Fine-tuned Predictive Model for Verifying POI Data

Way to Trusted Crowd Sourced GeoTagged Data

Monika Sharma[1]
Research Scholar, Dept. of Computer Engineering
Poornima University, Jaipur, Rajasthan, India

Vinod Bothale[3]
Associate Director, NRSC
ISRO, Balanagar, Hyderabad, India

Dr. Mahesh Bundele[2]
Principal and Director
Poornima College of Engineering
Jaipur, Rajasthan, India

Dr. Meenakshi Nawal[4]
Associate Professor
Dept. of Computer Engineering
Poornima University, Jaipur, India

***Abstract*—Mapping websites and geo portals are playing a vital role in daily life due to the availability of geo-tagged data. From booking a cab to search a place, getting traffic information, review of the place, searching for a doctor or best school available in the locality, we are heavily dependent on the map services and geo portals available for finding such information. There is voluminous data available on these sources and it is getting increasing every moment. These data are majorly collected through crowdsourcing methods where people are contributing. As a basic principle of Garbage in garbage out, the quality of this data impacts the quality of the services based on this data. Therefore, it is highly desired to have a model which can predict the quality/accuracy of the geotagged Point of interest data. We propose a novel Fine-Tuned Predictive Model to check the accuracy of this data using the best suitable supervised machine learning approach. This work focuses on the complete life cycle of the model building, starting from the data collection to the fine-tuning of the hyperparameters. We covered the challenges particularly to the geotagged POI data and remedies to resolve the issues to make it suitable for predictive modeling for classifying the data based on their accuracy. This is a unique work that considers multiple sources including ground truth data to verify the geotagged data using a machine learning approach. After exhaustive experiments, we obtained the best values for hyperparameters for the selected predictive model built on the real data set prepared specifically to target the proposed solution. This work provides a way to develop a robust pipeline for predicting the accuracy of crowdsourced geotagged data.**

***Keywords*—Crowdsourced; fine-tuning; geotagged data; hyperparameters; predictive model**

I. Introduction

We have been witnessing the data generation era where each day voluminous data is getting generated by people on different platforms like social media websites, microblogging websites, geo portals, web mapping websites, etc. Among these platforms, mapping websites and geo portals provide a wide variety of map data which has several important applications like traffic conditions, finding the route, business listing, etc. These maps contain a wide variety of Point of interest data such as public services like healthcare, schools, hotels, monuments, religious places, courts, open areas, business points, etc. The collection of such huge data is not possible without crowdsourcing. POI data are also known as geotagged data which includes the geographical information of a place along with the metadata. Many times, these data are contributed by general people hence there is no control over the quality of POI data. On the other hand, these data are used in various services like location route-finding which may lead to the wrong place if the metadata or geospatial data are not correct. There may be critical consequences of the wrongly tagged data. Hence, it is important to measure the quality of geotagged data. These data are voluminous hence there should be an automatic method or model to check the accuracy. As per our best knowledge, based on the published researches, there is no previous work available that can grade the POI data based on its accuracy so that users can decide the risk involved in using incomplete or less accurate data [1]. Therefore, the proposed research work has great significance in terms of quality assessment of POI data available on map websites or geo portals. The proposed approach includes multi-sourced verification methods to overcome the limitation of each source where we included ground truth data, web data, and some inferential data to check the accuracy of the tagged data. The proposed system categorizes the POI data into four different classes based on their accuracy level. The risk involved in using data from each category can be assessed based on the use case. Further we propose the fine-tuned predictive model to predict the appropriate class of the data based on its accuracy using state-of-the-art methods and techniques. The main contribution of this paper is given below:

- Defining classification criteria for different target labels for the desired dataset (Section III).
- Data preparation and conversion for desired dataset (Section IV).
- Improve the dataset after removing class imbalance and non-standardization issues (Section V).
- Implementing multiple learning algorithms to get the best model suitable for this dataset (Section VI).
- Fine-tuning of the hyperparameters for the suitable learning model (Section VII).

The rest of the paper is organized into eight sections: Section II describes the related work; Section III gives the problem description in detail along with the terminologies used in the paper. In Section IV, detailed data preparation processes are discussed. Model building is discussed in Section V. Section VI is about the experiments and results discussion. Fine-tuning of hyperparameters is explained in Section VII and at last, we conclude the work.

II. Related Work

Crowdsourced data available on geo portals are huge data with great potential where it can be used for better citizen-centric services. Therefore, it becomes important to analyze this data deeply to harness its power for robust applications. During this research work, we have explored various research papers related to POI data verification and Predictive machine learning models. This in-depth study is presented in three main categories as "POI data verification", "Imbalance class dataset" and "Predictive modeling". Research work related to each category is given below:

A. POI Data Verification

Researchers assessed the quality of POI data available on Different platform like Facebook, flicker, etc. and concluded that different level of accuracy is required for different use cases [1]. However, they considered very few parameters which are not sufficient to assess the accuracy of the POI. Authors searched the missing POI on Google Maps by using unsupervised learning methods where they extracted addresses from the web and enhanced the missing POI [2]. Outdated POI was searched using a semi-supervised method which includes decision tree classifier and Adaboost [3]. Finding the obsolete POI is an important work as the outdated POI may mislead the user. The authors formulated the Integer Linear Programming approach to identify the visited POI which is useful in handling the POI recommendation [4]. Researchers proposed detection of the POI boundaries using Boundary-dependent Explicit Semantic analysis [5]. POI data density is used to identify the urban functional area which can be used for future planning in that region [6]. Boundaries of the commercial centers were also identified using POI data [7]. These works are useful in planning regional events and business growth. Many times, multiple tags are available for the same data hence it is important to remove the duplicate information. The author suggested a non-negative matrix factorization method based on the maximum likelihood estimation to find the similarity between different labels assigned to the same POI [8]. Researchers work on the POI data organization using multidimensional ranking organization to arrange the POI data which can be retrieved easily during the recommendation process [9]. Mapping the POI data on correct location and category was suggested which can be used in POI recommendation [10]. Authors provide the statistics for POI accuracy on available sources that is useful in future development in this domain [11]. Lots of work is being done in the POI recommendation area [12-13]. POI data classification was done using various techniques including bounding box methods, land use identification and gazetteer information [14-17]. During the quality assessment for POI data, the researcher faced the limitation in obtaining the reference data [18]. We didn't find many works done in POI verification or accuracy measurement of the crowdsourced geotagged POI data.

B. Imbalance Class Dataset

There is no readily dataset available which can claim for corresponding reference data. Hence, to conduct any experiment in this direction, desired dataset must be prepared. Data set preparation is done using real data available on different map services and geo portals as explained in Section IV. This dataset is imbalanced where each classification category has different number of instances that induce significant differences in accuracy of the predictions. Therefore, the available method for balancing such imbalanced dataset is explored. Authors experimented on various datasets and indicated the difficulties in using imbalanced data and summarized various available methods to make the classes balanced [19-21]. They proposed the method to identify the boundary case example from the given data set and discussed the sampling methods like SMOTE, SPIDER etc. Some researchers assessed the multi label classification problem in imbalanced dataset and suggested the metrics SCUMBLE and SCUMBLELbl to assess the hard label problems [22]. They proposed the new method of resampling to overcome minority class issues. Metrics to assess the performance of the model built using imbalance dataset are suggested [23]. CatBoost and LogitBoost are superior for such datasets and MMMC is the best metrics to judge the performance [24]. Author suggested skew normalized scores to assess the performance of skewed data [25].

