Feature Engineering

Feature engineering is the process of using domain knowledge to extract **features** from raw data via data mining techniques. These **features** can be used to improve the performance of machine learning algorithms. **Feature engineering** can be considered as applied machine learning itself.

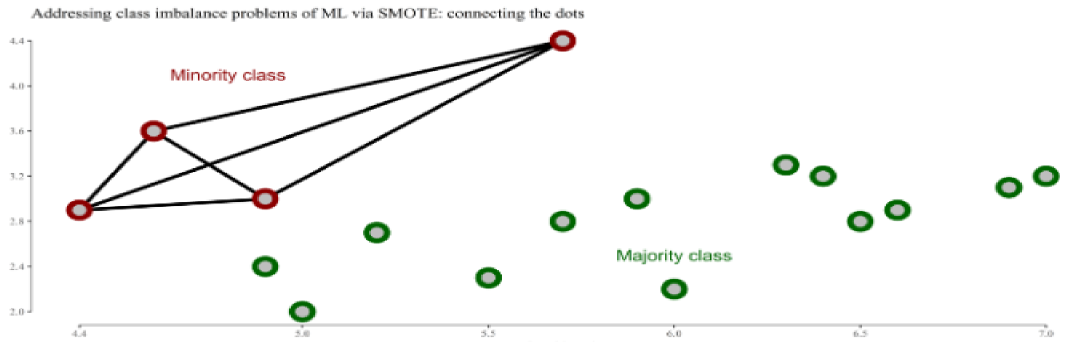
In our first Data experiment we analysed the data and identified that there are millions of records available, but after analysing we understood that there are around 33 columns of labelled data which are having duplicate set of values and there is less collinearity between the data. Currently our is to predict the accident severity by looking into the out target variable we identified that data is biased towards minority classes which is not good, so we applied SMOTE technique to do create the balanced dataset as PCA to reduce the dimensionality of the data, in future implementations we are going to apply K-Fold cross validation method to identify the best fit model to our data. The above mentioned are going to explain in detail in the following section with visualization images.

Synthetic Minority Oversampling Technique (SMOTE):

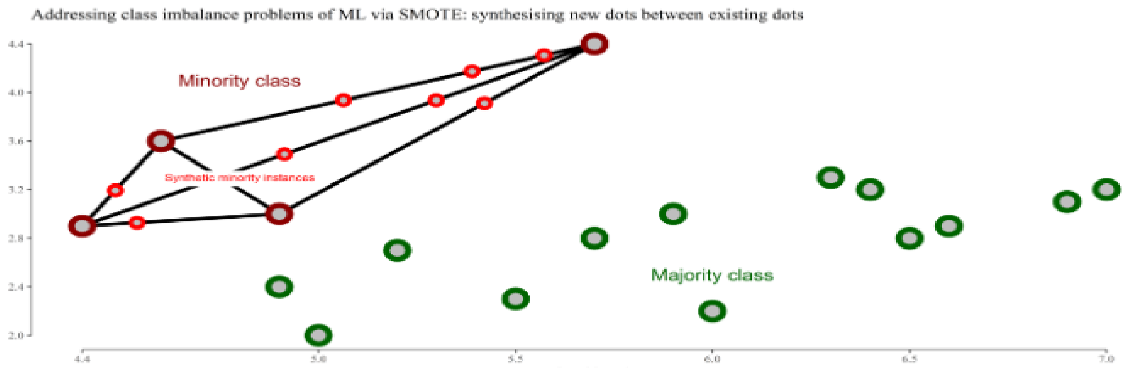
Definition: This is a statistical technique for increasing the number of cases in your dataset in a balanced way.

How it works:

SMOTE synthesises new minority instances between existing (real) minority instances. Imagine that SMOTE draws lines between existing minority instances like this.

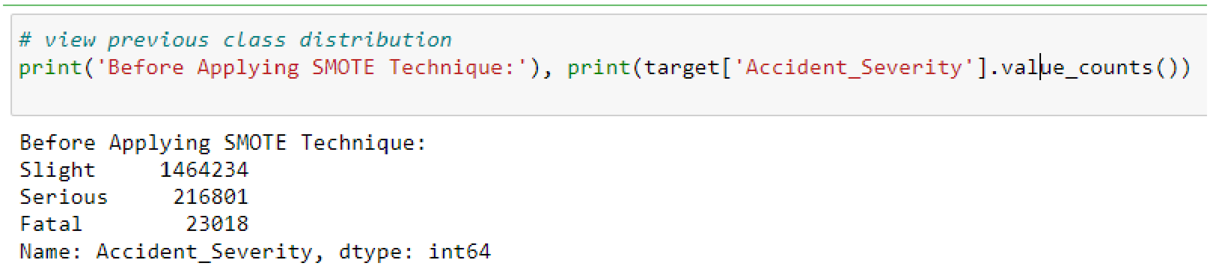


SMOTE then imagines new, synthetic minority instances somewhere on these lines.



