Final Project Report

Adam Wilson
CSCE 4273: Big Data Analytics and Management
010652626
Fayetteville, United States
arw018@uark.edu

***Abstract*—For my final project I identified two datasets that were part of the same challenge. These datasets were provided by Airbnb on Kaggle. The challenge for these datasets was to predict which country a new user was most likely to book their first stay in. The challenge included data on users as well as data on their website sessions. I use the two datasets to help solve the classification problem presented.**

***Index Terms*—classification, data analysis, gradient boosting**

I. Introduction

The data source for this project was hosted on Kaggle. I will walk through my full process for solving this big data problem. We will look at exploratory data analysis, data pre-processing, model selection, and evaluation. Once I have selected a model, I will make predictions on the test set then submit my results to Kaggle to receive a score. I will then reflect on these scores and look back at my process to identify areas of improvement or potential exploration that may lead to a better score.

II. Exploratory Data Analysis

The Airbnb first time user booking problem comes with a few different datasets that we can capture insight and important information from. There is a training set that contains information about users, there is also a session dataset that contains user website session information. There is also a dataset containing details about the destination country that each person has chosen. My plan is to merge the session data into the user data. The goal of the data analysis is to identify changes or improvements to the datasets that can be made to potentially increase model performance downstream.

A. User Data

Let's begin exploring this data by looking at how sparse the set is. I use a package called 'missingno' to identify empty column entries. Figure 1 shows a plot of the data to help visualize this.

We can see that there are a few columns that are sparse. This includes the date_first_booking column and the age column. These sparse columns will need to be addressed.

- Sparse columns.

This gives us a good look at the set as a whole. Next lets look into the categorical and numerical attributes separately.

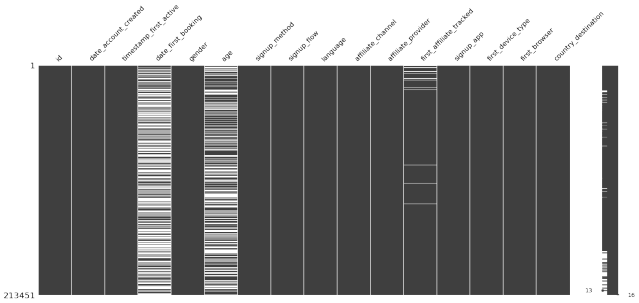

Fig. 1. The white lines show entries that are missing values.

a) Numerical Attributes: For the numerical attributes we want to identify if outliers are an issue. We also want to see what the distribution for each attribute is. Figure 2 shows the distributions of all the numerical columns, this plot shows us that there are also outliers in the age column. Another thing this figure shows us is that the distribution of the signup_flow column is concentrated heavily on one value. This may mean that we can drop this attribute all together, we will explore this further during the pre-processing section.

- Age column outliers.
- signup_flow column distribution concentration.

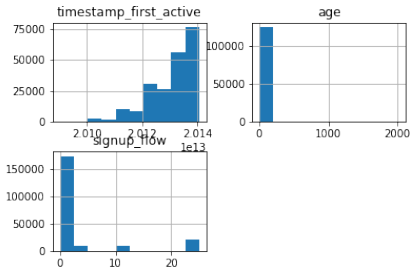

Fig. 2. Numerical attribute distribution

b) Categorical Attributes: For the categorical attributes we want to again look at the distribution and see if anything stands out. Figure 3 shows that our ages are distributed more heavily into the unknown category, next is female and then male. Nothing too out of the ordinary there. The signup_method, language, affiliate_channel, affiliate_provider, and signup_app columns favor one category very heavily, thought this is not an issue. The only column that I believe

is an issue is the first_browser column. You can see in Figure 4 that the distribution favors a few options heavily and the others become mostly noise. This must be addressed.

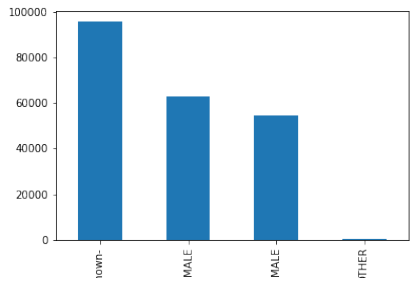

Fig. 3. Gender distribution

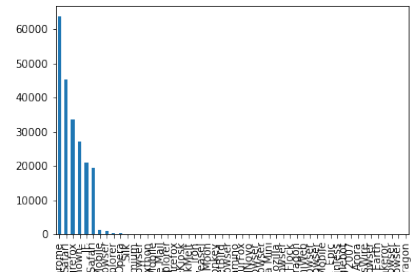

Fig. 4. first_browser distribution

- first_browser noise.