C. Predictive Modelling

Predictive modelling approaches were explored for different domains as we hardly found the research work done for accuracy prediction of POI data. Authors proposed for popularity prediction of the POI data for recommendation system [26]. POI classification using machine learning approaches were studied [27-28]. Some researchers compared POI data extracted from different sources and observed the variations [29]. Machine learning methods were used to compare POI datasets extracted from different sources. Most of the work done in this domain is related to POI recommendation [30-31]. As there are very few research articles available with respect to POI data and predictive modelling, we explored other domains where predictive modelling was applied [32-35].

D. Review Summary

Geotagging is the important activity to generate POI data and many researchers worked in this domain [36-39]. Some authors suggested the techniques to inference the geographical location based on text, image and categories [40-47]. Authors explained that even incomplete data can also be useful hence we divided the dataset into four classes and proposed a method to categorize the POI in a suitable class [48]. As per our best knowledge, such categorization and labeling of POI data available on geoportal are not tried before using predictive modeling where multiple parameters including ground truth data are used. Our proposed solution is considering many sources to extract the relevant features from the trusted sources such as Layout urban planning and development map of that region and the repositories of public service POI data. The

proposed model is based on state-of-the-art methods and techniques for a better solution.

III. Problem Setting

First, we give a brief introduction of the terms and abbreviations, and then formulate the problem for measuring the accuracy of geotagged data. Categorical Classes are defined in this section according to the net value calculated based on the penalty and gain as described in detail in Section IV.

A. Definitions

- Crowd sourcing is a data collection technique where a large number of people share the data on a common platform and collected data are called crowdsourced data.
- Geotagged POI data determines the Point of interest data that includes geographical information of a place along with other relevant metadata such as name, address, category, latitude, and longitude. It is used as tagged data interchangeably at some places.
- Ground truth data are the location data that are verified and provided by trusted contributors. This is considered as the reference data in this study.
- Category is a place type e.g. education, healthcare, bank, religious place, etc.
- In this work, the user is a contributor who has geotagged the data on some geoportal or mapping website. It could be anyone like an owner, local guide, and general user.
- Web Sources are the websites that give some information about the searched geotagged place. It must include the name and address of the searched tagged data.
- The geotagged data are said to be verified if some process is applied to check its accuracy which categories these in one of the classes specified as "Correct", "Incorrect", "Partially Correct" and "Almost Correct".

List of notations are given in Table I.

B. Tagging the Label

As we explained earlier, there is no required dataset available to work on this problem hence, we have prepared the dataset and defined the criteria to label the target parameter based on its class. Proposed work is our extended work where we conceptualized the verification Model for POI data [49]. In this model, ground truth data are involved in verification where the layout plan and the trusted repositories are considered as GT data. Corresponding geotagged data are extracted from geo portals or map websites that are considered as CS data. A layout plan was georeferenced and the detail of each asset is converted in to attribute table. The main features of the asset like name, address, pin code, latitude, longitude, and category are incorporated in model building. The formulation for accuracy label tagging is explained in the below subsection.

TABLE I. Notations

Notation	Description
GT	Ground Truth
POI	Point of Interest
Lat	Latitude
Long	Longitude
GIS	Geographical-Information System
Pincode	Postal Code
CS	Crowdsourced
Sr No	Serial Number

C. Problem Definition

Let's take G as the set of GT data and P as the set of POI data available on the geoportals or Map websites. Assuming A as the set of features or metadata, associated with POI data. L is the set of labels given to the target, based on the net values. "n" is the total number of records available in the data set. T is a set of target values as shown in "(1)" and "(2)". These sets and their relationship are described in below equations:

$$G=\{g_i\},\ where\ i=1\ to\ n;\ P=\{p_i\},\ where\ i=1\ to\ n;\ A=\{a_j\},\ where\ j=1\ to\ x; \quad (1)$$

$$L=\{\text{“correct”, “incorrect”, “partial correct”, “almost correct”}\},\ T=\{t_i\},\ where\ i=1\ to\ n; \quad (2)$$

$$f(g_i, p_i)=\{gain/penalty\},\ \forall a_j\ where\ j=1\ to\ x \quad (3)$$

$$\sum_{j=1}^{k}(gain)-\sum_{j=1}^{k}(penalty)=netvalue,\ \forall a_j\ where\ j=1\ to\ x \quad (4)$$

$$\forall t_i,\ f(t_i(netvalue)) \in L.\ where,\quad I_i=\begin{cases} incorrect, if\ netvalue \le 5 \\ partial\ correct, if\ 5 > netvalue \le 10 \\ almost\ correct, if\ 10 > netvalue \le 15 \\ correct, if\ netvalue > 15 \end{cases} \quad (5)$$

Comparison functions are applied on GT and corresponding CS data to get the gain and penalty as shown in "(3)". The parameters that provide positive support towards accuracy, referred to as "gain" and the negatively contributing parameters are referred to as "penalty". Details of the functions are covered in the next section. Once the net value is calculated based on the "(4)", entire dataset is labeled with four different classes as described in "(5)".

IV. Data Preparation

The success of any prediction model is entirely dependent on the training dataset. Hence, it is essential to get a correct, valid, consistent, and real dataset to get the actual image of the real data. For the proposed model, we need the GT data and the corresponding CS data. Therefore, we prepared the desired dataset by following the Process shown in Fig. 1. There are four main modules involved in dataset preparation. GT data are collected via two ways such as layout plans constructed for urban development and trusted repositories. For each GT data entity, CS data are extracted from geo portals and map websites using API provided by these service providers. Important features of both the dataset are name, address, pin

code, category, Latitude, Longitude, Geo boundary, Asset area, user profile, Time stamp, and reviews.

Intersecting features of these two datasets were selected for comparison and few strong evidential features were also added like contributor's profile data, distinct web sources and the freshness of the data. Contributor's profile exposes the locality where it belongs to. So, the chances are higher to provide correct POI information from that place by the contributor. A number of distinct web sources are directly proportional to the accuracy due to the availability of the same information at various sources. The latest timestamp found in the web source data provides the measurement of the freshness of the POI data. Logic for web scrapping to extract the additional features is explained in the flowchart shown in Fig. 2. Search engine URL and the search string having the name and address of the POI are provided to the web scraping process. HTML is parsed using Beautiful Soup Library of Python, and all the distinct URLs are fetched. Once the entire URL list is processed, the count will be returned. Specifically, we obtained the latest date information found in the web sources to measure the freshness.

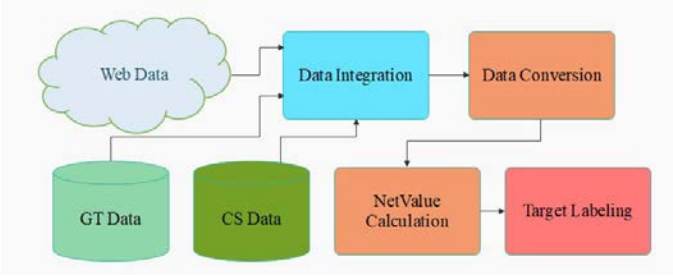

Fig. 1. Flow Diagram of Data Preparation.

