Mean Square Error) - It represents the sample standard deviation of the differences between predicted values and observed values (called residuals). Mathematically, it is calculated using this formula:

$$RMSE = \sqrt{\frac{\sum_{i=0}^{N-1}(y_i - yi^{\wedge})^2}{N}}$$

MSE (Mean Square Error) – It represents the average squared value difference between the estimated values and predicted values. It is a risk function , corresponding to the expected value of the squared error loss. Mathematically, it is calculated using this formula:

$$MSE = \frac{\sum_{i=0}^{N-1}(y_i - yi^{\wedge})^2}{N}$$

MAE (Mean Absolute Error) – It is the average of the absolute difference between the predicted values and observed value. It is a linear score. Its mathematical formula is :

$$MAE = \sum_{i=0}^{N-1} |y_i - y_i^{\wedge}|$$

R Squared (R^2) – It explains how the independent selected variables explain the variability in dependent variables. It is often used for explanatory purpose. Mathematically it is given as:

$$R^2 = 1 - \frac{\sum_{i=0}^{N-1}(y_i - yi^{\wedge})^2}{\sum_{i=0}^{N-1}(y_i - \bar{y})^2}$$

Where N is test data set size y_i is actual magnitude, $yi^{\wedge}$ is predicted magnitude and $\bar{y}$ is average magnitude.

Every Model resulted best score with Metric MSE 0.13 and the highest error with RMSE, R square metric depending on the model.

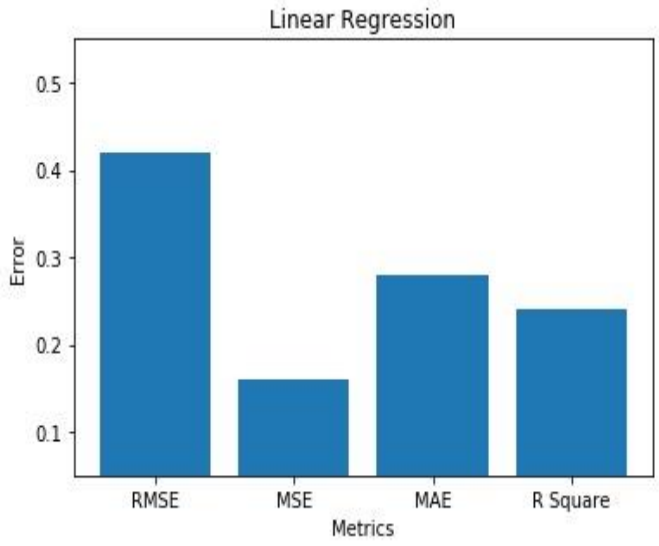

Fig 5: Linear Regression vs error metrics

We have experimented all the models with different parameters such as k folds, max iterations, max depth,

max no of trees and regularization parameter using Grid parameter. We considered the model with the best set of parameters obtained from the Grid search.

