What is the key features important on booking a car?

Erdi Olmezogullari

Context

- To take a quick look at data
- Feature extraction over original data (Feature Engineering)
- To explore relation between features in data
- To apply data cleansing
- To pick reasonable ML algorithm
- To pick reasonable sampling tech. to overcome skewness data

To take a quick look at data

- The data you submitted consists of 25 different features.
- The features are taking nominal, continuous values.
- Only a few of them are exposed with their names, below.
  - v1 --> unique query id
  - v2 --> unique vehicle id per query
  - v3 --> query date
  - v6 --> price of car
  - v22 --> rental duration
  - v23 --> pickup date
  - v25 --> target (booked or not booked)
- The label feature is taking two diffrent values (0 or 1). The ratios of 0s and 1s have skewness. It means that dataset is imbalanced data.
- It has 1108765 instances. 35846 instances has NA value feature.

Feature extraction over original data (Feature Engineering)

A few new feature were created over the feature v3. Year is ignored since it has taking only one value (ex: 2017)

- **v3_day_part, v3_day , v3_week , v3_dayofweek , v3_dayofyear**
- v3_day_part is taking four different values, such as early morning, morning, afternoon, and evening respectively according to hour and min value of query time.

A few new feature were created over the feature v3. Time and year were ignored since the both of them have only one value (ex: 00:00:00, 2017)

- **v23_day, v23_week, v23_dayofweek, v23_dayofyear, v23_month**

Taking difference of timestamps [pickup_date (v23) - query_date (v3)]. The difference is based on day.

- **diff_v23_v3**

Feature extraction over original data (Feature Engineering)

If diff_v23_v3 is positive and booked status (v25) is 1, it can be called as a new feature,

- **regular_booked**

If diff_v23_v3 is positive and booked status (v25) is 0, it can be called as a new feature,

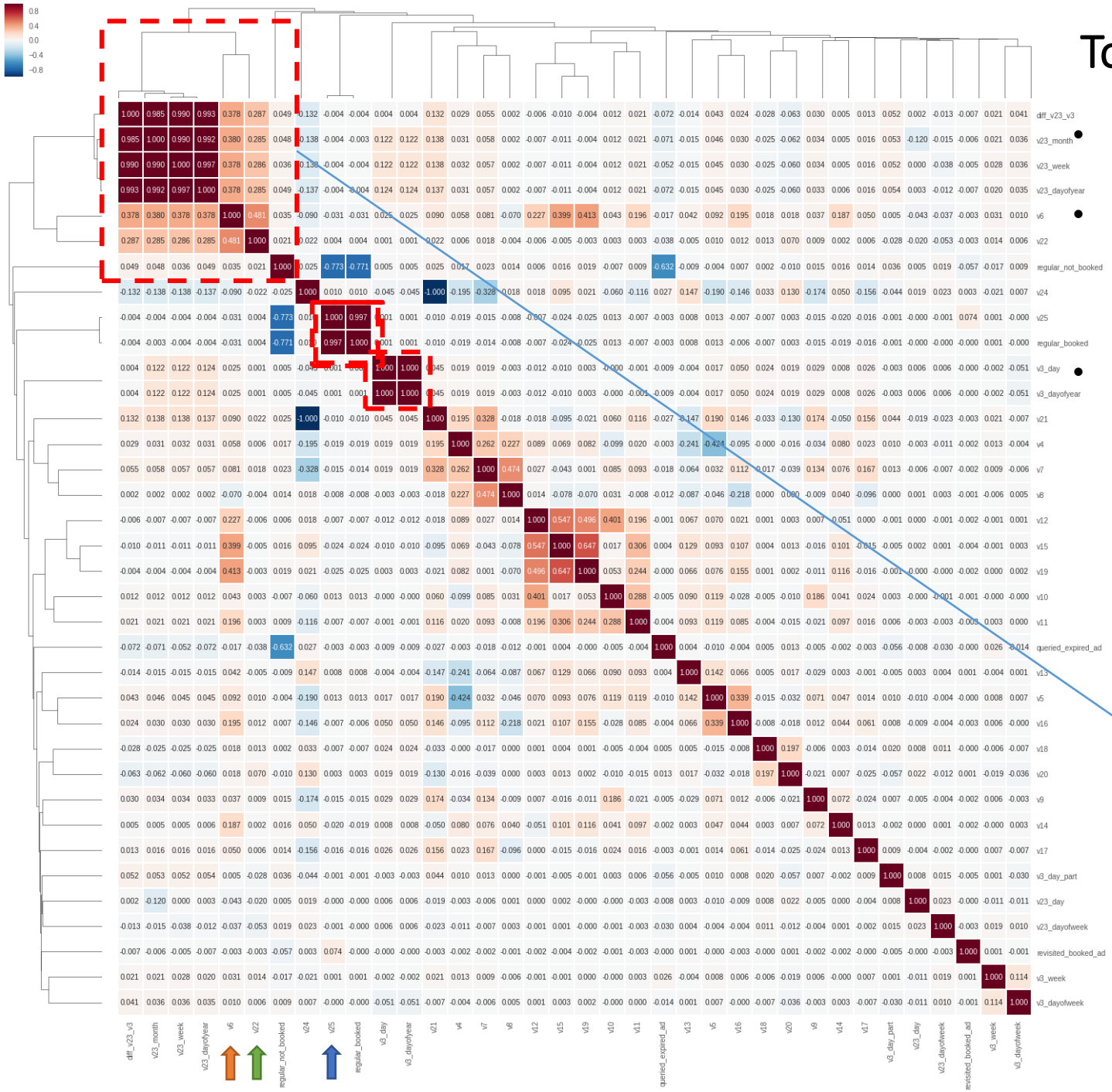
- **regular_not_booked**

There are another two different possible cases about queries according to time difference

- revisit ad after booking : **revisited_booked_ad**
- querying expired ad : **queried_expired_ad**

- Please, note that, the last possible situation is not normal case if we have a robust system, so the querying expired ad is more reasonable explanation.
- After feature extraction, we have 40 features including the original dataset.