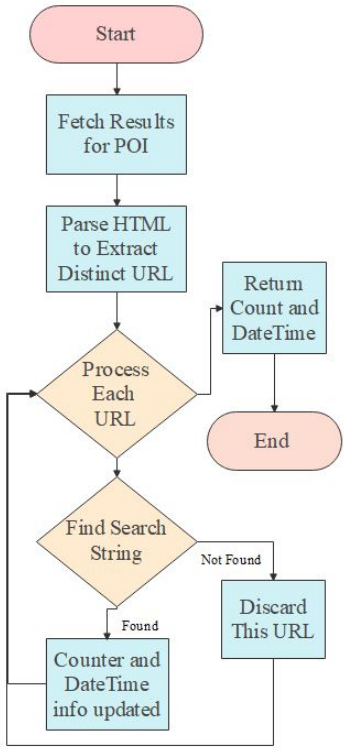

Fig. 2. Web Scrapping Process to Get Additional Information.

The integrated data are processed further to remove the incomplete, erroneous, and duplicate data in the data preparation module. Sample record of this integrated data is shown in Table II. Various comparison methods are applied to compare different features such as wordnet corpus of NLTK is used to compare the category data to consider semantic similarities [50]. Cosine similarity method is used for exact matching. For measuring the deviation in a location with respect to Geo Spatial Points, Haversine formula is used where we obtained the variation in Km. Sample data of comparison outcome values for two records R1 and R2 are shown in Table III. Gain and Penalty are required to calculate the Net value as given in "(4)". These are explained below.

1) Gain: Similarity Score is considered as gain because it supports the correctness of the data. A higher similarity score means more accurate data. Therefore, Name similarity, Address similarity, Pin code similarity, and a category similarity score are treated as gain values. Similarly, the higher web sources count is a positive sign of correctness. Hence it should be added to the gain. Higher user type is inclined towards more trusted users so it is also taken as gain. These user types are general user, local guide, surveyor, owner, or admin. General user has less credential and admin has the highest credential. These values are from 1 to 5. The higher the number, the better the chances to get the correct information.

2) Penalty: Higher variation in data pushes it towards an incorrect state. Therefore, variation is considered as a penalty. Location variation is the important factor to decide the correctness of the data. Using the haversine formula, variation distance is calculated and this deviation is subtracted from the threshold value which is considered as 1 km for this POC just for simplicity. Similarly, Staleness score is also taken as a penalty because a larger value signifies the staleness of the data that is inversely proportional to the accuracy. Hence it is added to the penalty.

After plug-in all the values of penalty and gain in "(4)", the net value is calculated which will be further considered to categorize the target data. Target values are categorized using "(5)" and the entire dataset is classified into four categories as explained in Section III.

TABLE II. SAMPLE DATA AFTER COMPARISON

Feature	Data value
Sr No	4973
Name Similarity	0.1176470
Add Similarity	0.300552458
PIN Similarity	1
Location Variation	533.690
Category Similarity	1
User Type Coding	3
Web Src Count	7
Staleness value	0

TABLE III. SAMPLE DATA AFTER TARGET LABELLING

Feature	Data value(R1)	Data Value(R2)
Sr_No	4973	6237
Name Similarity	0.1176470	1
Add Similarity	0.300552458	0.527072476
PIN Similarity	1	1
Location Variation	533.690	-0.77525
Category Similarity	1	1
User Type Coding	3	3
Web Src Count	7	8
Staleness value	0	0
Net Value	-520.27180484	15.30
TargetLabel	Incorrect	Correct

V. MODEL BUILDING

The Statistical analysis is required to understand the characteristics of the dataset to choose the appropriate approach for model building. Details of the statistical analysis and the treatment applied to the dataset are given in the following subsections.

A. Data Analysis

Once the data set is prepared as explained in the above section, JASP 0.14.1 software is used for statistical analysis to understand the characteristics of the data as displayed in Table IV.

Data shown in this table provides the basic statistical values including valid record, missing values, the mean value in each case, standard deviations, skewness, error in skewness, minimum values, and maximum value. There are lots of differences in the number of records in each class and the class "Correct" is a minority class. The range of the values is also wide which can impact the perfect model building. For some algorithms, higher values may dominate and results could be biased. The distribution graph is plotted for each feature with respect to each class specified in the target value. One of the graphs is shown in Fig. 3 and the detail of the other attributes is given in Appendix I.

TABLE IV. DESCRIPTIVE STATISTICS FOR NET VALUE AND THE TARGET CLASSES

Statistics	Almost Correct	Correct	Partial Correct	Incorrect
Valid	221	13	168	111
Missing	0	0	0	0
Mean	11.544	16.176	7.840	-456.775
Std. deviation	1.150	0.795	1.369	1441.913
Skewness	0.952	0.896	-0.346	-4.587
Std. Error of skewness	0.164	0.616	0.187	0.229
Minimum	10	15	5	-9413.181
Maximum	14.874	18	9.937	4.999

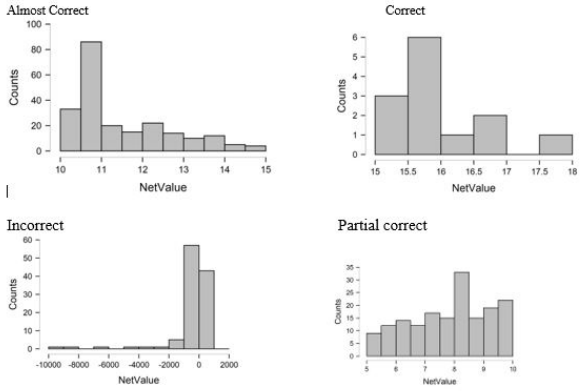

Fig. 3. Distribution Graph for All Four Classes v/s Netvalue.

It is observed that only partial correct data are closer to normal distribution. Distribution for the other classes is skewed as shown in the Fig. 3. High skewness may inject biased results and a false impression of the accuracy of the model could be achieved. Hence, before moving ahead, the dataset must be treated to overcome these issues related to the imbalanced classes and the wider range in the listed features. These processes are explained in the below sections.

B. Standardization and Class Balancing

In skewed data, learning may not be correct as the features having a large magnitude will dominate the objective function. Therefore, learning from other features will not happen impartially. We used MinMaxScaler class of sci-kit learn preprocessing package and set the range between -1 and 1 for required features. The impact of this normalization is shown in the results section.

To make the dataset balance for all the classes, SMOTE oversampling method is used [51]. In this method, interpolation is used to generate the synthetic instance based on the KNn approach. Selecting the number of neighbors is configurable and the default value is taken as 5. After interpolation, a fit-resample is applied to make sure the sufficient number of instances in test and training datasets before dividing it for model building. SMOTE method is available in oversampling package of the "imblearn" library of Python.

C. Model Selection Process

There must be a benchmark to measure the performance of the implemented model and there is a ZeroR classification method that provides the baseline performance for the selected model. It is a basic classification method that predicts the majority class without considering the available predictors in the dataset. After getting the benchmark value, other models need to be applied and compared. As per the statistics and the characteristics of the dataset, it is a multiclass classification problem. Suitable models for this problem are explained below in brief.