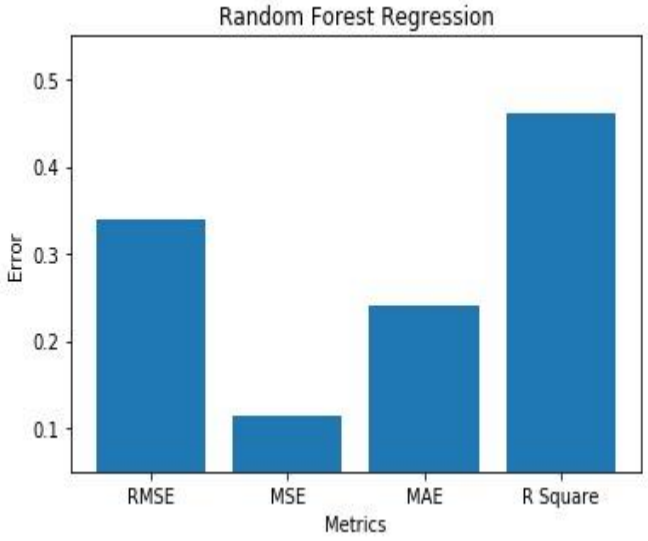

Fig 6: Random Forest Regression vs error metrics

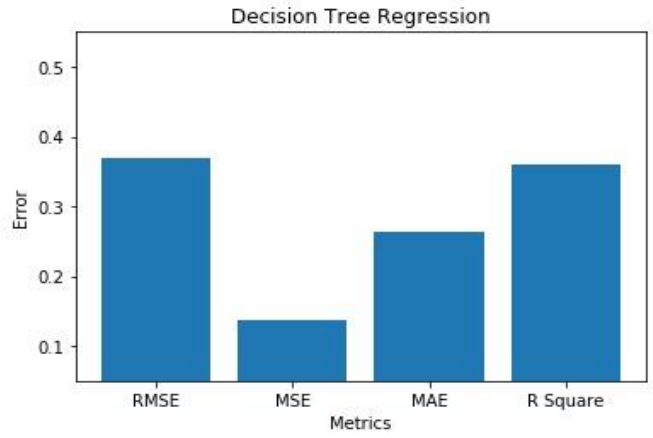

Fig 7: Decision Tree Regression vs error metrics

VI. CONCLUSION

Random forest has resulted the best performance in predicting the Magnitude for most of the Metrics. Because the Random forest is an ensemble technique that is obtained by constructing multiple decision trees and combines the result of each tree with a weight corresponding to the performance of each tree, it resulted in good fit and hence better performance.