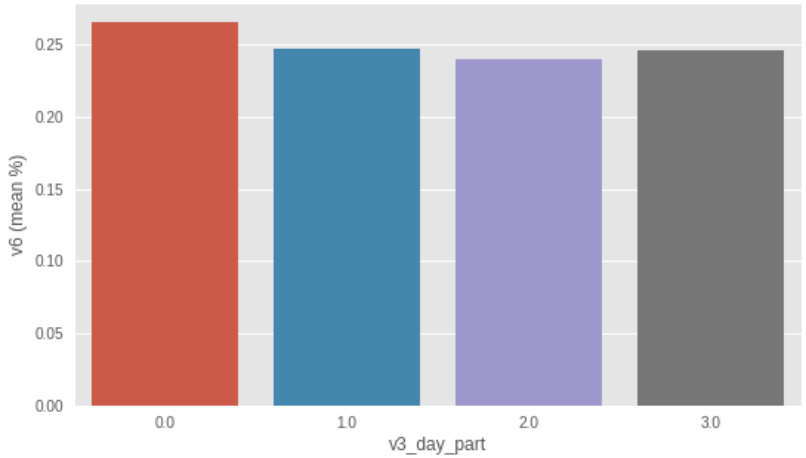
To explore relation between features in data

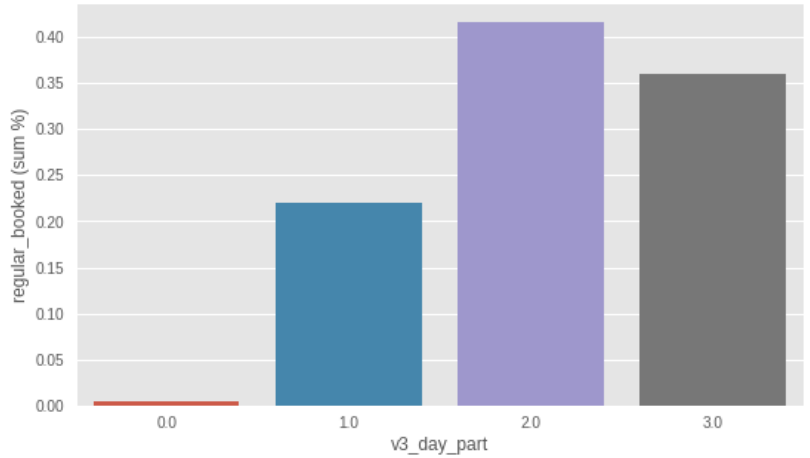


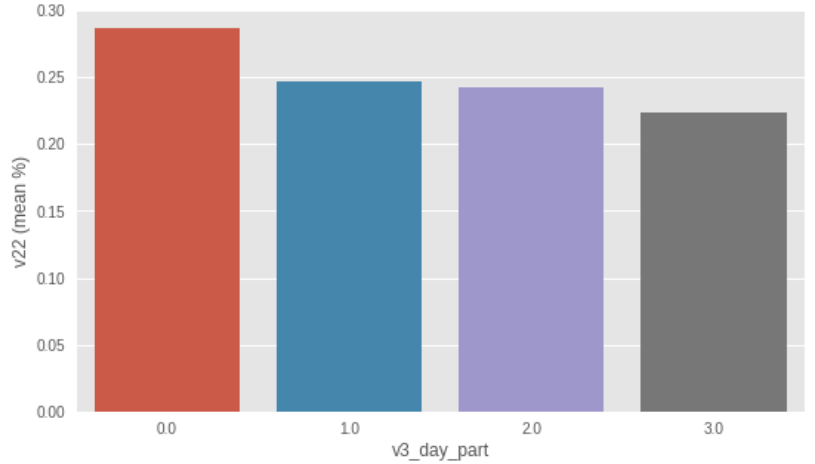
- To find linear relations amongst features, we plotted the correlation on the heatmap.
- We explored the correlations of features each other and then clustered them by correlation. The extracted features based on time have high correlation amongst themselves as you see at the red region on the top left of heatmap.
- Feature clustering by using dendrogram on heatmap is that giving idea about the order of linear relationship of features. We can get some valuable knowledge about the linear relation of features, below.

1. The feature **v6 (price of car)** is more close to **v3** (pickup date, its separated features features) on the heatmap.
2. **v23 (rental duration)** and **v6** have high correlation order in the clustering.
3. **v6** has high positive correlations between the following features according to dendrogram, respectively: v22, diff_v23_v3, v23_month, v23_week, v23_dayofweek
4. **v25** (booked status) has high positive correlation with **regular_booked**, and has high negative correlation with **regular_not_booked**. The both of them mostly will cause an overfitting problem in training step since the both of them have high linear relation between the target variable, **v6**. Before starting training step, we will get rid of these two new features

Exposing Customer Tendency in Querying and Booking a Car According to feature **v3_day_part**