1) Logistic regression: Logistic regression is suitable for multi-class classification because it gives the probability of the classes available in the target variable based on the

independent variable. By default, it can classify binary classes however using One vs one and One vs rest make it suitable for multiclass classification as well. This modified version is based on the SoftMax formula for a better approximation of the expected target class.

2) Decision tree: It is a supervised learning algorithm that can be used for classification and regression. It splits the data at the decision node according to certain rules based on the data. The entire dataset is considered at root note and the split logic is applied recursively on this dataset. Leaf node represents the class to which data belongs to. Splitting logic plays a vital role in a tree's classification accuracy.

3) Random forest: Random Forest is an ensemble of multiple decision trees and can be used in both regression and classification. It works on the principle of many weak learners rather than the one strong learner. It starts creating a decision tree corresponding to each instance and based on the majority voting, prediction results are generated. It overcomes the overfitting issue of the decision tree by dropping some features in each tree randomly. Based on the mode where the most frequently occurring outcome for the given data is chosen.

4) K-nearest neighbors: It is a supervised learning classifier based on lazy learning where the entire dataset is used as training data. Euclidian distance is calculated for available data points and the nearest similar items are picked. The accuracy of the model depends upon the data and it may not be able to classify the boundary cases.

5) Multi-Layer perceptron: It is a simple Artificial neural network which works based on the feed-forward principle. There are multiple layers divided into three parts. An input layer, an output layer, and the hidden layers. The input layer receives the input and passes it to the hidden layer where the SoftMax function is used as the activation function. The final prediction task is done at the output layer. There could be more than one hidden layer. The number of neurons on the output layers depends on the number of classes considered for the classification task.

VI. Experimental Set Up

To choose an appropriate algorithm, we have executed multiple tests using Weka 3.9 and all the selected algorithms were applied to the dataset. To apply the above-described algorithm, two well-known approaches, Cross-validation, and split train test was adopted. Both the approaches are given below in brief.

- Cross-validation: It is also known as k-fold cross-validation. In this approach, the dataset is split in to k subsets and the model is trained using k-1 subsets and the kth set is used for testing the performance of the model. The algorithm is executed in k iteration and the final prediction error is the average error observed in each iteration. This approach provides the insight to check the overfitting or selection bias issues.
- Split-Train-Test: The idea is to divide the dataset randomly into two parts where a bigger chunk is used to train the model which is called the training set. The leftover is used as the testing dataset which is unseen for the model and used to evaluate the performance of the model being used.

For cross-validation, K is set as ten which means the dataset is divided into ten parts and a total of ten iterations were executed. In the case of the split-train-test, the dataset is divided into two parts in 70-30 percentage, which means 70% of the data are used for training the model, and the rest 30 % are used as a test data set. The number of iterations is set as ten for this schema. Systematic analysis done for each model is presented in the following subsection.

A. Experimental Results

We analyzed the result obtained in each cycle for the different algorithms being used. Results are discussed below in the ordered way as they are executed.

1) Obtaining base performance: Zero R is used for base performance checks. Performance metrics obtained using ZeroR is set as a benchmark for other models. As shown in the Table V, accuracy and the kappa statistics are decreased after using a balanced class dataset. Hence 20.7009% is set as the benchmark accuracy level for other models. A negative value of Kappa statistics says that this model is very far from the optimum solution.

2) Multiple models application: Selected models were applied on the normalized dataset using both approaches as discussed in the previous section. Results using "Cross vaalidation" are given in Table VI. Logistic, Multilayer Perceptron, IBK, Multiclass Classification, decision tree, and the random forest are applied on the balanced and imbalanced dataset. Results shown in bold are the best performer. Here we can observe that Logistic regression is the best one.

Results obtained using the Split-Train-Test approach is shown in Table VII. As shown in the table, the best performance is shown by Logistic regression. Random forest and the Decision Tree are showing the second and third better performance. We have selected these top performer models for further analysis and once again executed them in python where we got almost same results. Therefore, we have plotted the learning curve for these three models as shown in Fig. 4. For decision tree and random forest models, learning on the small set of data are happening quickly whereas, for rest of the data it is taking time. It could be a case of local maxima so results may be biased. In the case of logistic regression model, the learning curve is much better as it is utilizing the maximum possible records from the dataset before converging. Hence the results obtained through logistic regression are more reliable. Therefore, logistic regression is selected for the final model building. There are certain hyperparameters that can be tuned to improve the performance. These hyperparameters and their impact are explained in the next section.

TABLE V. RESULTS OF ZEROR CLASSIFIER ON TWO DATASETS

Metrics	**Imbalanced Class Data set**	**Balanced Class dataset**
Accuracy	43.0799 %	**20.7009** %
Kappa Statistics	0	**-0. 0573**
MAE	0. 3303	0. 3753
RMSE	0. 4061	0. 4334

TABLE VI. PERFORMANCE RESULTS FOR CROSS VALIDATION SETUP

Model	**With Class balancer (Cross validation)**					**Without Class balancer (Cross validation)**				
	Accuracy	***Kappa***	***RMSE***	***F1***	***Time***	***Accuracy***	***Kappa***	***RMSE***	***F1***	***Time***
Logistic	97.81	0.9709	0.1063	0.978	0.09	99.61	0.9941	0.0491	0.994	1
Multilayer Perceptron	79.61	0.7282	0.2973	0.793	0.36	78.55	0.6641	0.2958	0.776	0.38
IBK	70.16	0.6022	0.3843	0.702	0	69.98	0.5442	0.383	0.700	0
Decision Tree	85.14	0.802	0.2597	0.853	0	86.93	0.8019	0.2419	0.870	0
Random Forest	**87.72**	**0.8363**	**0.2209**	**0.879**	**1**	**90.05**	**0.8484**	**0.1903**	**0.901**	**0.09**

TABLE VII. PERFORMANCE RESULTS FOR SPLIT-TRAIN-TEST SETUP

Model	**With Class balancer (Split-Train-Test)**					**Without Class balancer (Split-Train-Test)**				
	Accuracy	***Kappa***	***RMSE***	***F1***	***Time***	***Accuracy***	***Kappa***	***RMSE***	***F1***	***Time***
Logistic	**97.41**	**0.9654**	**0.1085**	**0.968**	**0.1**	**96.75**	**0.951**	**0.1192**	**0.952**	**0.1**
Multilayer Perceptron	80.09	0.7324	0.2874	0.808	0	77.92	0.659	0.3136	0.767	0
IBK	76.65	0.6864	0.3391	0.752	0.0	73.37	0.588	0.3629	0.722	0.01
Decision Tree	87.27	0.8293	0.2382	0.873	0	86.36	0.791	0.2591	0.863	0.01
Random Forest	**89.75**	**0.8625**	**0.1824**	**0.897**	**0**	**88.961**	**0.8304**	**0.2083**	**0.888**	**0**

