Accident Severity Prediction

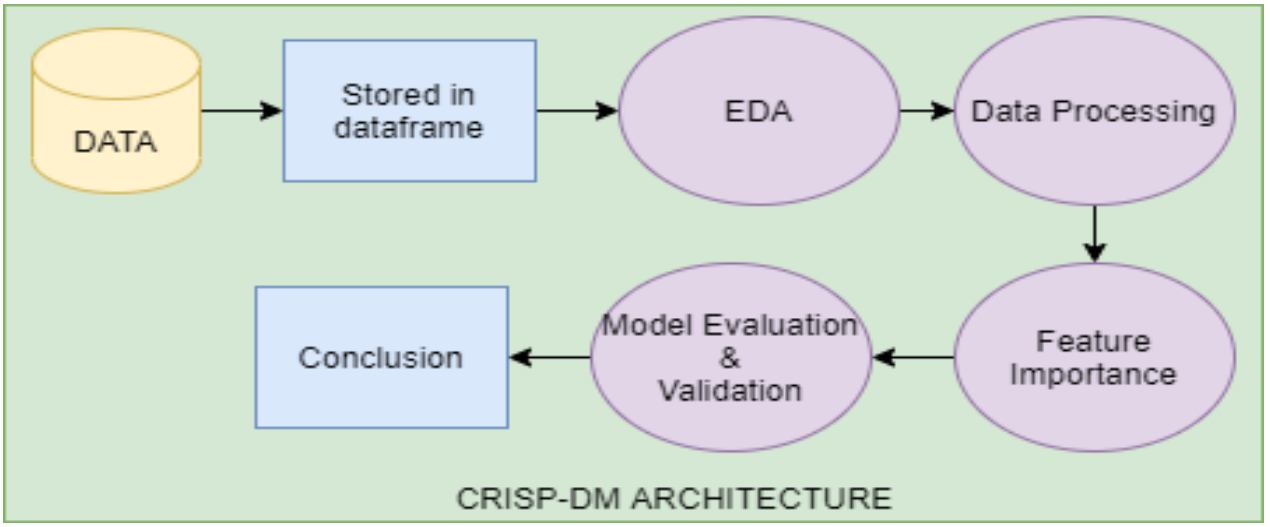
INTRODUCTION:

Neeraj at Karnataka Transportation Safety Authority is planning to set up traffic safety signs at places where a critical accident occurs and offer a safety guide to drivers in order to lessen the damage of a car accident. He wants to analyze which driving conditions (road condition, speed limit, illumination, day of week et al) are more prone to a critical accident and come up with countermeasures. Thus, he is planning to create and compare tree-based classifiers by training with the past accidents over the year and identify key variables.

OBJECTIVES:

1. Use Tree-based functions to analyze the severity of traffic accidents and identify key variables.
2. Use Split Data to separate training and validation data.
3. Run repetitive statements to find an optimal depth of Decision Tree and Evaluate Multiclass Classification to compare performance by depth and find an optimal depth.
4. Use Decision Tree (for depth optimization) and Random Forest (default) function to train a learning model for traffic accident data and evaluate the model performance using test data.
5. Use Binary Classification to validate the accuracy of respective models.
6. Use filters to retain only key variables at each tree's top node of Random Forest and sort and check key variables by assigning weight to the key variables.

HIGH LEVEL ARCHITECTURE:



SOLUTION APPROACH:

Task 1 : Use Tree-based functions to analyze the severity of traffic accidents and identify key variables.

To know the feature importances I took three approaches. In the first approach I selected top 15 features based on Recurrent Forward Selection method. In the second approach I used a tree based approach where I used a Decision Tree model to train the model and fetched the top 15 key features based on the gini impurity value(which is calculated as Entropy of parent node - Weighted Average Entropy of child nodes). In the third approach I used Random Forest to get the feature importance of top 15 key features. After that, I compared all the 3 approaches and observed that there are actually 13 features which are very much related to predict the accident severity. And the features are Urban_or_Rural_Area_1, Urban_or_Rural_Area_2, Road_Type_6, Speed_limit, Road_Type_1, Road_Type_3, Junction_Control_2, First_Road_Class, Road_Type_7, Junction_Control_0, Junction_Control_4,Road_Type_2 and Day_of_Week_4.

Task 2 : Use Split Data to separate training and validation data.