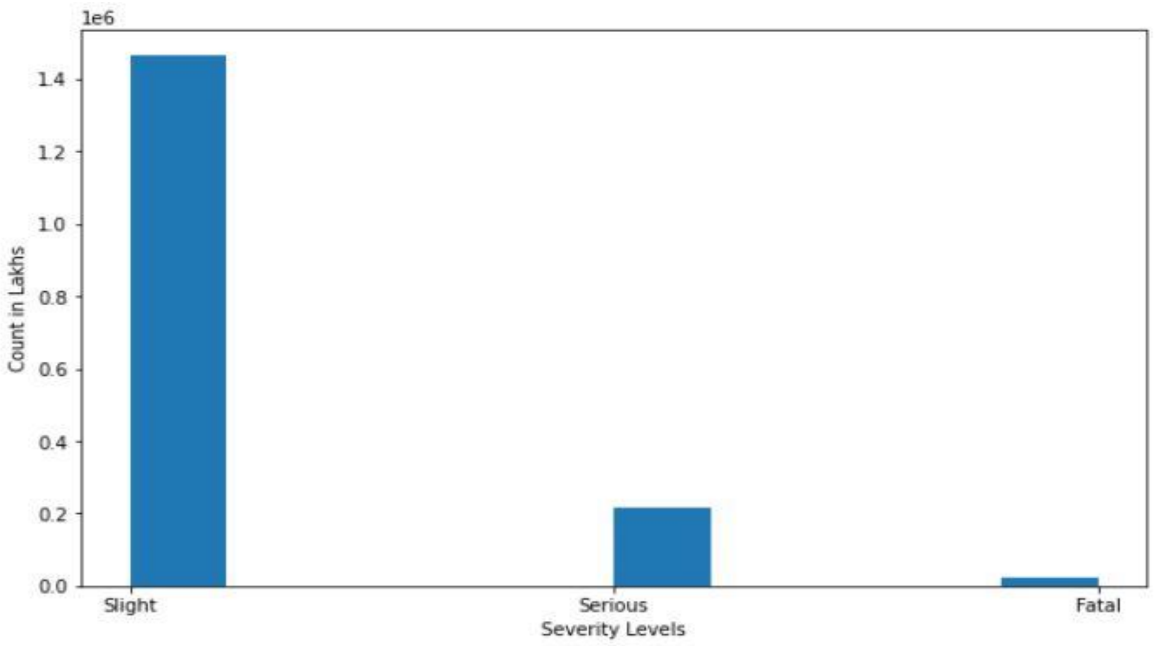
After synthesising new minority instances, the imbalance shrinks from 4 red versus 13 green to 12 red versus 13 green. Red flowers now dominate within the ranges typical for red flowers on both axes.

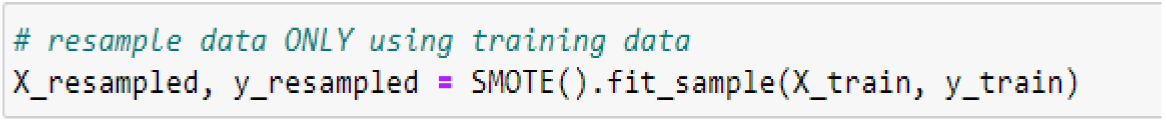
Our Experiment results on UK Accident Data:



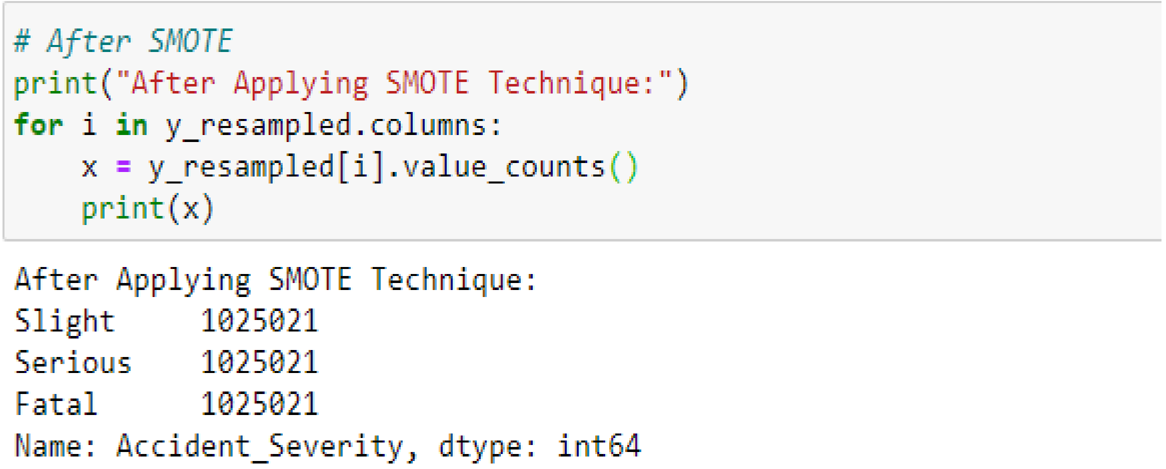
```
# view previous class distribution
print('Before Applying SMOTE Technique:'), print(target['Accident_Severity'].value_counts())
```

```
Before Applying SMOTE Technique:
Slight      1464234
Serious      216801
Fatal         23018
Name: Accident_Severity, dtype: int64
```