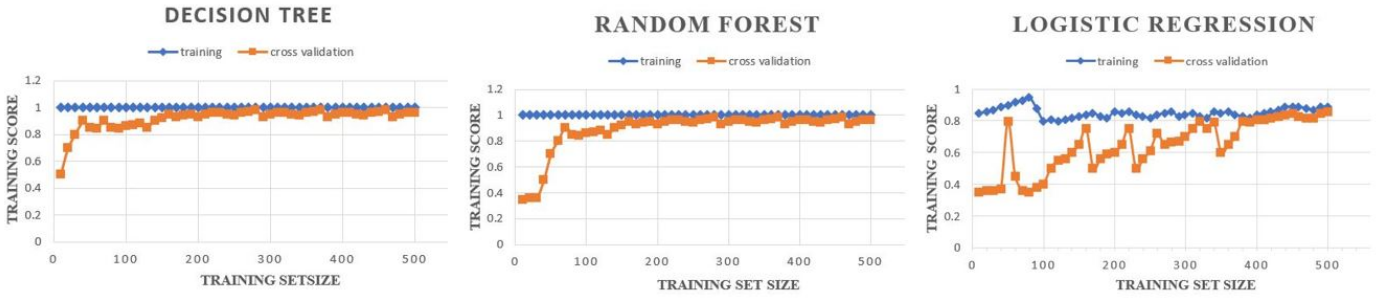

Fig. 4. Learning Curve for Decision Tree, Random Forest and Logistic Regression.

VII. Fine Tuning

Each algorithm has some hyperparameters which needs to be tuned to make the model efficient on the desired data set as there is no pre-stated rule to choose the value of these hyperparameters. For decision tree and random forest models, learning on the small set of data happen quickly whereas, for rest of the data it is taking time. It could be a case of local maxima so results may be biased. In the case of logistic regression model, the learning curve is much better as it is utilizing the maximum possible records from the dataset before converging. Therefore, Logistic regression model is selected for final fine tuning to obtain the best performance.

In the case of logistic regression, we considered below explained three parameters for fine tuning of the model.

A. Solver

It's a mathematical model which helps to calculate the optimum prediction by reducing the loss factor. There are different methods available in logistic regression such as "saga", "lbfgs", "newton-cg", "sag", "liblnear", etc. to achieve the best optimization based on the characteristics of the dataset available to work on. Newton-cg uses the quadratic function for loss minimization which makes it expensive. It can handle the loss in case of multiclass problems easily. Limited-memory Broyden–Fletcher–Goldfarb–Shanno (lbfgs), stores only a few vectors that support the approximations for the next terms. It can also handle the multinomial loss easily and it is used as the default solver in sci-kit learn library of python. Liblinear uses the coordinate descent algorithm for optimization purposes. It applies automatic parameters selection and suitable for large predictors dataset. Stochastic Average Gradient (SAG) uses smooth convex functions and suitable for large data set as it easily gets converge. SAGA is a variant of SAG which supports sparse multinomial logistic regression and suitable for a very large dataset.

B. Regularization

It is also known as the penalty method. It is a way to avoid the overfitting of a model. Regularization overcomes the overfitting issue by adding some bias using tuning parameters. There are multiple methods available that are L1, L2, and Elastic Net. These methods are to regularize the higher coefficient values. L1 uses the absolute magnitude values of the coefficient and ignores the Zero values. L2 takes the square of the magnitude to avoid the sparse coefficient matrix. Elastic net is a combination of L1 and L2 and mostly used where the number of predictors is larger than the number of observations [52]. It groups the strongly correlated predictors so the contribution or removal of these variables is done together.

C. C-Value

This is the inverse of the solver strength. As per the documentation of the sci-kit learn, a smaller value indicates the stronger regularization.

All the solvers don't support each penalizing method. Table VIII shows the solver and the penalty supported by them. From the below table, we observe that penalty L2 is applicable with each solver so we are considering L2 for further analysis with different combinations of c value and the solver choice. The logic for getting the accuracy with standard deviation in each case is given in Algorithm 1.

TABLE VIII. Solver and Compatible Regularization Method

Solver	Compatible Regularization Method
Newton – cg	L2
Sag	L2
Saga	L2, Elastic net
Lbfgs	L2

The solver list, penalty method list, and solver strength list are passed as the input parameters. RepeatedStratifiedKFold class from sci-kit learn of python is used as a cross validator for taking different random records for repeating n times.

ALGORITHM 1: Analysis of Hyperparameters

```
Input:
solverList= {'newton-cg', 'lbfgs', 'liblinear'},
penaltyList={'l2','none'}, cValues = [100, 10, 1.0, 0.1, 0.01],
executionScheme= {simple logistic, k fold logistic, stratified k
fold}, dataset
Output:
outcomeList
        for each cValues do
         for each executionScheme
         cv = createInstanceOf_ExecutionScheme ()
         gs = GridSearchCV (estimator, param_grid, cv, scoring,
error_score)
         fitGridSearchOnDataset ()
         SetOutcome (accuracy, stdDeviation)
         endFor
 endFor
output outcomeList
```

After that, GridSearchCV of model selection package from sci-kit learn is used for executing the process for various parameters. It takes a model function as the estimator, param_grid as the list or dictionary of the parameters which need to be applied. It also takes the cross-validator estimator, a scoring list that may have multiple scoring parameters and an error score. We have tested for penalty "L2" and 'none' as the other options are not applicable. Best accuracy is observed with Newton-cg and the C value as 100. Accuracy decreases for other values of the C parameter. Solver Sag and Saga show minor change and low accuracy. If the penalty is set as "None" then Liblinear is out from the choice as it doesn't support any regularization method apart from L2. For the rest of the solver, again the Newton-CG performed the best with 100 as the C value. L2 regularization method gives the best result for Newton-cg solver and 100 as the strength value however it must be verified the same with a large dataset.

VIII Conclusion

In this research, the proposed predictive model was built and fine-tuned via exhaustive experiments done on the

prepared dataset. Results show that the performance varies for an imbalanced dataset which is just half when applied to a balanced dataset. It removes the overfitting chances that lead for more accurate, nonbiased results. We also applied different models on the data set for two different approaches known as Cross-validation and Split-train-test to obtain the best model. Logistic regression is found as the best performer among six different models however random forest and decision tree were also good but they were suffered from the local maxima issue. Hence the logistic regression is chosen for the final model and it was fine-tuned for various parameters using exhaustive experiments. The final model is obtained with Newton-CG solver, L2 penalty and 100 as the solver strength parameter for this dataset. The proposed model gives 96.40% accuracy on unseen data. This is a novel predictive model, based on multiple sources of information to judge the accuracy of the geotagged data. This work provides the way to move forward for various use cases depends on the geotagged data for better development work for citizens. There are many preprocessing steps involved to make the data useful for prediction. As of now, these processes are discrete and costly in terms of execution time. Going forward, we are planning to increase the dataset with the small threshold value for location variation and build a robust pipeline to cover the data preprocess work using the state-of-the-art framework available for the machine learning pipeline. Proposed work opens the way to ensure the quality of geotagged data available on mapping websites and geoportals.

VIII. Supplementary Material

Detailed descriptive statistical analysis of dataset is provided in Appendix I.

References

[1] Jonietz, David, and Alexander Zipf. “Defining Fitness-for-Use for Crowdsourced Points of Interest (POI).” ISPRS International Journal of Geo-Information 5, no. 9 (August 24, 2016): 149. doi:10.3390/ijgi5090149.