This step I did at section 7. I used the train_test_split method of model_selection module to split my data into train and test sets with the rest ratio as 0.15. I found that our dataset is imbalanced. So I took the train set data and did the upsampling using SMOTE so that our model will be trained with an equal number of classes. And after upsampling I again split the data into train and validation sets with the test ratio as 0.3.

Task 3 : Run repetitive statements to find an optimal depth of Decision Tree and Evaluate Multiclass Classification to compare performance by depth and find an optimal depth.

Here I used RandomizedSearchCV cross validation technique to get the optimal depth and found that with minimum samples split equals to 2 and maximum depth value as 5 our model gave the best accuracy. This is how we found our best hyper parameters. The input to the RandomizedSearchCV is the model, parameters that need to be tuned along with the values and scoring that we will use to know the best parameter.

Task 4 : Use Decision Tree (for depth optimization) and Random Forest (default) function to train a learning model for traffic accident data and evaluate the model performance using test data.

After doing hyper parameter tuning we trained our DecisionTree with the best parameter and also trained RandomForest and we checked the accuracy of the two models in test sets. And we found that accuracy of the tuned Decision tree is better than Random Forest.

Task 5 : Use Binary Classification to validate the accuracy of respective models.

We have checked multiple models to evaluate the best model. And we found that the accuracy of the Decision Tree is best. We also checked with the unseen test set that we split before doing upsampling that our accuracy is also very good. So we can conclude that our model is generalized.

Task 6 : Use filters to retain only key variables at each tree's top node of Random Forest and sort and check key variables by assigning weight to the key variables.

After training the Random Forest model we find all the base models(list of DecisionTreeClassifier) using estimators_ attribute. After that for each estimator we found the key variables using feature_importances_ attribute. After that we sorted those feature importances based on the gini impurity score. And we remove those variables whose gini impurity value is 0. After that we build a DataFrame where for every estimator we have the corresponding important features as well as corresponding weights being stored.

In [35]:
```
rows = []
for index,clf in enumerate(estimators):
    key_features = dict(sorted(dict(zip(X.columns,clf.feature_importances_ )).items(),key = lambda x:x[1],reverse=True))
    key_features = {k: v for (k, v) in key_features.items() if v > 0}
    rows.append(["DecisionTreeClassifier_{0}".format(int(index+1)),key_features,",".join(key_features.keys()),",".join(map(s

display_tree_features = pd.DataFrame(rows,columns=["Decision Tree Estimator","Key Features","Features","Weights"],index=list
display_tree_features
```

Out[35]:

	Key Features	Features	Weights
DecisionTreeClassifier_1	{'Urban_or_Rural_Area_1': 0.33401422472434794,...	Urban_or_Rural_Area_1,Speed_limit,Road_Type_6,...	0.33401422472434794,0.24164538384009696,0.1482...
DecisionTreeClassifier_2	{'Urban_or_Rural_Area_2': 0.7252390616277096, ...	Urban_or_Rural_Area_2,Road_Type_6,Road_Type_2,...	0.7252390616277096,0.2658065890874346 5,0.00394...
DecisionTreeClassifier_3	{'Urban_or_Rural_Area_1': 0.2972421521202719, ...	Urban_or_Rural_Area_1,Speed_limit,Road_Type_6,...	0.2972421521202719,0.26401460162070967,0.17838...
DecisionTreeClassifier_4	{'Urban_or_Rural_Area_1': 0.7175233614212883, ...	Urban_or_Rural_Area_1,Road_Type_6,Road_Type_2,...	0.7175233614212883,0.2695194326057582,0.007372...
DecisionTreeClassifier_5	{'Urban_or_Rural_Area_2': 0.6191037811355886, ...	Urban_or_Rural_Area_2,Road_Type_6,Junction_Con...	0.6191037811355886,0.2735857045543231,0.085592...
DecisionTreeClassifier_6	{'Road_Type_6': 0.3424703219683284, 'Urban_or_...	Road_Type_6,Urban_or_Rural_Area_2,Speed_limit,...	0.3424703219683284,0.28407908267280185,0.26651...
DecisionTreeClassifier_7	{'Urban_or_Rural_Area_1': 0.3010601704831565, ...	Urban_or_Rural_Area_1,Speed_limit,Road_Type_6,...	0.3010601704831565,0.2886272735556115,0.257934...