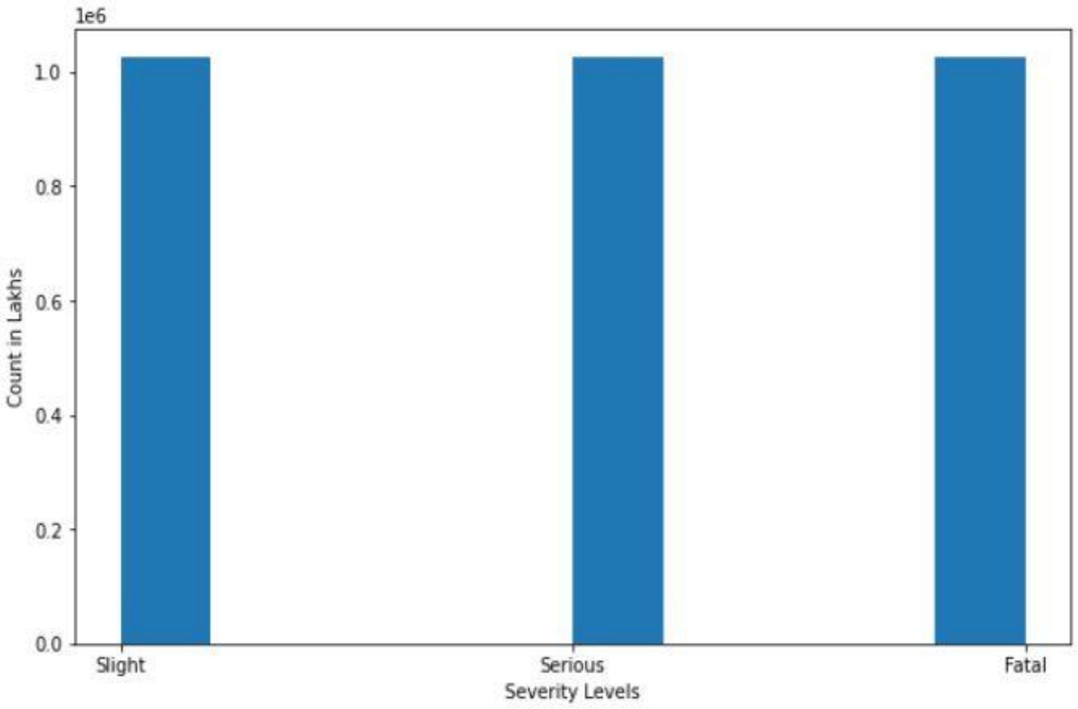


```
# resample data ONLY using training data
X_resampled, y_resampled = SMOTE().fit_sample(X_train, y_train)
```



```
# After SMOTE
print("After Applying SMOTE Technique:")
for i in y_resampled.columns:
    x = y_resampled[i].value_counts()
    print(x)
```

```
After Applying SMOTE Technique:
Slight     1025021
Serious    1025021
Fatal      1025021
Name: Accident_Severity, dtype: int64
```



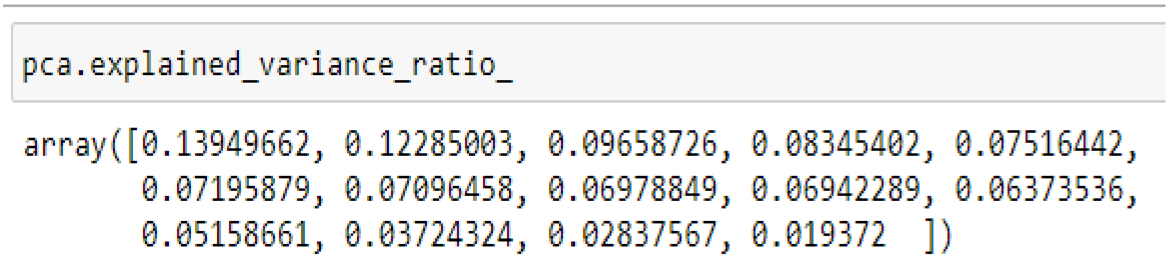
Principal Component Analysis:

In our dataset we have 33 columns and relatively large dataset of 20 million records, so we tried to reduce the dimensionality of the data by applying PCA.