We now we have our list of items for the user data to address in pre-processing.

B. Session Data

Let's start examining our session data by again looking at sparsity. Figure 5 shows the sparsity of the session dataset. We can see that there are some missing values in the action_type and action_detail columns, and the rest is looking quite well. Let us again break this set down into its numerical and categorical attributes.

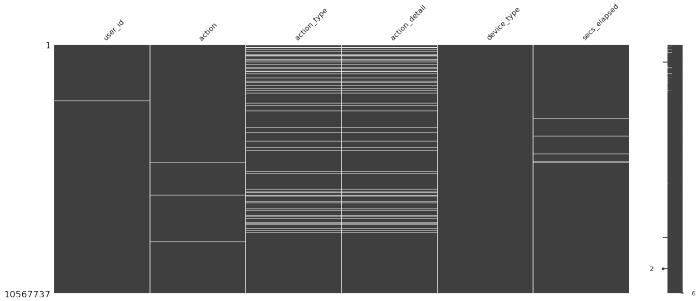

Fig. 5. The session dataset.

a) Numerical Attributes: The only numerical column to address for this dataset is the secs_elapsed attribute. The distribution for this column is tightly grouped, and outliers are not an issue as we want to capture them. There are no identifiable issues here.

b) Categorical Attributes: The categorical values tell us a little bit about what the user was doing during each of their sessions. The action column seems to be spread very thin across each of the categories. This may mean that it might be difficult for the model to identify a pattern here. However, I think this info could also be extremely valuable so we will leave it be. As show in Figure 6, the action_type column is distributed heavily into about half of the columns, this is not an issue. I believe this column is fine as is as well. There are a ton of different categories for action_detail. I believe this introduces more noise than anything else. I believe we will not lose any information if we drop this column. The device_type attribute is distributed heavily into about half of the categories just like action_type. I feel that this column can also be dropped. We are capturing device information in the user data.

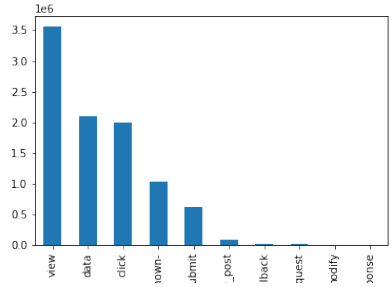

Fig. 6. action_type distribution.

This leaves us with two issues to address going forward.

- Drop action_detail.
- Drop device_type.

III. Data Pre-Processing

Now that we have explored our data and have identified the different problems we must address, lets walk through these issues and solve for each. After we have tackled the different problems, we can move on to further processing.

- Age column outliers.
- Drop action_detail.
- Drop device_type.
- signup_flow column distribution concentration.
- first_browser noise.
- Sparse columns.

A. Age Column Outliers and Dropping Columns From Session Data

These issues are easy enough to handle so I wanted to include them in the same section. Dropping of the attributes is extremely easy. We will just remove them from our data frame. The age column outliers are also an easy fix, we will

remove entries whose age is less than zero or greater than one hundred and twelve.

B. Signup Flow Column Distribution

We saw back in Figure 2 that the signup_flow field was heavily concentrated. Instead of dropping this attribute I would like to keep it in the data set. It represents the landing page the user was on when they decided to sign up. I think this may be valuable info to capture. There is nothing to change here.

C. First_Browser Noise

The first_browser categorical attribute has a lot of noise in it. The entries are distributed heavily into a handful of categories. I want to group categories outside of these few into a single category called other. This will reduce some of the noise we introduce into the model.

D. Sparse Columns in User Data

The sparse columns in the user data can be filled in using imputation. We will use KNN imputation to handle both the date_first_booking and age columns. This will cluster the values and then apply values based on the centroid of the nearest cluster. here is also some cleanup to perform such as changing unknown category names to a cleaner format, as well as formatting the date formats into a standard format.

As we can see in Figure 7, the new user data set contains no sparse columns, and we are ready to combine our two datasets and then look at constructing our pipeline.