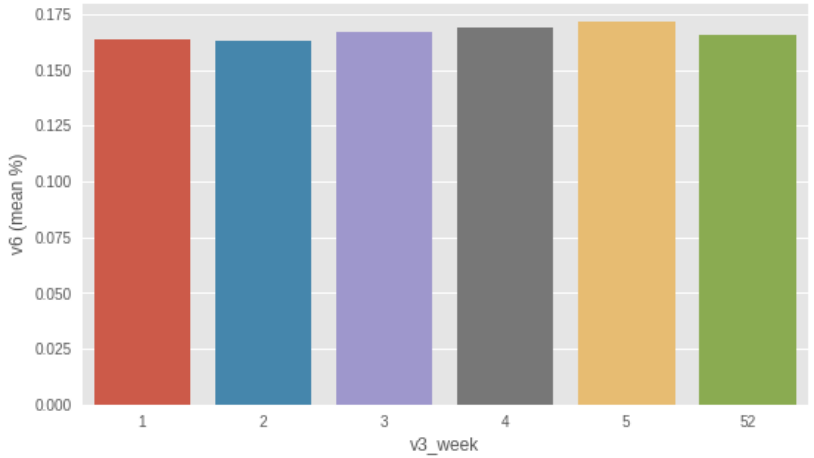


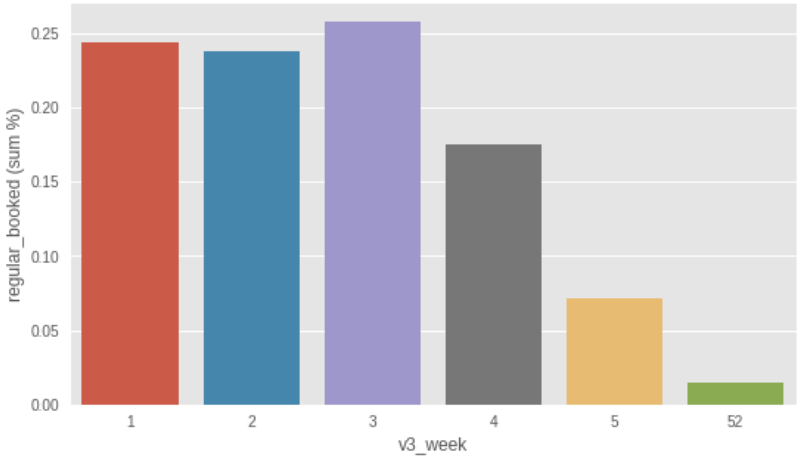


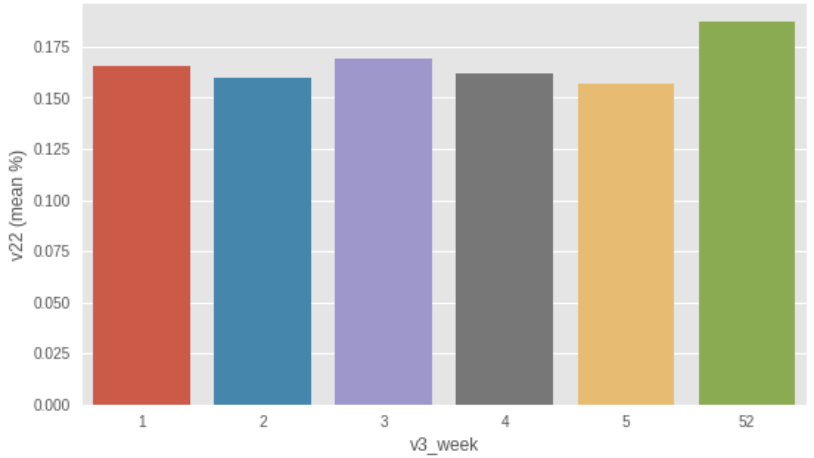


- Customers mostly are booking a car in afternoon. It is roughly 45% of all booked car as we see in regular_booked graph.
- Customers rarely are booking a car in early morning. It is less 2% of all booked car as we see in regular_booked graph.
- Customers mostly are preferring high average rental duration when they are booking in early morning as we see in v22 graph even though the number of average booking is the lowest in the early morning.
- Customers mostly are paying more average money when they are booking in early morning as we see in v6 graph because they prefers long rental duration instead of short rental when they book a car in early morning.
- They prefer mostly moderate price and rental duration when they book a car in afternoon.

Exposing Customer Tendency in Querying and Booking a Car According to feature **v3_week**

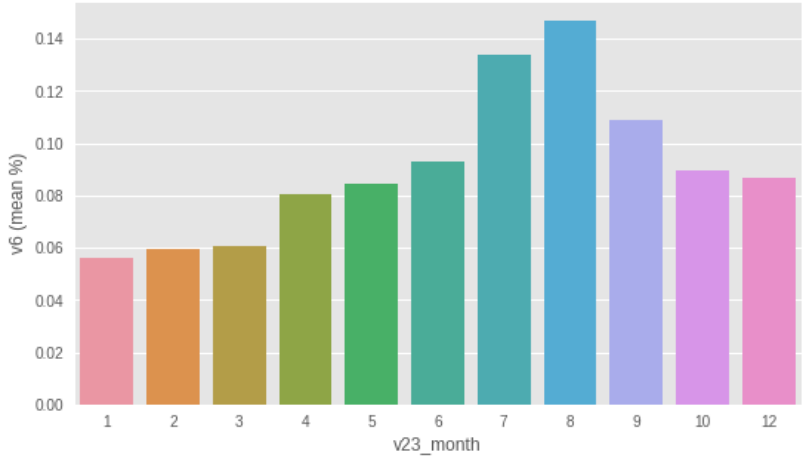


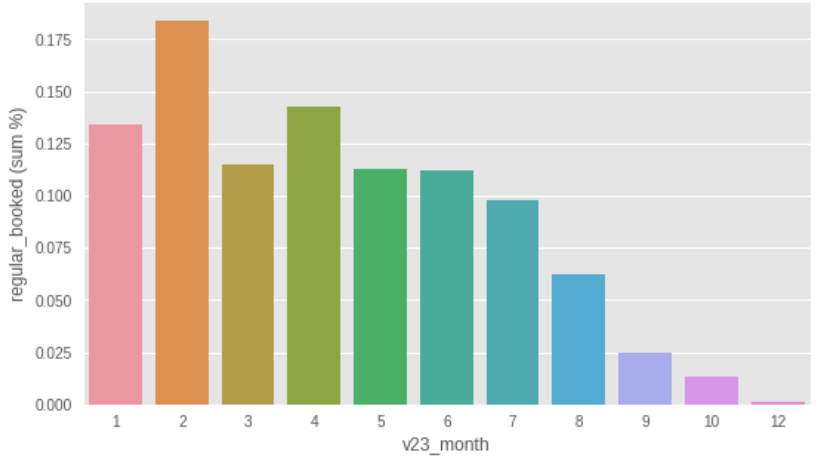


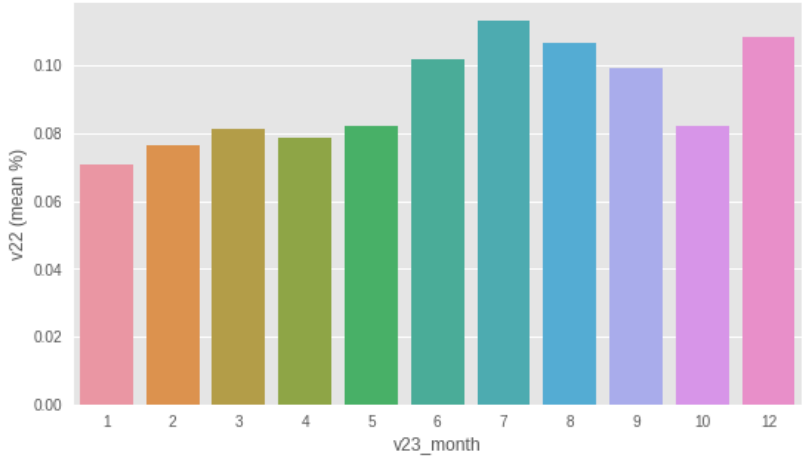


- Customers never don't book any car after week 5th up to week 52nd.
- The number of booking is going down up to week 52nd.
- The most average booking is done in week 3rd by customers.
- The highest average rental duration is being preferring in week 52th when they book a car.
- The average prices they prefer are moderate each week except for between 6-51 weeks.
- Although the most lowest booking car is being done in week 52th by customers, they also are preferring long rental duration.

Exposing Customer Booking Preferences Over Booked Car According to feature **v23_month**