Definition: PCA is a technique used to emphasize variation and bring out strong patterns in a dataset. Its often easy to explore and visualize.

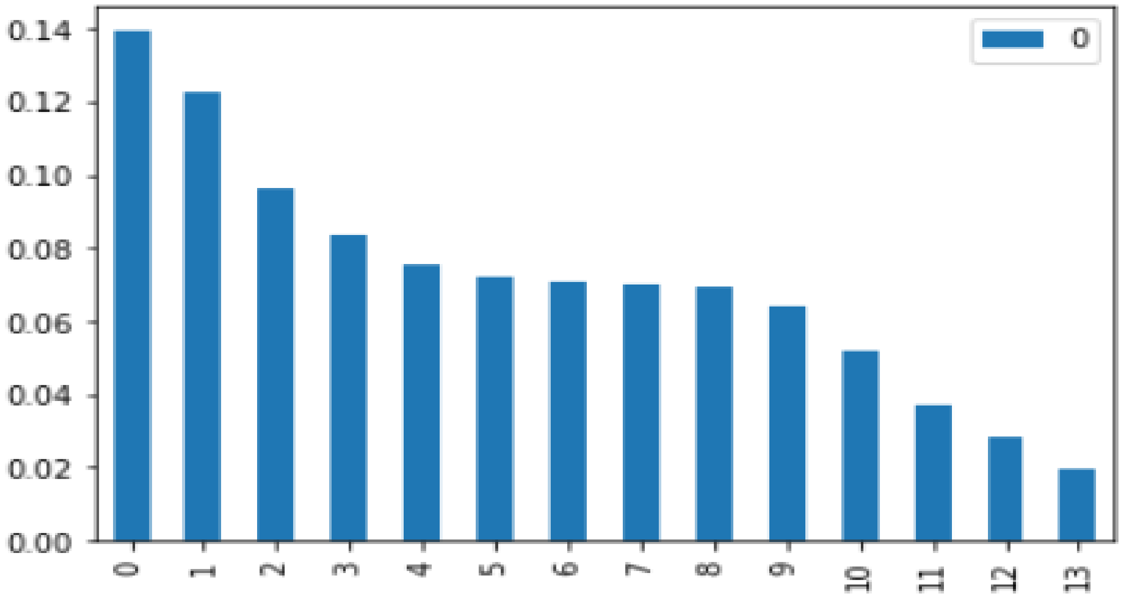
PCA creates the linear combination of original predictor variables which captures the maximum variance in the dataset and create the component one is PC1 and Second Principal component captures the remaining variance and so on.

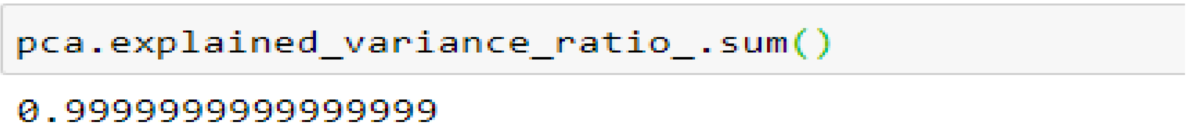
We have performed the PCA on our dataset and identified that PCA is not a perfect fit because it contains lot of categorical and duplicate variables, so using PCA reducing dimensionality did not help us much but some of the insights are draws from the PCA and following are the visualization.



```
pca.explained_variance_ratio_
```

```
array([0.13949662, 0.12285003, 0.09658726, 0.08345402, 0.07516442,
       0.07195879, 0.07096458, 0.06978849, 0.06942289, 0.06373536,
       0.05158661, 0.03724324, 0.02837567, 0.019372  ])
```





```
pca.explained_variance_ratio_.sum()
```

```
0.9999999999999999
```

Model Selection: Here we want to give further steps in identifying the best fit model to our data using K fold cross validation.

Definition: Cross-validation is a resampling procedure used to evaluate machine learning models on a limited data sample.

How it works?

1. Shuffle the dataset randomly.
2. Split the dataset into k groups
3. For each unique group:
    - Take the group as a hold out or test data set
    - Take the remaining groups as a training data set
    - Fit a model on the training set and evaluate it on the test set
4. Retain the evaluation score and discard the model
5. Summarize the skill of the model using the sample of model evaluation scores

So, in the future development will follow the above steps and identify the best fit model.