Data Services Tutorial

IMDB Movies Dataset ML

Shashank Reddy Boosi

Data Cleaning

- The first step of data cleaning is to drop columns and the columns that are dropped are `movie_id` - irrelevant for modelling, `homepage` - URL is useless to make predictions, `spoken_languages` - will be using `original_languages`, `status` - All have the same value, so redundant, `overview` & `tagview` - these are removed as the most important words are retrieved through `keywords`, `original_title` as the title doesn't have any effect.
- I then extract the year (`release_year`) and month(`release_month`) from `release_date` as they can influence the revenue and ratings of the movie. I drop `release_date`
- Our next step is to tackle the features which are in the form of json objects, the features are `cast`, `crew`, `genres`, `production_companies` and `production_countries`. As per the analysis and the number of attributes that are mainly contributing to the features, I decided to choose `5` and `3` attributes from the features and store them as list of strings. But coming to the `crew`, I only extracted the director cause he has the most weight out of all the crew and crew comprises of 100s of people who contribute to a movie. So from crew I extract `director`. I then drop the `crew` feature.

Feature Extraction and Transformation

- The first step of the feature extraction and transformation is to deal with the list of strings for the features with json objects. String transformation is applied to those feature where the names are lowercased and stripped of spaces. So that there exist a difference between `Johnny English` and `Johnny Travis` which results in unique features.

- The second step is to use a method which encodes the target with the features (`Target Encoding`), where the features are replaced with the target values and so I get a list of numericals and I take the mean of the list to achieve a numeric value which

represents that feature in a tuple. This method is applied to all features which contains string.

- The third step is to label encode the `original_languages` which converts the languages into numbers for modeling.
- Finally, I divide the features and target depending on the modeling method and apply `standard scaling` across all the features to mormalize the values and thus achieve `feature balancing`.

Modeling

Modelling is done using 2 different methods Regression and Classification. The Data Cleaning and the Feature Extraction process for both the methods is similar.

Regression (Lasso Regression)

- Many different types of regression algorithms are used and all the algorithms behaved more or less the same for this data. Finally, I used `Lasso Regression` to model because the difference between `lasso` to other algorithms is the L1 regularization that lasso does before modelling effects the performance of the model.
- The Pearson Correlation of 0.96 is achieved using `Lasso Regression`.

Classification (Support Vector Machine)

- Classification algorithms like `Support Vector Machine (SVM)`, `Linear Discriminant Analysis (LDA)` and `Random Forest Classifier` are used to experiment and model the data but I chose Support Vector Machine mainly because of handling the non-linear structure of data which is crucial because most datasets are non-linear in nature and SVM uses the concept of kernels to achieve that. In our problem we use the `rbf` kernel which is a non-linear kernel, and the main reason to use a non-linear kernel is that, it is always good to assume non-linearity in the test data.
- The Support Vector Machine with RBF kernel achieved an accuracy of `97%`, precision of `96%` and a recall of `96%`.

All the metrics are extracted to a CSV file which is generated by the code.