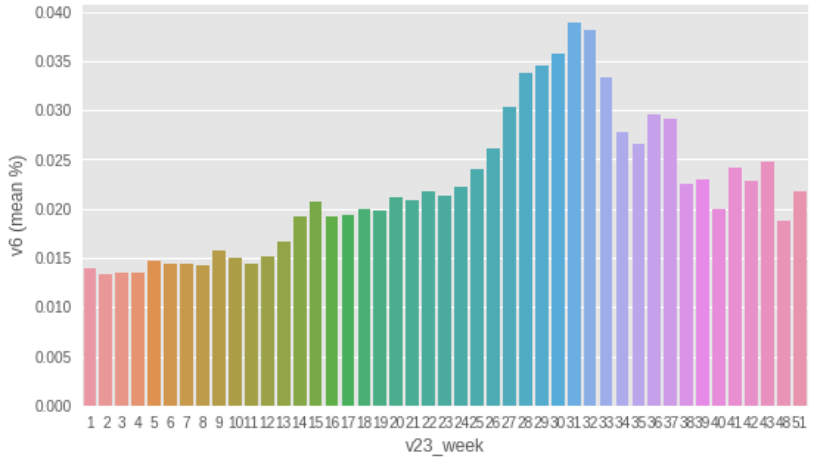


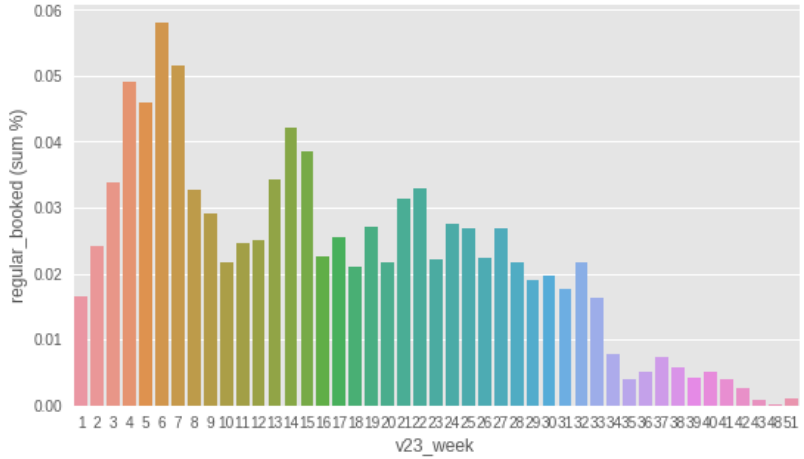


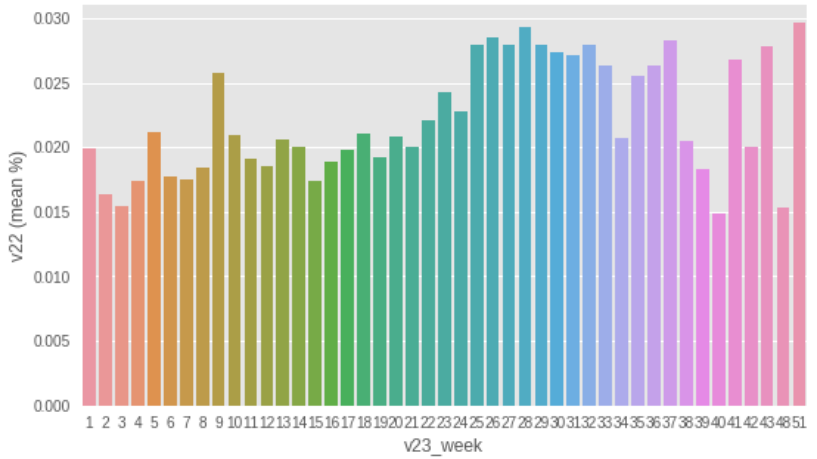


- The number of average booked car monthly is going down up to December as we see in regular_booked graph.
- However, There is no booked car in November.
- The highest number of average booked car with low average price and moderate average rental duration is being done in February by customer.
- The lowest number of average booked car with moderate average price and high average rental duration is being done in December by customer. Probably, customers may have new-year and Christmas vacation so they prefer a long rental duration.
- The average price of car and rental duration are mostly high around summer time between May and October although the number of average booked car is going down. Probably, customers may have a vacation in summer so they prefer a long rental duration to short one.

Exposing Customer Booking Preferences Over Booked Car According to feature **v23_week**

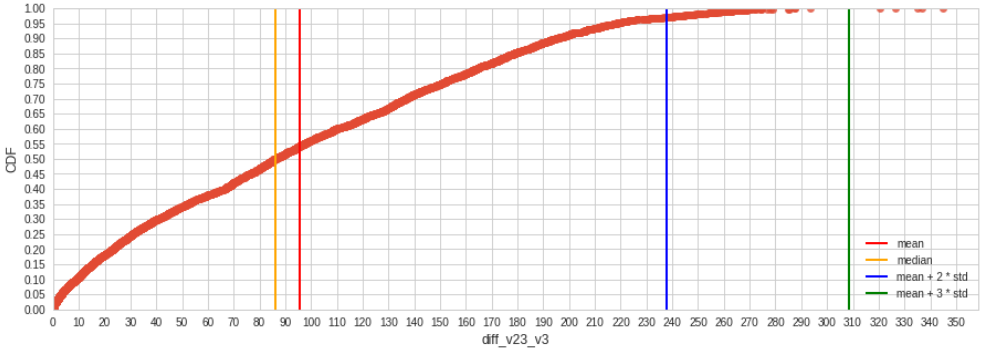




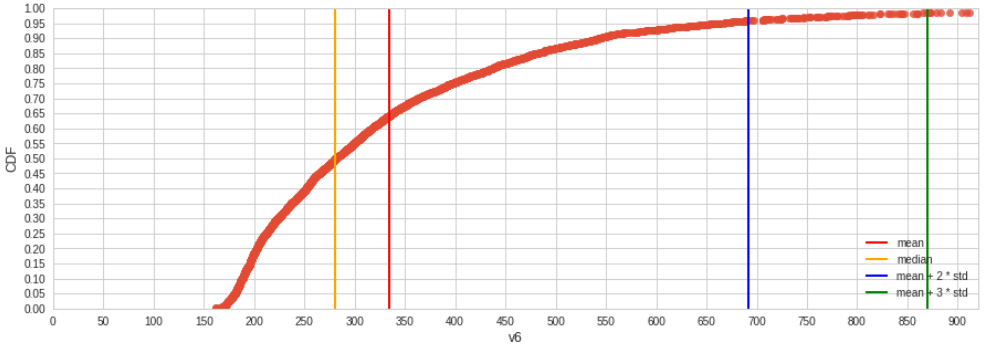


- In weekly plot, we can see more details about customers' preferences.
- However, There is no booked car in week 44th-47th, 49th-50th and week 52th.
- The highest number of average booked car with low average price and moderate average rental duration is being done in week 6th (February 6, 2017 February 12, 2017) by customer.
- The lowest number of average booked car with moderate average price and high average rental duration is being done in week 51st (December 18, 2017 - December 24, 2017) by customer. Probably, customers may have new-year and Christmas vacation so they prefer a long rental duration.
- To explain that preferences well, We need to take look at a few next week after week 51st.

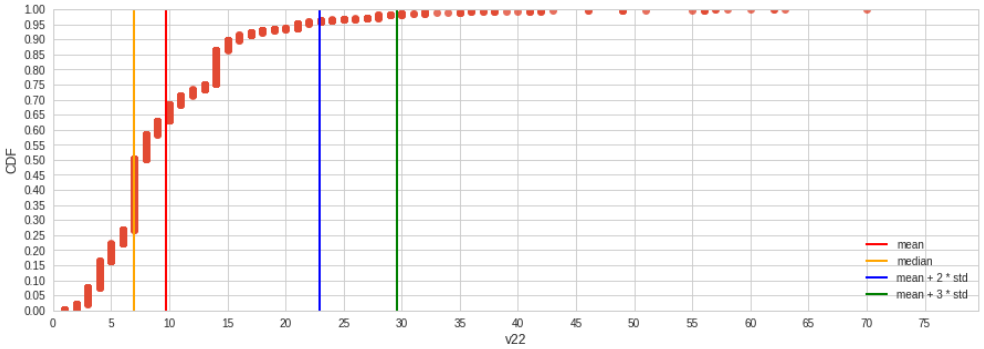
Exposing Customer Booking Preferences by Using CDF over **diff_v23_v3, v6, v22**



