**Cow Activity Prediction using ML**

By

**Arya Padmakumar**

**Objective of the Project** : Classify Cow’s activities into 9 categories based on Data collected from IMU SENSORS

**Importance of the study** :- Predicting the activity of the cow helps over a period of time helps is better livestock management

**Data Source for the project** : IMU Data (Accelerometer, Gyroscope, Magnetometer) in the form of nine csv files pertaining to each activity.

**The Process**

**Basic Processes – Importing the basic and classification packages, model parameters, loading the csv file , renaming and reading the csv file.**

1. Imported Data Manipulation and Wrangling packages Pandas and Numpy.
2. Imported the Visualization packages Matplotlib and Seaborn.
3. Imported the Classification packages like Support Vector, Random Forest, Decision Tree and Linear Regression.
4. Imported the model parameters or matrix like confusion matrix , accuracy\_score, precision\_score, recall\_score, f1\_score.
5. The nine csv files were first loaded onto google drive , the drive was mounted onto google collab notebook , each file was renamed and read.
6. All the files were concatenated into a single dataframe called df. This dataframe had 12263524 records and 11 features.
7. Since the dataframe df was very large in number , 2 smaller dataframes df1 and df2 of 20000 records each were created using random sampling. Checked if the count of records under each feature of the 2 smaller databases df1 and df2 matched with that of the original dataframe df using pie chart. Since it was closely matching, decided to work using df1 and df2.

**Analysing and Visualising the data set**

1. Checked the top and bottom 5 rows of the dataset.
2. Checked the number of rows and columns of the dataset.
3. Checked the type of data under each column of the dataset
4. Checked for null values . There were no null or missing values.
5. Checked the description ( count, minimum and maximum values, 25-50-75percentile, standard deviation of the data under each column having int64 or Float64 and transposed the same data.
6. Created a heatmap to visualize the correlation between each numeric variable.
7. Created distribution plot to visualize the skewness of distribution of each numeric variable.
8. Input variables and output variable were separated into 2 different datasets
9. Created count plot for the output variable ‘ label ‘ to visualize the imbalance in the class.
10. The percentage of each output variable class was found out.

**Outlier Identification using Box Plot and Treatment using IQR Method**

1. Creating a common function to identify outliers for each feature.
2. Plotted box plots for each feature to identify outliers.
3. Createdcommon function to replace outlier values with the median value and print boxplot thereafter to recheck the removal of outliers and  print upper and lower values of each feature .

**Data Normalisation**

1. Normalisation of input variables were done using MinMaxScaler and a new dataframe of input variables was created.

**Feature Selection**

1. Those input features which impacts the output variable (outcome) the most were selected using Ch2 test and Mutual Info Classifier. Insignificant features **gyr\_x, gyr\_y, gyr\_z , acc\_y and mag\_z** were dropped.

**Spliting of data**

1. Data Split was done to split the data first into Training and Combined Data ( Testing Data + Validation Data) in a 80/20 ratio and then the Combined Data was divided in a 50/50 ratio into Testing data and Validation Data

**Building Model**

1. For dataset df1, model was built for Support Vector Classifier (SVC) and Random Forest and for dataset df2, model was built for Decision Tree and Random Forest. The models were built and trained on test data and prediction was done for test data.

**Evaluation of the model**

1. Evaluation of all the models were done using Confusion Matrix and using accuracy\_score, recall\_score, precision\_score, f1\_score.
2. SVC, Random Forest, Decision Tree and Linear Regression Classifiers returned accuracy scores of 39%, 93%, 85% and 43% .
3. Confusion matrix was drawn using Seaborn for each model.
4. For Decision Tree, Criterion taken was ‘Entropy’ and Maximum Depth taken was 10.
5. For Random Forest , n\_estimator was 10 with Criterion as ‘Entropy’

**SVC – Grid Search – Cross Validation**

1. For SVC, best parameters and best estimators were obtained. They were {'C': 1000, 'gamma': 0.1, 'kernel': 'rbf'} and SVC(C=1000, gamma=0.1)respectively.
2. Cross Validation was done by importing GridSearchCV.
3. The grid was fitted into the training dataset.
4. Inspected the best parameters using GridSearchCV in the best\_params\_attribute and best estimator in the best\_estimator\_attribute
5. The predictions were re-run and confusion matrix was re-obtained using Y\_test and grid\_predictions.
6. Better values for model Evaluation parameters were re-obtained using Y\_test and grid\_predictions.
7. The re-estimation returned an accuracy score of 58%

**SMOTE -**

Checking if the evaluation parameter scores can be bettered if imbalance of output values is cancelled using Oversampling method.

1. Imblearn was pip installed
2. SMOTE was fitted to X\_train and Y\_train data as X\_train\_smote and Y\_train\_smote
3. Printed the count of each output class before and after the SMOTE process

**Re-evaluation of the models after the SMOTE process**

1.The Classification Models were rebuilt using X\_train\_smote,Y\_train\_smote

2. The models were re-evaluated using confusion matrix and using accuracy\_score, recall\_score, precision\_score, f1\_score.

3. SVC, Random Forest, Decision Tree and Linear Regression Classifiers returned accuracy scores of 32%, 90%, 73% and 73% .

**Under Sampling Technique**

1. Under Sampling Technique was used on a 3ed Collab Sheet to neutralize the imbalance in the out put classes by bring all the counts to be same as the lowest count.
2. This process was successful as shown by the count plot, pie chart , shape of the resultant datasets and value\_count.
3. Model Building was not done since it was taking more time due to the high number of records.

**Conclusion**

Classification models using Random Forest, Decision Tree were the best for prediction for this project followed by Linear Regression and SVC