EC349 Individual Project

2023-12-05

Data Science Methodology

I have chosen John Rollins' General Data Science Methodology for several reasons. Firstly, it considers Analytic Approach separately. With complex datasets but limited domain knowledge, having an emphasis on examining preliminary trends within descriptive and diagnostic analytics helped me identify variables with predictive power. Secondly, it is iterative which I applied by evaluating and refining my models, leading to improvement in accuracy. Thirdly, it is holistic, focusing both on technical aspects of data analysis and problem understanding. Examining the context of the problem as well as technical aspects was useful in ensuring my models are not only robust but also align with real-world context and potential applications.

Problem Understanding

The project goal is to use data from Yelp to predict how users like different establishments in terms of the star ratings in reviews. The main objective is to identify relevant variables that are likely to influence users' star ratings and thus have the most predictive power.

Analytic Approach

I begin by examining the various datasets to identify which data can and should be used to predict 'stars'.

Descriptive Analytics

review_data_small

'stars' have a mean of 3.748 and median of 4, showing that the ratings overall seem to be skewed towards the higher end. The most popular rating is 5, followed by 4 and 1 (Figure 1). This may be explained by the fact that people tend to leave reviews when they had a particularly positive or negative experience. This skewness may also result in a bias towards positive reviews in the data.

```
#source("/Users/ekaterina/Documents/EC349 project/Code.R")
#options(repos=list(CRAN="https://cran.rstudio.com"))
#tinytex::install_tinytex(force=TRUE)

#summary(review_data_small$stars)

#ggplot(review_data_small, aes(x=stars))+geom_histogram(binwidth=1, color='black')+ggtitle("Distributio
```

'useful', 'funny', and 'cool' may be indicative of the star rating as they are direct reactions to the review. Given that both the first quartile and median for all three is 0, and that the third quartile for two of them is 0, there is a higher level of sparsity for these variables.

```
#source("/Users/ekaterina/Documents/EC349 project/Code.R")
#summary(review_data_small$useful)
#summary(review_data_small$funny)
#summary(review_data_small$cool)
```

user_data_small

User features and history may be predictive of the reviews they tend to leave.

‘review_count’ data ranges broadly, with a small number of users who are highly active. Higher ‘review_count’ may be associated with more nuanced reviews and consistent rating behaviours. The histogram shows a long-tail distribution with a small number of users having a high number of reviews.

‘average_stars’ shows most users rating positively and having extreme ratings. This data can provide context on a user’s rating behaviour.

```
#source("/Users/ekaterina/Documents/EC349 project/Code.R")

#summary(user_data_small$review_count)

#ggplot(user_data_small,aes(x=review_count))+geom_histogram(binwidth =1,color = "black") +ggtitle("Dist

#summary(user_data_small$average_stars)

#ggplot(user_data_small, aes(x=average_stars))+geom_histogram(binwidth = 0.5, color = "black") +ggtitle
```

Skewness observed above may result in bias in my model, and it may be less predictive for example for unextreme ratings like 3, or users with low amount of reviews.

Diagnostic Analytics

I identify patterns through correlation analysis to see **why** ‘stars’ may behave a certain way. Correlations below are between ‘stars’ and variable mentioned.

review_data_small

‘useful’ and others have correlation of <0.1 in absolute values, suggesting they may be poor predictors.

user_data_small

Most of the correlations with ‘stars’ are weak except for ‘average_stars’ (0.582), suggesting higher predictive power.

tip_data

Very low correlation with ‘compliment_count’ (0.003).

business_data

For numerical variables in business data there is generally weak correlation except for 'business_stars' which has correlation of 0.489 with 'stars'.

For attributes, I look at relatively simple variables. Certain attributes have >100 unique categorisations, which can increase complexity of the model and result in overfitting. The correlations coefficients suggest an overall weak to moderate relationship between the selected business attributes and 'stars'.