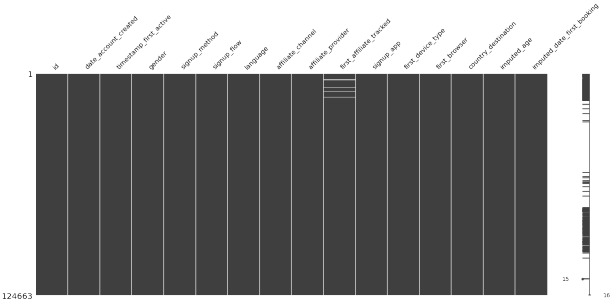

Fig. 7. Our new user dataset.

E. Combining User and Session Data

The focus now is to combine our two datasets into one input data that we can feed into our model. The idea here is to left join the session data onto the user data via the user_id field. We will aggregate the action field by counting it, and the seconds elapsed field by taking the sum. There is also a decision to be made here on whether we one-hot encode the different action types before merging. This expands the dimensionality of the input data by quite a bit. I am going to not use one-hot at this stage. Figure 8 shows the resulting training dataset. Once we have the two datasets joined, we can move onto building our pipeline for encoding and model training.

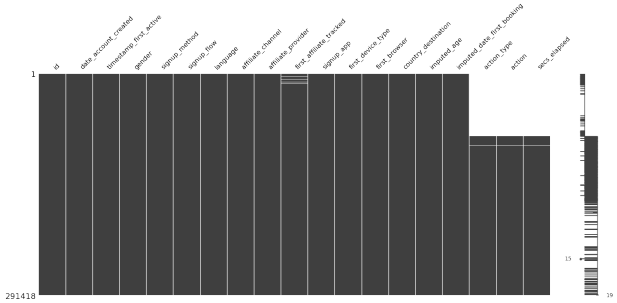

Fig. 8. Our training dataset.

F. Pipeline Construction

For our encoding we are going to implement two encoders into our sklearn pipeline. We will use a label encoder as well as one-hot encoding for our categorical attributes and use a constant imputer for our numerical values. These are created using sklearn and implemented during a pipeline run. We also want to calculate our scores for our models using the evaluation method established in the Kaggle contest rules. This score is calculated using normalized discounted cumulative gain. Once we have our scoring method established, we can add our preprocessing to our pipeline.

IV. Model Selection

For our classifier I want to target XGBoost. It is an algorithm that uses gradient boosting to focus on the places individual decision tree models went wrong. It is a method of ensembling. We will use a combination of hyperparameters and use grid search to find the optimum combination. We will use stratified k fold cross validation to compare our models for selection. The results are show in Table 1. We can see that our results are quite good! The hyperparameters in row one offer the best results on the validation set. Those are hyperparameters we will use for our prediction model.

TABLE I
Model Selection Results

Learning Rate	n Estimator	Max Depth	Mean Score
0.1	200	4	0.919159
0.1	100	5	0.918923
0.1	200	3	0.918293
0.1	100	4	0.918210
0.1	50	5	0.918165

V. Evaluation

Now that we have our model hyperparameters selected we can fit our model with the training data and then make our predictions. After running our predictions and submitting our results to Kaggle we receive a score of 0.72043. Not too bad!

VI. Reflection

I think I am happy with my score and my process given the time I had to complete this project. I learned a lot about data pre-processing and wrangling. I think that it was good

experience to learn how to merge two datasets to tell a unified story within one set that can then be used to train a model. I also enjoyed participating in a Kaggle challenge, even if it was one that is old and no longer active. I think I want to continue to participate in Kaggle competitions in my free time. I also got some experience with enesmbling and gradient boosting.

There are several ways I can improve my process or my score from this point. I think that there is more feature engineering that can be done. I would love to do things like add a field that is the number of days between when a user first made their account and when they made their first booking. Or perhaps a count of the number of sessions a user was in before they booked their first stay. I think spending more time with the data and preparing it would result in a better score.

VII. Conclusion

Overall, I really enjoyed this project. It was fun to get to select a dataset that we found interesting and solve a problem using it. I like the freedom. I would really encourage future students to find a Kaggle competition they like as a good place for them to start. The structure these competitions provide can be very beneficial if you are having trouble selecting a dataset or problem.

References

[1] https://www.kaggle.com/c/airbnb-recruiting-new-user-bookings
[2] https://www.kaggle.com/c/airbnb-recruiting-new-user-bookings/data
[3] https://scikit-learn.org/stable/index.html