- Customers mostly are booking a car in prior to roughly 240 days and early. It means that 95% of the booked car were booked in prior to roughly 240 days and before. P(D <=240 days) = 0.95
- Or, 95% of the booked car were booked in prior to roughly at least 5 and lately. P(D >= 5 days) ~= 0.95
- P(D >= 5 days and D <=240 days) ~= 0.90
- Median days costumer preferred is P(D <= 85 days) ~= 0.5



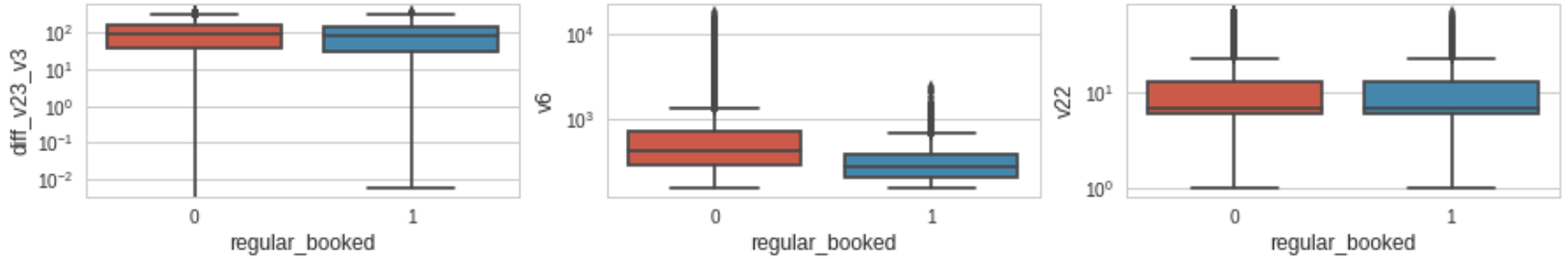
- We assumed that the unit of money is EUR.
- Customers mostly are booking a car which is cheaper than 690 EUR. It means that 95% of the price of booked car is cheaper than 960 EUR. P(v6 <= 690 EUR) ~= 0.95
- Or, 95% of the price of booked car is more expensive than 175 EUR. P(v6 >= 175 EUR) ~= 0.95
- P(v6 >= 175 EUR and v6 <= 690 EUR) ~= 0.90
- Median price of car costumer preferred is P(v6 <= 280 EUR) ~= 0.5



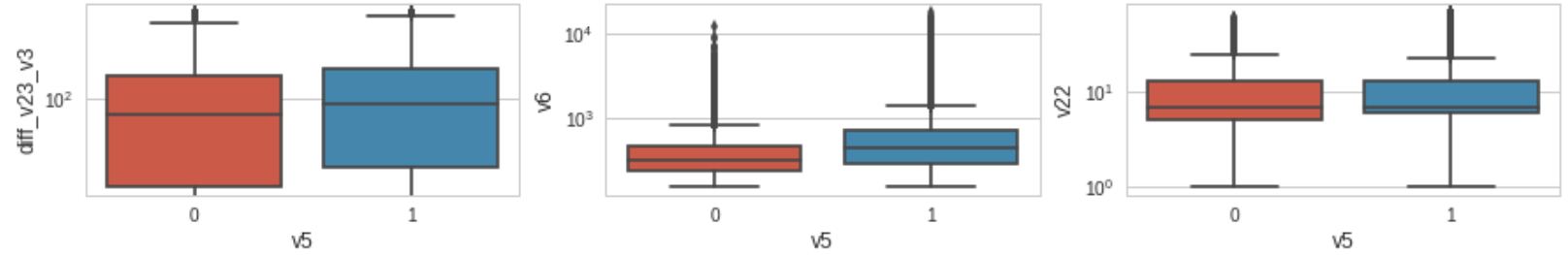
- We assumed that the unit of rental duration is day.
- Customers mostly are booking a car whose rental duration is less than 23 days. It means that 95% of the rental duration of booked car is less than 23days. P(v22 <= 23 days) ~= 0.95
- Or, 95% of the rental duration of booked car is higher than 3 days. P(v22 >= 3) ~= 0.95
- P(v22 >= 3 days and v22 <= 23 days) ~= 0.90
- Median price of car costumer preferred is P(v22 <= 6 days) ~= 0.5

Box plotting Time Difference, Price, Rental Duration (**diff_v23_v3, v6, v22**)

To expose the distribution of labeled features values (0 and 1), we plotted the boxplots of regular_booked and v25 by diff_v23_v3, v6, v22.



The shape of regular_booked by 0s and 1s is almost same. They have almost same median and ranges



The shape of regular_booked by 0s and 1s is almost same. They have almost same median and ranges

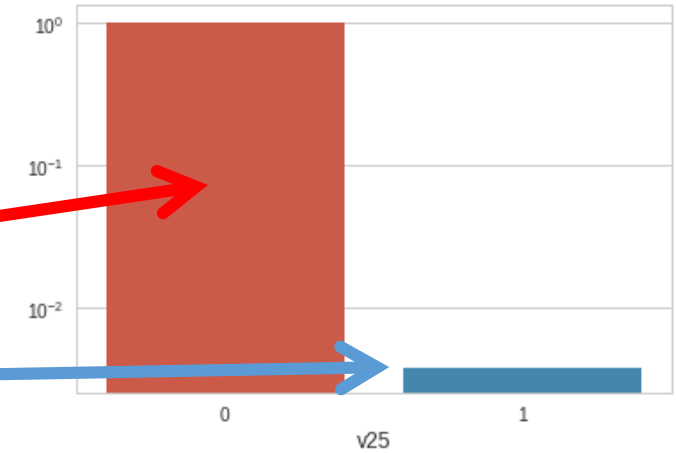
According to the both type of plots, the labeled feature is most likely generated synthetically because there is just small shifting between 0s and 1s on all plots. However, the number of instance is not balanced. So, that issue can be called as imbalanced data problem since labeled feature has skewness.