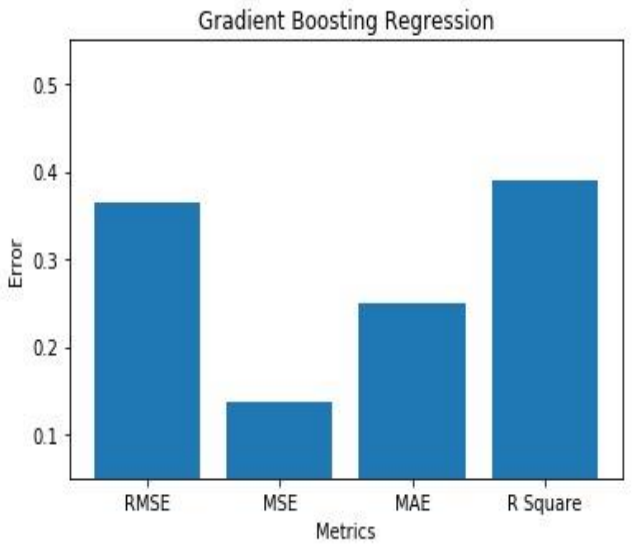

Fig 8: Gradient Boosting Regression vs error metrics

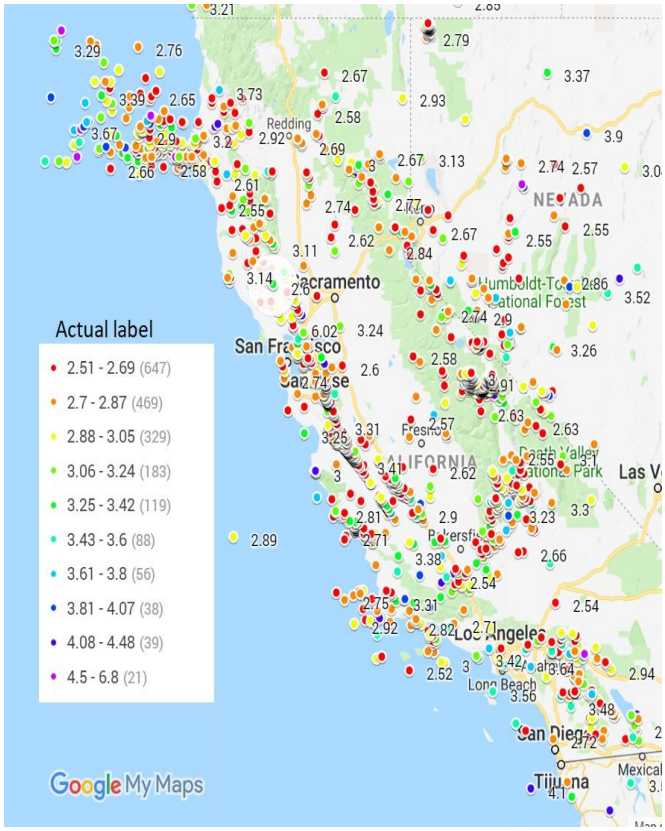

Fig 9: Actual earthquake magnitude labels

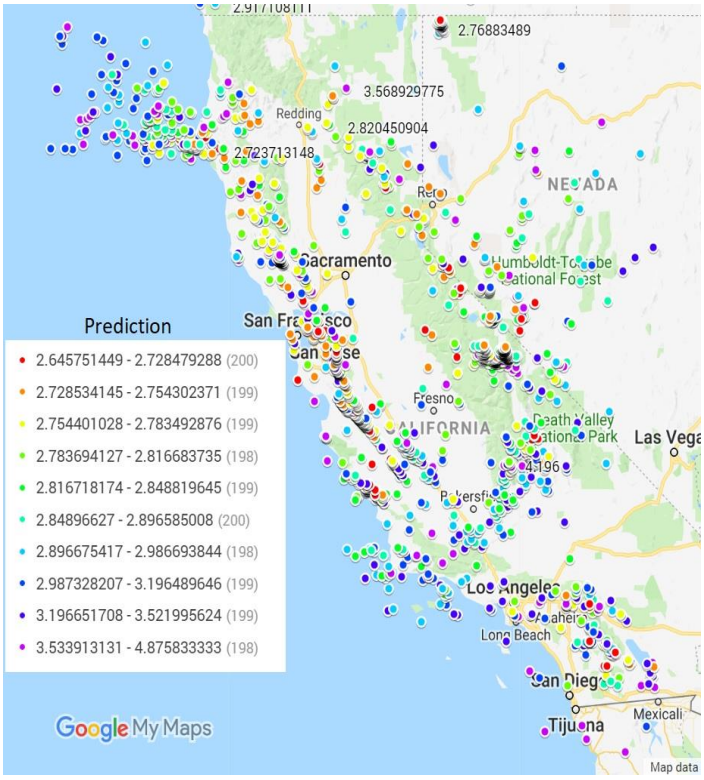

Fig 10: Predicted earthquake magnitude labels

VII. CONTRIBUTION

Varun, ram gopal:

Data Collection, formatting, model training (Gradient Boosting, Decision Tree Regressor)

Ashish, Siddharth:

Data pre-processing and model training (Linear Regression, Random forest Regression)

VIII. REFERENCES

L. F. Sá, A. Morales-Esteban, and P. Durand. A Seismic Risk Simulator for Iberia. Bulletin of the Seismological Society 416 of America, 106(3):1198–1209, 2016.

Q.Wang, D. D. Jackson, and Y. Y. Kagan. California earthquakes, 1800-2007: A unified catalog with moment magnitudes, uncertainties, and focal mechanisms. Seismological Research Letters, 80(3):446–457, 2009.

G. Asencio-Cortés, F. Martínez-Álvarez, A. Morales-Esteban, and A. Troncoso. Medium-large earthquake magnitude prediction in Tokyo with artificial neural networks. Neural Computing and Applications, 28(5):1043U1055, 2017.

R. Console, M. Murru, F. Catalli, and G. Falcone. Real time forecasts through an earthquake clustering model constrained by the rate-and-state constitutive law: comparison with a purely stochastic ETAS model.

D. A. Rhoades. Application of the EEPAS model to forecasting earthquakes of moderate magnitude in southern California.

A. Panakkat and H. Adeli. Neural network models for earthquake magnitude prediction using multiple seismicity indicators.International Journal of Neural Systems, 17(1):13–33, 2007.