Capstone Project: Hand gestures and its recognition and prediction

|  |
| --- |
|  |
|  |
| Team members:  Related image  Moses Bothello Mohan Kohad Naseema Memon Navika Sharma Sahil Kedar |

Information on the Project:

* A Vicon movement catch camera framework was utilized to record 12 users performing 5 hand stances with markers joined to one side glove.
* A total of 12 sensors set on the glove were used to collect data.
* Every instance has a minimum set of 3 markers’ data to be utilized as framework for the hand, and additional 9 different markers which can be included or occluded.
* 3 markers were joined to the thumb with one over the thumbnail and the other two on the knuckles. 2 markers were connected to each finger with one over the fingernail and the other on the joint between the proximal and center phalanx.
* Likewise, because of self-impediment because of the direction and setup of the hand and fingers, numerous records have missing markers. Incidental markers were likewise conceivable because of ancient rarities in the Vicon programming's marker recreation/recording process and different articles in the catch volume. Thus, the quantity of noticeable markers in a record changed extensively.
* At last, any record that contained less than 3 markers was evacuated. The prepared information has all things considered 12 markers for each record and in any event 3.
* Because of the way where information was caught, all things considered, for a given record and client there exists a close to copy record beginning from a similar client. We prescribe in this manner to assess grouping calculations on a forget about one-client premise wherein every client is iteratively forgotten about from preparing and utilized as a test set. One at that point tests the speculation of the calculation to new clients. A User ascribe is given to suit this system.
* This dataset might be utilized for an assortment of assignments, most clear of which is pose acknowledgment by means of characterization. One may likewise endeavor client ID. On the other hand, one may perform grouping (compelled or unconstrained) to find marker disseminations either as an endeavor to foresee marker personalities or acquire measurable depictions/perceptions of the stances.
* Abstract: 5 types of hand postures from 12 users were recorded using unlabeled markers on fingers of a glove in a motion capture environment. Due to resolution and occlusion, missing values are common.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  | | --- | --- | |  |  |  |  |  |  |  | | --- | --- | --- | --- | | Data  Characteristics: | Multivariate | Number of Instances: | 78095 | | Attribute Characteristics: | Real | Number of Attributes: | 38 | | Associated Tasks: | Classification, Clustering | Missing Values? | Yes |      |  |  | | --- | --- | | Column name/  Coordinate | Percentage of missing values in a column. | | X4 | 4 | | X5 | 16.7 | | X6 | 33.1 | | X7 | 50.1 | | X8 | 60.9 | | X9 | 69.3 | | X10 | 81.1 | | X11 | 99.9 | | Y3 | 0.9 | | Y4 | 4 | | Y5 | 16.7 | | Y6 | 33.1 | | Y7 | 50.1 | | Y8 | 60.9 | | Y9 | 69.3 | | Y10 | 81.1 | | Y11 | 99.9 | | Z3 | 0.9 | | Z4 | 4 | | Z5 | 16.7 | | Z6 | 33.1 | | Z7 | 50.1 | | Z8 | 60.9 | | Z9 | 69.3 | | Z10 | 81.1 | | Z11 | 99.9 |   **Class ranges from 1 to 5 with:**  **1=Fist (with thumb out**)  2=Stop (hand flat)  3=Point1 (point with pointer finger)  4=Point2 (point with pointer and middle fingers)  5=Grab (fingers curled as if to grab)  C:\Users\sanas\AppData\Local\Microsoft\Windows\INetCache\Content.MSO\9DC40256.tmp C:\Users\sanas\AppData\Local\Microsoft\Windows\INetCache\Content.MSO\728C79E.tmp Image result for GRAB HAND GESTURES IMAGES  **Overview of final process**  Step by step walk through of solution  Procedure and EDA:   1. First, we loaded the dataset called Posture.csv. The dataset contains total 38   numerical variables with total 78096 observations.   1. Here the variables are the coordinates of the hand motion and as per the motion   the coordinates are marked. Our target variable is Class.   1. There are total 974700(32.8%) missing cells in the data. 2. The dataset is highly cardinal. There is no presence of categorical data. 3. For ex: The variable X3 contains 0.9% missing value i.e. 690 missing values 4. So, we firstly changed our data type of class variable to int data type 5. We dropped all the features containing more than 50% of missing data i.e. from X7 to Z11 6. After that we changed the remaining features data type into float data type 7. We dropped class 0 as it was mentioned in the problem statement to ignore 8. We then checked whether all the missing values in the form of (‘?’) are changed into nan values 9. After successfully changing the values into null values we then moved ahead with the single   imputation method which actually was not appropriate approach so we later on we decided  to go for multiple imputation method   1. We moved ahead with one of the multiple imputation methods which is called MICE 2. We had to change to our approach to our problem because we had some computational   errors when we used MICE method for imputation. So due to limited resources, we look  at other imputation methods.   1. Later we decided to go with fast KNN imputation which led to successful execution.   Reason for using KNN Imputation:  The KNN algorithm applied to this multi classification problem is a simple, valid and non-parameter method.  The traditional KNN has a fatal defect that the time of similarity computing is huge. The practicality will be  lost when the KNN algorithm is applied to this dataset with the high dimension and huge samples. We have  used a method called TFKNN(Tree-Fast-K-Nearest-Neighbor) is presented, which can search the exact k  nearest neighbors quickly. In the method, an SSR tree for searching K nearest neighbors is created, in  which all child nodes of each non-leaf node are ranked according to the distances between their central  points and the central point of their parent. Then the searching scope is reduced based on the tree.  Subsequently, the time of similarity computing is decreased largely.  Observations and Results:   1. Now we started applying the Random Forest Classifier model which gave us the   train score as 0.99 and test score as 0.95 and accuracy score as 0.95.  (we splinted data into train and test into 70:30)   1. We noticed that Simple imputation and fast KNN imputation gave us the same result but   KNN took almost 4 hours of execution time.   1. We also measured our model on other evaluation parameter such as precision, recall,   f1 score to see the performance of the model.  Model Evaluation  After successfully applying random forest classification technique:   * Train Score: 0.9992865766655691 * Test Score: 0.9526655000213411 * Accuracy Score 0.9526655000213411 * Cohen Kappa Score 0.9408083402947903 * Matthews Score 0.9408318735834827 * Accuracy does not only imply the accurate prediction of classes we need to see   other evaluation parameters like precision, recall, f1 score.   * By looking at the precision, recall, f1 score we can say that the model is good. * **Confusion matrix: -**   download  Classification Report: -  precision recall f1-score support  1.0 0.99 0.99 0.99 4869  2.0 0.91 0.94 0.93 4619  3.0 0.96 0.96 0.96 4826  4.0 0.94 0.92 0.93 4373  5.0 0.95 0.95 0.95 4742  accuracy 0.95 23429  macro avg 0.95 0.95 0.95 23429  weighted avg 0.95 0.95 0.95 23429  Comparison to benchmark:  We had set an accuracy scoring standard of 0.90-0.99, with the class imbalance at its minimum  After doing the predictive analysis and cross checking, we get an accuracy score of 0.95 which  fulfills the accuracy standard we had set for the model.  Apart from the general accuracy, we had also aimed to build our model on the best prediction  algorithm so that it neither underfits nor overfits. For that our training accuracy score and testing  score have to be almost equal. When we run our actual model, our training accuracy is almost at  0.99 and test score is at 0.95. The difference between two scores are negligible, and so we  consider them to be equal and acceptable.  Because it’s a classification model, the precision, recall and F1 score are necessary to gauge the  overall robustness of the model along with the accuracy of its prediction as the model has to  predict the target from one of the classes available with the least possible error.  We had set our preferred precision, recall scores in the range of 0.80 – 0.99. We note that our  average precision of our actual model is 0.95 across all the classes. All our precision score,  recall scores of the actual model are in the range of 0.90 to 0.99. This shows that our model is  highly capable of distinguishing between different classes.    Graphical Patterns and Graphs  Correlation Heatmap including all the coordinates.    Bar charts depicting the number of missing values across all classes  (POSTURES)  **1=Fist (with thumb out**)    2=Stop (hand flat)    3=Point1 (point with pointer finger)    4=Point2 (point with pointer and middle fingers)    5=Grab (fingers curled as if to grab)    7.png   1. The heatmap is built upon the library missing no which depicts the correlation between   missing numbers in a dataset.   1. From the heatmap we see that several coordinates of the markers are strongly correlated   to each other. This hints at a pattern amongst the missing values and it may affect our modeling  approach towards the dataset.   1. After building bar charts of all coordinates across all classes, we note that the percentage of   missing values is vastly different across different classes/postures. All the bar plots of each class  is different from each other. This furthers our analysis that the missing values are different  according to different postures.  Implications  These models can be fed into various hardware. Such projects and  models can be critical for human use in daily life. Augmenting these  models with artificial intelligence, various physical machines/devices can  be built and our project work can form base for those physical  machines/devices.  For Example:   1. These setups can be used for enabling very young children to interact with computer. 2. Medically monitoring patients 3. Communicating in video conferencing. Like creating an algorithm for WhatsApp so   people having disabilities of hearing can communicate effectively where each any  and every sign language get converted into subtitles.  Limitations of the data and the model:   * As we saw the dataset was huge and purely cardinal data which means there was no   categorical data but purely continuous data.   * Due to this simple eda was not possible on this dataset. * By doing some analysis we saw that there was pattern in the missing data and single   imputation was not possible because we may introduce human error or bias within the data.   * So, we plot the heatmaps to see the correlation and bar charts to see how much of   missing data is present in each feature and the nature of missing data.   * Our main principle was that when we had detected the pattern in the missing data, we   wanted our system to recognize as well so we went for multiple imputation technique   * We tried implementing MICE but it led to kernel crash. So due to limitation of computational   resources, we go for Knn Imputation.   * Running a fast KNN technique was time consuming an took nearly 5 hours to execute.   Conclusion:  Hence, we conclude that using Knn Imputation and Random Forest Classifier led to  a successful classification model with 95% accuracy and a good performance by  other Evaluation Metrics. Using multiple knn imputation technique assured us that  the machine took the pattern of missing data into consideration and accordingly  classified the data, after it was established that the missing data was missing at random.  Closing reflections:  Some of the algorithms used in our project made us realize that computational capacity is something  we need to consider before running these algorithms.  We did learn during this process and tried overcoming it, however considering time constraint we couldn’t  explore more models on this dataset.  Also, this dataset is something new to our understanding because of its highly cardinal data, we had look  Beyond traditional ways of EDA. It took thorough efforts to first understand data and then get our concepts in  line with this data.  Dealing with missing value imputation was tough, as there were columns with more than 50% missing  values. And since the data was cardinal, guesswork was impossible nor we could impute based on any  factual data.  The model can be further honed and go to prototype stage if we apply deep learning algorithms.  Considering computational limitations, we couldn’t explore more many models.  We would like to thank our mentor and institute  for this brilliant opportunity. |
|  |