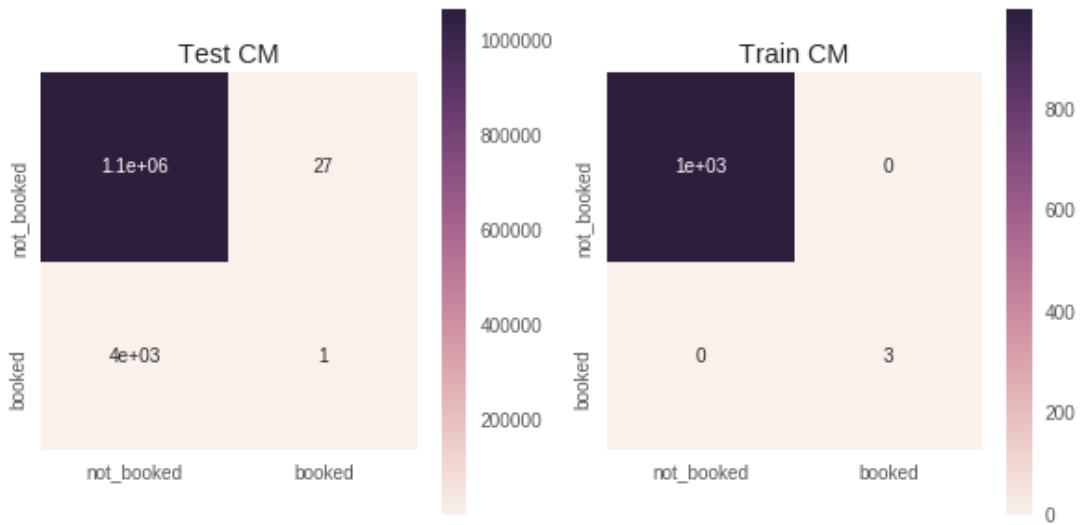
Training Phase

- We got rid of features ('regular_booked', 'regular_not_booked') which are correlated with v25.
- We implemented two different approaches by using same Random Forest classifier, below.
  1. **Approach#1** is sampling the original data to create a new small dataset which will be represent our original data. In this approach, the both of dataset's (original and generated) label ratio will be same. After got the small dataset, we applied cross_validation and performance evaluation on dataset, which consists of 1000 instances.
  2. **Approach#2** is applying just cross_validation on the whole original data.

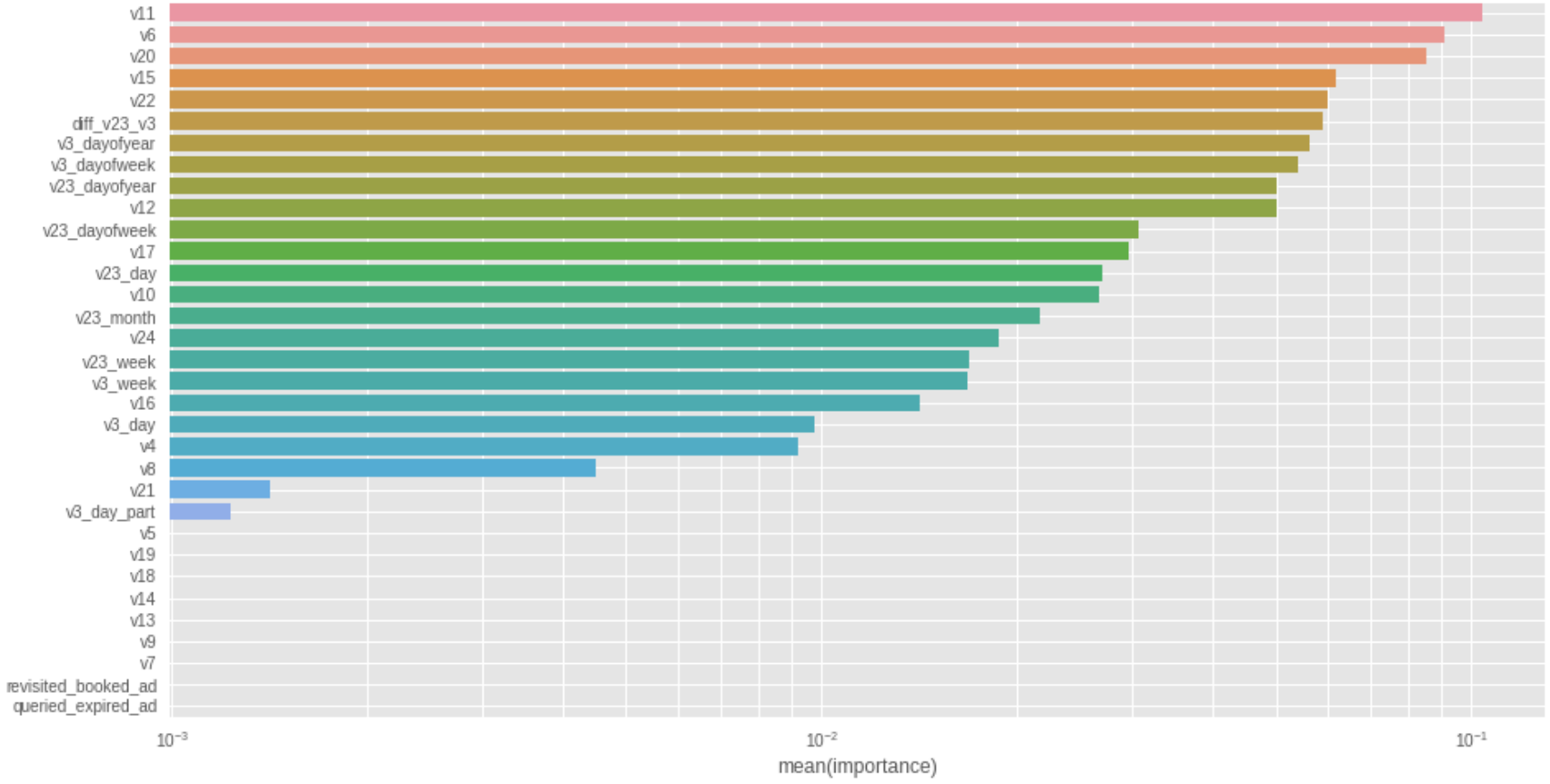
Training on Original Dataset by Using Random Forest

```
Cross Validation Scores:  [ 0.99009901  0.99009901  0.99009901  1.          1.          1.
  1.          1.          1.        ]
Cross Validation Accuracy: 1.00 (+/- 0.01)
--------------------Testing Performance--------------------
             precision    recall  f1-score   support

 not_booked       1.00      1.00      1.00   1067957
     booked       0.04      0.00      0.00      3963

avg / total       0.99      1.00      0.99   1071920

acc:  0.996278640197
--------------------Training Performance--------------------
             precision    recall  f1-score   support

 not_booked       1.00      1.00      1.00       996
     booked       1.00      1.00      1.00         3

avg / total       1.00      1.00      1.00       999

acc:  1.0
```





Training on Original Dataset by Using Random Forest



- According to our original dataset, we got the features' importance in the plot
- The most top 10 contributions are being made on booking a car by features, respectively:
  - v11, v6, v20, v15, v22, diff_v23_v3, v3_dayofyear, v3_dayofweek,v23_dayofyear, v12, v23_dayofweek
- The extracted features related to v3 and v23 are mostly making good contribution on the label data, v25.

Performance Evaluation - 1

- As we saw the result of approach1 and approach2 in the previous slide, we have overfitting problem because the label feature’s shape is imbalanced.
- The precision and recall metric ratios are pretty even bad in approach1.
- Under these circumstances, actually, we don't need to build a ML model because the most labels consist of 0s (99%). So, if we don't want to build a ML model, we can label whole new incoming instances as 0.
- Therefore, we need to use some special sampling techniques to overcome skewness data.

