Name : Junyu Ren

Student ID : z5195715

COMP 9321 Assignment 3 Report

Part 1 : Regression

I use gradient boosting regression model to do part1. The correlation is stably more than 0.2, which is better than linear regression model (around 0.17) I once tried.The MSR is still huge although i tried different groups of features, ranging from 9 * 10^15 to 1.5 * 10 ^16.

I select budget, original language, runtime, release date,production company, production country, cast, crew,keywords from candidate feature pool plus 2 generated feature: number of cast and number of crew of each movie. For 'release_date', I only get the month of it because i think movies released in Dec or Jan tend to have higher revenue because of Christmas and new year holiday. For 'original_language', I label them with natural numbers. For 'production company', 'production country', 'keywords', I only get first item's name and also label them by natural numbers.For 'cast', I get the first cast of each movie and label them by natural number.Additionally, I count the number of cast of each movie into a new feature 'nb_cast'. For 'crew', I firstly found director of each movie and label them with natural numbers.Then i count the number of all crews of each movie into a new feature 'nb_crew'.As for 'budget' and 'runtime', since they are originally numerical values, I firstly want to do min-max normalization into [0,1] but the effect is not good.So, I did not process them any more and leave them as they are.

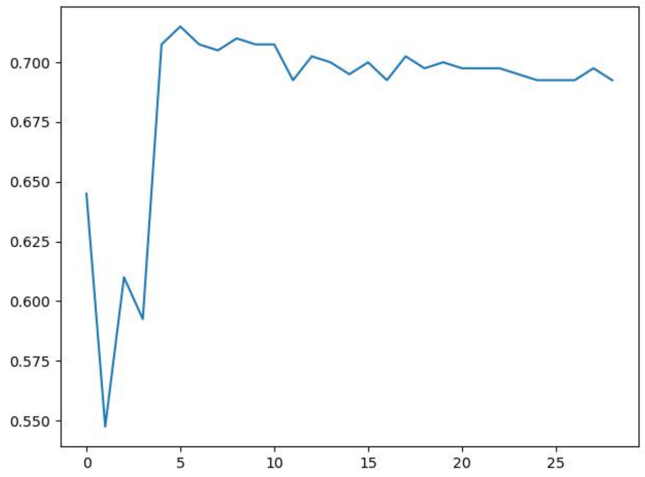
For performance improvement, more complex data processing may have better outcome. For instance, there are 758 first-actor in training set, we can calculate the frequency of each one and get the first 100 actors.We create a 100-dim vector with only 0 and 1.If this actor is in this movie, denote as 1 otherwise 0.The similiar thing we can do towards crew and keywords, etc. However, it is so computing and storage intensive and I am afraid my CPU cannot hold it.

There are also some problems I faced in processing data.The first is about JSON format data.With the lessons learnt from Assignment1, I use 'ast' library to transfer dictionary-like json format into an dictionary. Moreover, there are some missing data found in some columns. In this assignment, I give all kinds of missing data a class 'Unknown', and also a number in the map dictionary.There are other ways to deal with missing value such as replacing them as mean, median or just remove a row containing missing value.But i don't think them are suitable here.

Part 2 Classification

I use one of the most simple and understandable classification algorithm K Nearest Neighbour (KNN) to do classification. The average precision score is 0.68, average recall is 0.55 and accuracy score is 0.71.This is at least better than predicting all the rating as 3 in validation set (validation set has 277 class 3 and 123 class 2, if go with majority voting, accuracy could be 277/400 = 0.693).

Because KNN is not performed well on high-dimensional data, so, I just choose raw budget and runtime feature then get above metrics values. However, I also tried other way of processing data. I once binary classified all numerical values into high&low (for budget and runtime), many&few (for cast ,crew, number of cast and number of crew) compared them with mean and medium, denote positive class as 1 and negative class as 0. However, when I realized that this is a multi-class classification problem (there are class 1, 2 and 3 in training set), I give up binary classifying features of training set.Moreover, I tried to add extra features, number of cast and number of crew together with runtime and budget, but the evaluation metrics told me it's when this 2 features added, accuracy is almost same as baseline (0.693). So, finally, I just choose budget and runtime without any processing.Then, I want to find the best k in range of (1,30), I plot a figure of k vs accuracy and determine the best k is around 7.



For performance improvement, I consider other classification models such as random forest and get very similar outcome as KNN algorithm. Maybe I can select different features and try more models like XGBoost and SVM and do hyper-parameter adjustment to get higher accuracy score.