Data Understanding and Organisation

Incorporating variables from multiple datasets reduces bias while enhancing predictive accuracy, which is confirmed in my analysis. To balance this with the risk of fitting noise and computational constraints, I prioritised relevant, numeric, and simple categorical data from smaller datasets.

Certain data was collected through merging datasets, allowing me to explore patterns for each dataset and expand the model by incrementally adding features.

The datasets exhibited varying levels of complexity and a lot of missing data. Imputing without domain knowledge about variables, or contextual reason for why they are missing, could skew the data from true distributions, introducing biases. Thus, I focused on using a diverse set of variables and models that internally deal with NA values.

```r
#source("/Users/ekaterina/Documents/EC349 project/Code.R")
#gg_miss_var(model3_data)
#miss_var_summary(model3_data)
```

I prepared the data by renaming conflicting columns, converting Boolean variables to binary, converting between variable types, and creating dummy variables.

Modelling

Model 1 – Ordered Multinomial Logit

To minimise overfitting and understand the impact of each variable I add, starting with a limited set of features having the highest correlation coefficients with 'stars'.

I use an ordered multinomial logit due to the discrete, ordered nature of stars which may not be accounted for by linear models like ridge. A nonlinear model will also capture more complex patterns.

Across several model variations, average_stars and business_stars have significant coefficients and high t-values. logit_model11 has the lowest residual deviance (970.79) and AIC (984.79), suggesting it may be the best in terms of balancing fit and complexity. Accuracy of logit_model11 is 0.6, above the baseline accuracy (0.4653). However, the confusion matrix reveals the model struggling with lower ratings, potentially due to the skewed distributions of 'stars'. Furthermore, the wide confidence interval suggests high uncertainty. The removal of many missing observations may overinflate accuracy and not be applicable to large amounts of real-life data. Therefore, I explore further variables and models.

Exploration of variables

Next, I considered all variables, excluding those with near-zero correlations or complex relationships. Many showed weak correlation with 'stars' but may still capture useful patterns and help in predicting lower ratings.

I check for multicollinearity and to observe relevance, I perform univariate analysis to compare mean star ratings across groups for each predictor.

```
#source("/Users/ekaterina/Documents/EC349 project/Code.R")
#print(mean_stars_by_useful)
#print(mean_stars_by_funny)
```

Model 2 – XGBoost

XGBoost deals with missing values internally, and it also should improve my predictive power through boosting as it averages over different predictors by fitting a decision tree on the residuals. This is particularly useful for some features where the significance isn't obvious but with additional boosts, they may improve predictive power. Using all the variables so far, the model achieved an accuracy of 0.54 with no evident overfitting, again showing stronger performance for higher ratings through the high true positives, sensitivity, and precision. Adding other variables marginally improves accuracy, but visualising the importance reveals significance of features not previously considered.

```
#source("/Users/ekaterina/Documents/EC349 project/Code.R")
#xgb.plot.importance(importance_xgb2)
```

Since there is no overfitting and the features are more diverse, I tune the hyperparameters of the model. Increasing the depth to 9 while maintaining eta at 0.1, strikes a balance between fit and bias. To balance out the uneven distribution of ratings, I introduce subsampling to increase randomness, potentially improving predictions for lower ratings. The resulting accuracy is 0.5472 and shows no overfitting.

```
#source("/Users/ekaterina/Documents/EC349 project/Code.R")
#print(cm6)
```

Biggest Challenge

Finding the model that works was the biggest difficulty. Logit stopped working on an expanded set of features that had more NAs. With decision trees, only a small number of missing variables could be included and small perturbations to features changed the accuracy and the model a lot due to the sequential approach. Random forests would deal with the oversplitting issue of decision trees by dealing with the correlation part of variance through resampling but the 'forest' package doesn't deal with NA values. I attemped to overcome this issue by imputing variables with missForest but it proved to be computationally intensive and reaching the vector memory limit. Using XGBoost solved the problem as it deals with missing values, but initially it was difficult to find a balance between adding more variables and thus reducing bias or improving robustness, and efficiency.