Overcoming Imbalanced Data (Skewness) Problem

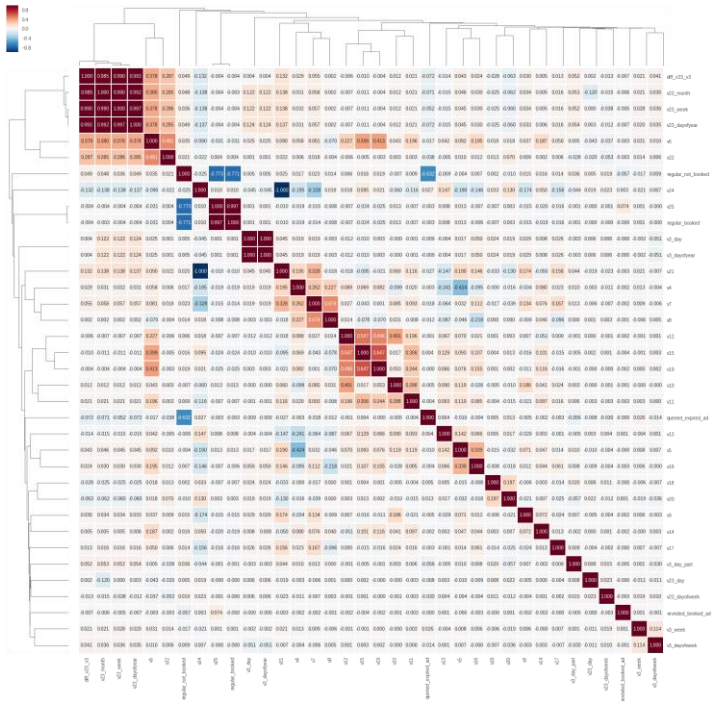
There are a few sampling techniques to overcome that problem, below

- Oversampling
- Undersampling
- SMOTE
- ADASYN

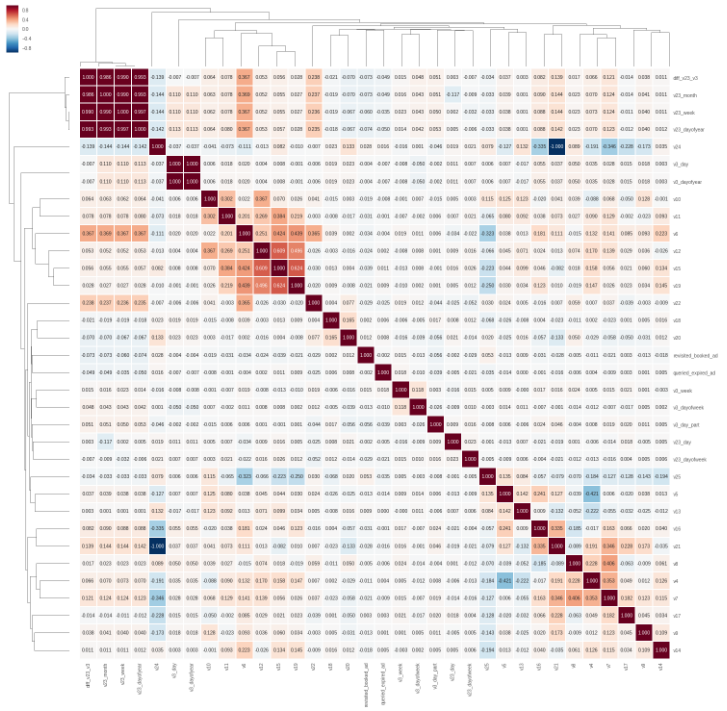
We used oversampling and SMOTE. However, we decided to use oversampling technique at the end of day since it is faster than SMOTE (based on SVM) in terms of performance hardware and time cost.

- We increased the ration of minority up to 0.7% instead of creating equal ratio classes.
- After solving imbalanced data issue, we applied same approaches again.

Original vs Oversampled Dataset



Correlation and dendrogram of original dataset

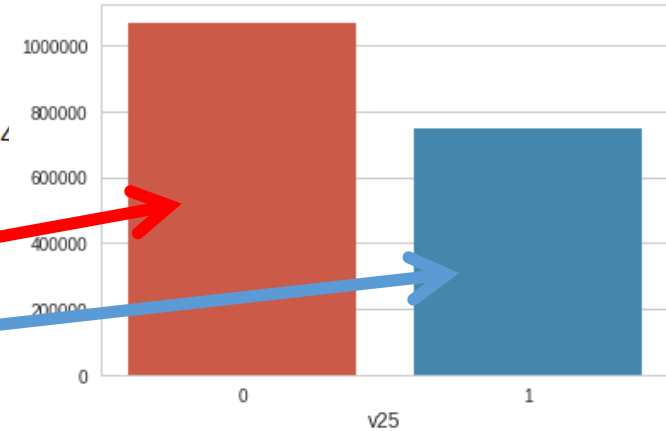


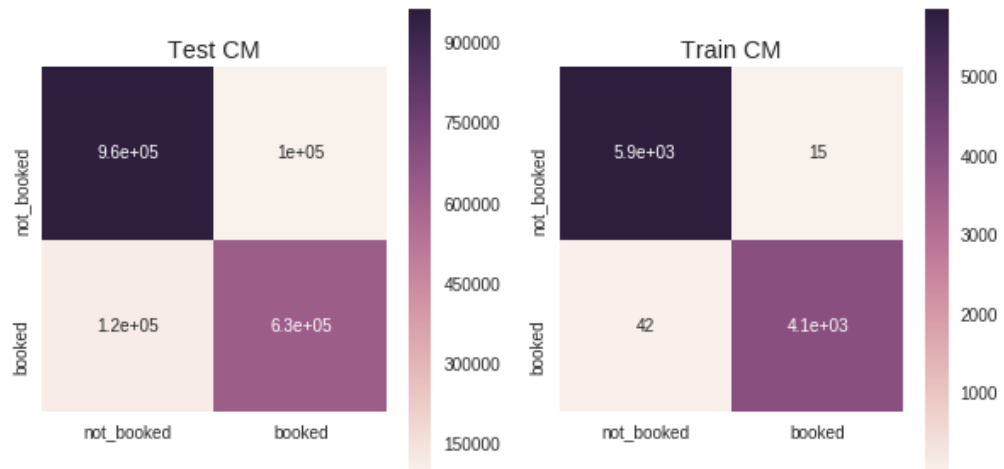
Correlation and dendrogram of oversampled dataset

- If we take a look at their correlation matrix,
- We can see the difference well. Some feature obviously has more high correlation each other after oversampling.
- So, clustering of feature also changed since the initial correlation values were changed

Training on Small Dataset by Using Random Forest – Oversampling

```
Cross Validation Scores:  [ 0.88011988  0.88411588  0.856       0.875       0.857       0.864
  0.89489489  0.88188188  0.88288288]
Cross Validation Accuracy: 0.88 (+/- 0.02)
--------------------Testing Performance--------------------
             precision    recall  f1-score   support

 not_booked       0.89      0.91      0.90   1063071
     booked       0.86      0.84      0.85    744150

avg / total       0.88      0.88      0.88   1807221

acc:  0.88050160993
--------------------Training Performance--------------------
             precision    recall  f1-score   support

 not_booked       0.99      1.00      1.00      5882
     booked       1.00      0.99      0.99      4117

avg / total       0.99      0.99      0.99      9999

acc:  0.994299429943
```