[2] Chang, Chia-Hui, Hsiu-Min Chuang, Chia-Yi Huang, Yueng-Sheng Su, and Shu-Ying Li. “Enhancing POI Search on Maps via Online Address Extraction and Associated Information Segmentation.” Applied Intelligence 44, no. 3 (October 15, 2015): 539–556. doi:10.1007/s10489-015-0707-5.

[3] Chuang, Hsiu-Min, and Chia-Hui Chang. “Verification of POI and Location Pairs via Weakly Labeled Web Data.” Proceedings of the 24th International Conference on World Wide Web (May 18, 2015). doi:10.1145/2740908.2741715.

[4] Suzuki, Jun, Yoshihiko Suhara, Hiroyuki Toda, and Kyosuke Nishida. “Personalized Visited-POI Assignment to Individual Raw GPS Trajectories.” ACM Transactions on Spatial Algorithms and Systems 5, no. 3 (September 25, 2019): 1–28. doi:10.1145/3317667.

[5] Bui, Thanh-Hieu, Yong-Jin Han, Seong-Bae Park, and Se-Young Park. “Detection of POI Boundaries through Geographical Topics.” 2015 International Conference on Big Data and Smart Computing (BIGCOMP) (February 2015). doi:10.1109/35021bigcomp.2015.7072827.

[6] Hu, Sheng, Zhanjun He, Liang Wu, Li Yin, Yongyang Xu, and Haifu Cui. “A Framework for Extracting Urban Functional Regions Based on Multiprototype Word Embeddings Using Points-of-Interest Data.” Computers, Environment and Urban Systems 80 (March 2020): 101442. doi: 10.1016/j.compenvurbsys.2019.101442.

[7] Lifang, Zhu, Chen Yixin, Liu Yang, Zhang Yue, and Wang Jing. “POI Data Applied in Extracting the Boundary of Commercial Centers.” 2017 IEEE 2nd International Conference on Big Data Analysis (ICBDA)((March 2017). doi:10.1109/icbda.2017.8078693.

[8] Zhou, Jingbo, Shan Gou, Renjun Hu, Dongxiang Zhang, Jin Xu, Airong Jiang, Ying Li, and Hui Xiong. “A Collaborative Learning Framework to Tag Refinement for Points of Interest.” Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (July 25, 2019). doi:10.1145/3292500.3330698.

[9] Liu, Lei, Jingwen Li, and Suxian Ye. “Research on the Method of Constructing POI Data Model Based on AP-MRO.” 2018 IEEE 4th Information Technology and Mechatronics Engineering Conference (ITOEC) (December 2018). doi:10.1109/itoec.2018.8740599.

[10] Li, Yurui, Hongmei Chen, Lizhen Wang, and Qing Xiao. “POI Representation Learning by a Hybrid Model.” 2019 20th IEEE International Conference on Mobile Data Management (MDM) (June 2019). doi:10.1109/mdm.2019.00010.

[11] Hochmair, Hartwig H., Levente Juhász, and Sreten Cvetojevic. “Data Quality of Points of Interest in Selected Mapping and Social Media Platforms.” Progress in Location Based Services 2018 (December 9, 2017): 293–313. doi:10.1007/978-3-319-71470-7_15.

[12] Liu, Shudong. “User Modeling for Point-of-Interest Recommendations in Location-Based Social Networks: The State of the Art.” Mobile Information Systems 2018 (2018): 1–13. doi:10.1155/2018/7807461.

[13] Massimo, David, and Francesco Ricci. “Harnessing a Generalised User Behaviour Model for Next-POI Recommendation.” Proceedings of the 12th ACM Conference on Recommender Systems (September 27, 2018). doi:10.1145/3240323.3240392.

[14] Janowicz, K., and C. Keßler. “The Role of Ontology in Improving Gazetteer Interaction.” International Journal of Geographical Information Science 22, no. 10 (October 2008): 1129–1157. doi:10.1080/13658810701851461.

[15] Jiang, Suhui, Yu Kong, and Yun Fu. “Deep Geo-Constrained Auto-Encoder for Non-Landmark GPS Estimation.” IEEE Transactions on Big Data 5, no. 2 (June 1, 2019): 120–133. doi:10.1109/tbdata.2017.2773096.

[16] Xing, Hanfa, Yuan Meng, Dongyang Hou, Fangjie Cao, and Haibin Xu. “Exploring Point-of-Interest Data from Social Media for Artificial Surface Validation with Decision Trees.” International Journal of Remote Sensing 38, no. 23 (August 23, 2017): 6945–6969. doi:10.1080/01431161.2017.1368101.

[17] Yuanrong He, Yuanmao Zheng, Jian Deng, and Huoping Pan. “Design and Implementation of a POI Collection and Management System Based on Public Map Service.” 2016 Fourth International Conference on Ubiquitous Positioning, Indoor Navigation and Location Based Services (UPINLBS) (November 2016). doi:10.1109/upinlbs.2016.7809971.

[18] De Tré, Guy, Daan Van Britsom, Tom Matthé, and Antoon Bronselaer. “Automated Cleansing of POI Databases.” Quality Issues in the Management of Web Information (2013): 55–91. doi:10.1007/978-3-642-37688-7_4.

[19] Krawczyk, Bartosz. “Learning from Imbalanced Data: Open Challenges and Future Directions.” Progress in Artificial Intelligence 5, no. 4 (April 22, 2016): 221–232. doi:10.1007/s13748-016-0094-0.

[20] Napierala, Krystyna, and Jerzy Stefanowski. “Types of Minority Class Examples and Their Influence on Learning Classifiers from Imbalanced Data.” Journal of Intelligent Information Systems 46, no. 3 (July 9, 2015): 563–597. doi:10.1007/s10844-015-0368-1.

[21] Vluymans, Sarah. “Learning from Imbalanced Data.” Studies in Computational Intelligence (November 24, 2018): 81–110. doi:10.1007/978-3-030-04663-7_4.

[22] Charte, Francisco, Antonio J. Rivera, María J. del Jesus, and Francisco Herrera. “Dealing with Difficult Minority Labels in Imbalanced Mutilabel Data Sets.” Neurocomputing 326–327 (January 2019): 39–53. doi: 10.1016/j.neucom.2016.08.158.

[23] Chawla, Nitesh V. “Data Mining for Imbalanced Datasets: An Overview.” Data Mining and Knowledge Discovery Handbook (n.d.): 853–867. doi:10.1007/0-387-25465-x_40.

[24] Tanha, Jafar, Yousef Abdi, Negin Samadi, Nazila Razzaghi, and Mohammad Asadpour. “Boosting Methods for Multi-Class Imbalanced Data Classification: An Experimental Review.” Journal of Big Data 7, no. 1 (September 1, 2020). doi:10.1186/s40537-020-00349-y.

[25] Jeni, Laszlo A., Jeffrey F. Cohn, and Fernando De La Torre. “Facing Imbalanced Data--Recommendations for the Use of Performance

Metrics." 2013 Humaine Association Conference on Affective Computing and Intelligent Interaction (September 2013). doi:10.1109/acii.2013.47.

[26] Yang, Yang, Yaqian Duan, Xinze Wang, Zi Huang, Ning Xie, and Heng Tao Shen. "Hierarchical Multi-Clue Modelling for POI Popularity Prediction with Heterogeneous Tourist Information." IEEE Transactions on Knowledge and Data Engineering 31, no. 4 (April 1, 2019): 757–768. doi:10.1109/tkde.2018.2842190.