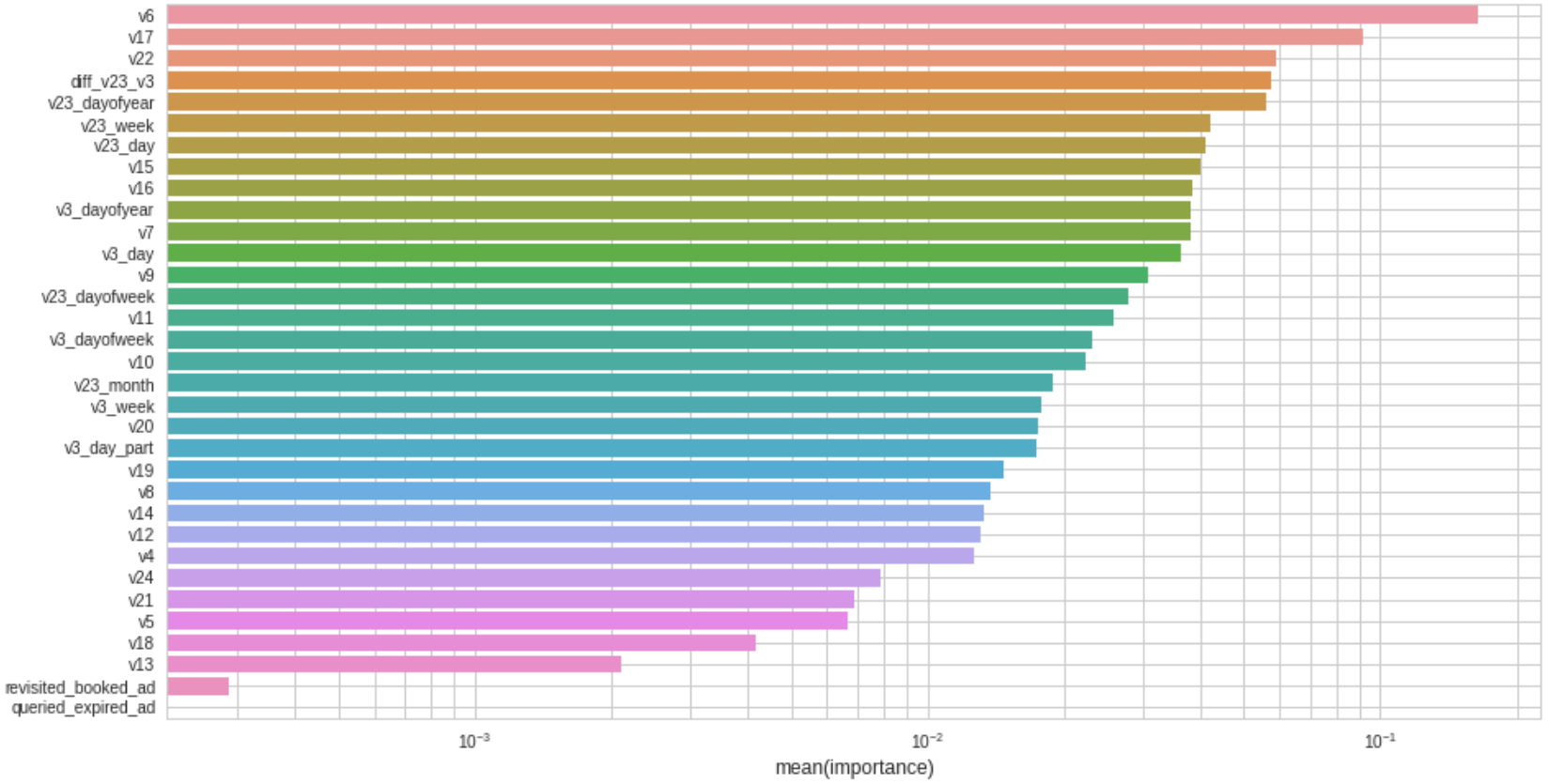




Performance Evaluation - 2

- As we saw the result of approach1 in the previous slide, we coped with overfitting problem.
- Approach1 is giving a reasonable result on small sample dataset which represents whole original data.
  - The performance of approach#1 is better than performance of approch#2.
  - The precision and recall metric ratios are pretty good even though we tested the rest of oversampled dataset (1807221 instances) on our model that we built on the small dataset (10000 instances) in approach1.
- According to cross_validation results of apporch#2 in the previous slide, it will not reasonable to apply that approach on whole data in 10 Kfold cross_validation. Because, it is still overfitting although we performed on whole oversampled data.

Training on Small Dataset by Using Random Forest – Oversampling



- According to our small dataset generated over oversampled data, we got the features' importance in the plot
- The most top 10 contributions are being made on booking a car by features, respectively:
  - v6, v17, diff_v23_v3, v23_dayofyear, v23_day, v15, v16, v23_dayofyear, v11, v12, v3_dayofweek
- The extracted features related to v3 and v23 are mostly making good contribution on the label data, v25.
- v6, price is most important role on building decision tree after applying oversampling technique.
- revisited_book_ad and queried_expired_ad have the lowest contribution level, respective since the both of them related to after booking action. It means that they don't not have effect on booking.

Conclusions

- Sometimes oversampling is causing an overfitting problem. To overcome that problem,
  - We need to retrieve more dataset.
  - SMOTE and ADASYN or another creative sampling technique can be used.

- According to feature importance of Random Forest classifier, we can make a comment about features' contribution on labeled features.
  - v6 (price of car) has most importance. it means that it is playing important role to build a tree. So, we can think that v6 has the highest influence on booking a car.

Conclusions - Business Cases

- Since time based feature has more influence on booking, we may make some seasonal campaign/discounts distribute coupons (discounted oil in X station) to increase number of booking and engagement.
  - Especially, after 5th week of year there is no booking up to 51st week (roughly at the end of year). We need to focus on that period.
- A simple rule based recommendation engine can be developed with respect to querying time (v3_day_part) to recommend relevant results to costumers.
  - For example, in early morning, costumers prefer a long rental duration. That information may be categorized according to the pickup month, as well. We can recommend costumer a simple booking packages.
- We may segment costumer into different categories (e.g. flex, moderate, urgent) according to their queries on top of feature diff_v23_v3. It may be some subscription type.

Future work

- The most analysis were done with respect to booked cars. However, we need to find the root cause why customer is not booking a car even after querying.
- We can explode the other continuous variables by using CDF to define manually discretization points. It may affect the contribution and accuracy.
- Feature selection algorithm (forward, backward) and statistical approaching (Chi2, ANOVA) can be applied to select suitable feature and reduce size of data to overcome computing cost.
- To make a forecasting costumer preferences and costumer tendency, Time series analysis can be used to decompose data into trend, seasonal patterns over querying time (v3) and pickup date (v23).
- We can investigate why the customers are querying expired deal.
- To boost the result, maybe we can use another supervised learning classifier, such as Logistic Regression with multiple input for binary classification.