[27] Jiang, Shan, Ana Alves, Filipe Rodrigues, Joseph Ferreira, and Francisco C. Pereira. "Mining Point-of-Interest Data from Social Networks for Urban Land Use Classification and Disaggregation." Computers, Environment and Urban Systems 53 (September 2015): 36–46. doi: 10.1016/j.compenvurbsys.2014.12.001.

[28] Raj, Pethuru, and Chellammal Surianarayanan. "Digital Twin: The Industry Use Cases." The Digital Twin Paradigm for Smarter Systems and Environments: The Industry Use Cases (2020): 285–320. doi: 10.1016/bs.adcom.2019.09.006.

[29] Zhou, Yang, Mingjun Wang, Chen Zhang, Fu Ren, Xiangyuan Ma, and Qingyun Du. "A Points of Interest Matching Method Using a Multivariate Weighting Function with Gradient Descent Optimization." Transactions in GIS 25, no. 1 (October 5, 2020): 359–381. doi:10.1111/tgis.12690.

[30] Yu, Ruiyun, Dezhi Ye, and Jie Li. "RePiDeM: A Refined POI Demand Modeling Based on Multi-Source Data*." IEEE INFOCOM 2020 - IEEE Conference on Computer Communications (July 2020). doi:10.1109/infocom41043.2020.9155294.

[31] Zhang, Zhiqian, Chenliang Li, Zhiyong Wu, Aixin Sun, Dengpan Ye, and Xiangyang Luo. "NEXT: a Neural Network Framework for Next POI Recommendation." Frontiers of Computer Science 14, no. 2 (August 30, 2019): 314–333. doi:10.1007/s11704-018-8011-2.

[32] Asniar, and Kridanto Surendro. "Predictive Analytics for Predicting Customer Behavior." 2019 International Conference of Artificial Intelligence and Information Technology (ICAIIT) (March 2019). doi:10.1109/icaiit.2019.8834571.

[33] Bindra, Simranjeet Kour, Akshay Girdhar, and Inderjeet Singh Bamrah. "Outcome Based Predictive Analysis of Automatic Question Paper Using Data Mining." 2017 2nd International Conference on Communication and Electronics Systems (ICCES) (October 2017). doi:10.1109/cesys.2017.8321154.

[34] Lu, Yafeng, Robert Kruger, Dennis Thom, Feng Wang, Steffen Koch, Thomas Ertl, and Ross Maciejewski. "Integrating Predictive Analytics and Social Media." 2014 IEEE Conference on Visual Analytics Science and Technology (VAST) (October 2014). doi:10.1109/vast.2014.7042495.

[35] Wazurkar, Parth, Robin Singh Bhadoria, and Dhananjai Bajpai. "Predictive Analytics in Data Science for Business Intelligence Solutions." 2017 7th International Conference on Communication Systems and Network Technologies (CSNT) (November 2017). doi:10.1109/csnt.2017.8418568.

[36] Arenga, Delan Zoe H., Jennifer C. Dela Cruz, Franch Maverick A. Lorilla, and Paul A. Tangian. "Cloud-Based Flora Repository System with Geo-Location Mapping for Mt. Hamiguitan Sanctuary Exploration." 2018 IEEE Region Ten Symposium (Tensymp) (July 2018). doi:10.1109/tenconspring.2018.8692056.

[37] Singh, Sanket Kumar, and Davood Rafiei. "Strategies for Geographical Scoping and Improving a Gazetteer." Proceedings of the 2018 World Wide Web Conference on World Wide Web - WWW '18 (2018). doi:10.1145/3178876.3186078.

[38] Yoshikatsu, Nagata. "Geographic Names on Old Maps of Early 20th Century Toward a Spatio-Temporal Gazetteer: A Study on Their Accuracy in Northeast Thailand." 2017 Pacific Neighborhood Consortium Annual Conference and Joint Meetings (PNC) (November 2017). doi:10.23919/pnc.2017.8203528.

[39] Yu Liu, Runqiang Li, Kaichen Chen, Yihong Yuan, Lingli Huang, and Hao Yu. "KIDGS: A Geographical Knowledge-Informed Digital Gazetteer Service." 2009 17th International Conference on Geoinformatics (August 2009). doi:10.1109/geoinformatics.2009.5293495.

[40] Abbas, S. Syed Ameer, S. K. Ajin, and P. Deepak Sundar. "Realization of Multimodal Geo-Tag Using ARM-A53 with Python." 2017 International Conference on Intelligent Computing and Control (I2C2) (June 2017). doi:10.1109/i2c2.2017.8321905.

[41] Jurgens, T. Finethy, J. Mccorriston, Y. T. Xu, and D. Ruths, "Geolocation Prediction in Twitter Using Social Networks: A Critical Analysis and Review of Current Practice," pp. 188–197. doi=10.1.1.700.5386.

[42] Niu, Wei, James Caverlee, Haokai Lu, and Krishna Kamath. "Community-Based Geospatial Tag Estimation." 2016 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM) (August 2016). doi:10.1109/asonam.2016.77522.

[43] Qian, Xueming, Yisi Zhao, and Junwei Han. "Image Location Estimation by Salient Region Matching." IEEE Transactions on Image Processing 24, no. 11 (November 2015): 4348–4358. doi:10.1109/tip.2015.2462131.

[44] Radke, Mansi A., Nitin Gautam, Akhil Tambi, Umesh A. Deshpande, and Zareen Syed. "Geotagging Text Data on the Web—A Geometrical Approach." IEEE Access 6 (2018): 30086–30099. doi:10.1109/access.2018.2843814.

[45] Ssin, Seung Youb, Joanne E. Zucco, James A. Walsh, Ross T. Smith, and Bruce H. Thomas. "SONA: Improving Situational Awareness of Geotagged Information Using Tangible Interfaces." 2017 International Symposium on Big Data Visual Analytics (BDVA) (November 2017). doi:10.1109/bdva.2017.8114625.

[46] Utomo, Muhammad Nur Yasir, Teguh Bharata Adji, and Igi Ardiyanto. "Geolocation Prediction in Social Media Data Using Text Analysis: A Review." 2018 International Conference on Information and Communications Technology (ICOIACT) (March 2018). doi:10.1109/icoiact.2018.8350674.

[47] Zhang, Hang, Lin Li, Wei Hu, Wenjing Yao, and Haihong Zhu. "Visualization of Location-Referenced Web Textual Information Based on Map Mashups." IEEE Access 7 (2019): 40475–40487. doi:10.1109/access.2019.2907570.

[48] Zhang, Liming, and Dieter Pfoser. "Using OpenStreetMap Point-of-Interest Data to Model Urban change—A Feasibility Study." Edited by Michael Szell. PLOS ONE 14, no. 2 (February 25, 2019): e0212606. doi: 10.1371/journal.pone.0212606.

[49] Sharma, M., Bothale, V., Bundele, M. and Nawal, M. "Geotagging: Systematic Anatomization and Conceptual Model for POI Verification." International Journal of Innovative Technology and Exploring Engineering 9, no. 11 (September 10, 2020): 339–348. doi:10.35940/ijitee.k7820.0991120.

[50] Princeton University "About WordNet." WordNet. Princeton University. 2010.

[51] Chawla, N. V., K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer. "SMOTE: Synthetic Minority Over-Sampling Technique." Journal of Artificial Intelligence Research 16 (June 1, 2002): 321–357. doi:10.1613/jair.953.

[52] Scikit-learn: Machine Learning in Python, Pedregosa et al., JMLR 12, pp. 2825-2830, 2011.

APPENDIX I

Attribute wise Descriptive Statistical Analysis of the original dataset is presented here. The number of records is not the same for each class as shown in the respective tables. The range for minimum and maximum values is also varying for some attributes. Hence the class balancing and standardization methods were applied before model building.

A. *Name Similarity*

Name of the POI is matched in GT data and corresponding CS data, and compared using wordnet English language corpus. Very few records have exact same name in the Ground Truth data set and the Crowdsourced data available on Geoportals or mapping websites. Comparison score varies from 0 to 1 where 1 shows the perfect matching as shown in Table IX.

TABLE I. DESCRIPTIVE STATISTICS FOR NAME SIMILARITY BASED ON THE TARGET CLASSES

Descriptive Statistics	Almost Correct	Correct	Incorrect	Partial Correct
Valid	221	13	168	111
Missing	0	0	0	0
Mean	0.955	0.919	0.785	0.812
Std. deviation	0.151	0.031	0.326	0.311
Minimum	0	0.889	0.118	0
Maximum	1	1	1	1

B. *Address Similarity*

Address of the POI is matched in GT data and corresponding CS data and compared using wordnet English language corpus of NLTK library of Python. Here we considered the semantic meaning of the words like clinic, hospital, and medical institute all were considered as matching. Therefore, the mean value in case of almost correct is 0.715 which is acceptable as shown in Table X.

TABLE II. DESCRIPTIVE STATISTICS FOR ADDRESS SIMILARITY BASED ON THE TARGET CLASSES

Descriptive Statistics	Almost Correct	Correct	Incorrect	Partial Correct
Valid	221	13	168	111
Missing	0	0	0	0
Mean	0.715	0.919	0.335	0.453
Std. deviation	0.277	0.160	0.216	0.216
Minimum	0.087	0.527	0	0
Maximum	1	1	1	1

C. *PIN Similarity*

Pin code of the POI is matched in GT data and corresponding CS data and compared using the cosine similarity method. Here we considered exact matching based on the number of digits. As pin code in India has six digits where initial two digits give the information about the state code so if these two digits are not matching then it must go in incorrect however if only last digit is varying that means the variation is within the same locality so it can be considered as Almost Correct as shown in Table XI.

TABLE III. DESCRIPTIVE STATISTICS FOR PIN SIMILARITY BASED ON THE TARGET CLASSES

Descriptive Statistics	Almost Correct	Correct	Incorrect	Partial Correct
Valid	221	13	168	111
Missing	0	0	0	0
Mean	0.920	1	0.635	0.805
Std. deviation	0.254	0	0.470	0.357
Minimum	0	1	0	0
Maximum	1	1	1	1

E. Distance Variation

Distance variation is the GPS location variation in GT data and corresponding CS data and comparison is done using the Haversine distance calculation method. For implementing the idea, we have kept the threshold as one kilometer just for simplicity. If the variation is within the threshold value, it is considered as matching otherwise not. There is a large difference between minimum and maximum values which needs to be standardized before using in model building as shown in Table XII.

TABLE IV. DESCRIPTIVE STATISTICS FOR DISTANCE VARIATION BASED ON THE TARGET CLASSES

Descriptive Statistics	Almost Correct	Correct	Incorrect	Partial Correct
Valid	221	13	168	111
Missing	0	0	0	0
Mean	-0.778	-0.958	464.041	-0.109
Std. deviation	0.334	0.085	1442.937	1.173
Minimum	-1	-1	-1	-1
Maximum	1.292	-0.775	9156.581	4.639

F. Category Similarity

Category of the POI is matched in GT data and corresponding CS data and compared using wordnet English language corpus of NLTK library of Python. Here we considered the semantic meaning of the words as taken in case of Name and address contents. As the mean value suggests, even for the incorrect data, most of the records shows the similarity value more than 0.7 as shown in Table XIII.

TABLE V. DESCRIPTIVE STATISTICS FOR CATEGORY SIMILARITY BASED ON THE TARGET CLASSES

Descriptive Statistics	Almost Correct	Correct	Incorrect	Partial Correct
Valid	221	13	168	111
Missing	0	0	0	0
Mean	0.809	1	0.736	0.762
Std. deviation	0.314	0	0.347	0.352
Minimum	0.111	1	0	0
Maximum	1	1	1	1

G. UserType Coding

There are five types of contributors/users such as general user, local guide, surveyor, owner, or admin who tag the data. Admin is the system admin who is inserting verified geotagged data. Hence there are few less chances to get the wrong data. Surveyor is the authorized person who is manually verifying the locations and tagging. They also considered trusted users. The local guide has more knowledge about the vicinity to which they belong. The rest of the users are considered a general user. Therefore, encoding is done based on their trust factor, and the admin is assigned the highest value whereas the general user is given the lowest value within the range of 1 to 5. Statistics are shown in Table XIV.

TABLE VI. DESCRIPTIVE STATISTICS FOR USERTYPECODING BASED ON THE TARGET CLASSES

Descriptive Statistics	Almost Correct	Correct	Incorrect	Partial Correct
Valid	221	13	168	111
Missing	0	0	0	0
Mean	3.059	3.692	1.685	1.738
Std. deviation	1.682	1.316	0.991	0.930
Minimum	1	1	1	1
Maximum	5	5	5	5

I. *WebSrc Count*

Using Web scrapping, POI data are searched on the web and distinct web sources are identified which give the same address and name information about that POI. A number of distinct web sources are directly proportional to the accuracy of the data. However, few exceptional cases may arise due to the new business setup. In these scenarios, other parameters like user type and latency period are considered to categorize the data in one of the four classes. Table XV gives the statistical idea about related data.

TABLE VII. DESCRIPTIVE STATISTICS FOR WEBSRCCOUNT BASED ON THE TARGET CLASSES

Descriptive Statistics	Almost Correct	Correct	Incorrect	Partial Correct
Valid	221	13	168	111
Missing	0	0	0	0
Mean	4.905	9.154	4.090	4.280
Std. deviation	3.266	0.899	2.310	1.876
Minimum	1	8	1	1
Maximum	10	10	10	9

J. *Latency Period*

This attribute gives the idea about the freshness of the data. The latest timestamp is found through the process of web scraping and the data availability on the geoportals. The most recent is considered as better which is calculated by taking the difference in the current data and the latest timestamp. The smaller the difference leads to better accuracy. Statistics of this data are given in Table XVI.

TABLE VIII. DESCRIPTIVE STATISTICS FOR LATENCY PERIOD BASED ON THE TARGET CLASSES

Descriptive Statistics	Almost Correct	Correct	Incorrect	Partial Correct
Valid	221	13	168	111
Missing	0	0	0	0
Mean	0.597	0	1.000	1.119
Std. deviation	0.922	0	1.514	1.218
Minimum	0	0	0	0