**Capstone 1**

**Bank Marketing Dataset**



Submitted By, Submitted On,

Sankeerth Ankam October 10, 2018

**CONTENTS**

**1. Abstract**

**1.1 Problem Statement  
 1.2 Client  
 1.3 Dataset**

**2. Data Inspection**

**3. Data Pre-Processing**

**4. Exploratory Data Analysis**

**4.1 EDA with numeric variables  
 4.2 EDA with categorical variables   
 4.3 Box plot of all the numeric variables  
 4.4 Effect of classes of response variable with respect to other variables**

**5. Feature Engineering**

**5.1 Consolidate category classes  
 5.2 Binning ‘age’   
 5.3 Categorize ‘day’   
 5.4 Merging ‘marital’ and ‘age’  
 5.5 Inclusion and Exclusion of ‘duration’  
 5.6 Treating outliers**

**6. Ready for Machine Learning**

**6.1 Standardization and Normalization  
 6.2 Upsampling and Downsampling  
 6.3 Dummy Variables**

**7. Machine Learning**

**7.1 Logistic Regression  
 7.2 K-Nearest Neighbors  
 7.3 Support Vector Machines  
 7.4 Random Forest**

**8. Choosing the best model**

**1. Abstract**

**1.1 Problem Statement:**

To elevate the enrollment rate of a campaign (term deposit), understanding the clients and their behavior (from their data) plays a significant role. The goal of this project is – Given a client’s attributes*, predict whether or not they end up subscribing for a term deposit*.

**1.2 Client:**

The data is related with direct marketing campaigns (phone calls), of a Portuguese banking institution (name of the firm has been anonymized, for confidentiality reasons).

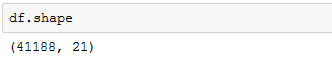
**1.3 Dataset:**

This dataset is collected from [University of California, Irvine – Machine Learning Repository](mailto:https://archive.ics.uci.edu/ml/datasets/bank+marketing).

|  |  |
| --- | --- |
| **Bank Client Data** |  |
| 1 | age (numeric) |
| 2 | job : type of job (categorical: 'admin.','blue collar','entrepreneur','housemaid','management','retired','self employed','services','student','technician','unemployed','unknown') |
| 3 | marital : marital status (categorical: 'divorced','married','single','unknown'; note: 'divorced' means divorced or widowed) |
| 4 | education (categorical: 'basic.4y','basic.6y','basic.9y','high.school','illiterate','professional.course','university.degree','unknown') |
| 5 | default: has credit in default? (categorical: 'no','yes','unknown') |
| 6 | housing: has housing loan? (categorical: 'no','yes','unknown') |
| 7 | loan: has personal loan? (categorical: 'no','yes','unknown') |
| **Related with the last contact of the current campaign:** |  |
| 8 | contact: contact communication type (categorical: 'cellular','telephone') |
| 9 | month: last contact month of year (categorical: 'jan', 'feb', 'mar', ..., 'nov', 'dec') |
| 10 | day\_of\_week: last contact day of the week (categorical: 'mon','tue','wed','thu','fri') |
| 11 | duration: last contact duration, in seconds (numeric). Important note: this attribute highly affects the output target (e.g., if duration=0 then y='no'). Yet, the duration is not known before a call is performed. Also, after the end of the call y is obviously known. Thus, this input should only be included for benchmark purposes and should be discarded if the intention is to have a realistic predictive model. |
| **Other attributes** |  |
| 12 | campaign: number of contacts performed during this campaign and for this client (numeric, includes last contact) |
| 13 | pdays: number of days that passed by after the client was last contacted from a previous campaign (numeric; 999 means client was not previously contacted) |
| 14 | previous: number of contacts performed before this campaign and for this client (numeric) |
| 15 | poutcome: outcome of the previous marketing campaign (categorical: 'failure','nonexistent','success') |
| **Social and Economic context attributes** |  |
| 16 | emp.var.rate: employment variation rate |
| 17 | cons.price.idx: consumer price index |
| 18 | cons.conf.idx: consumer confidence index |
| 19 | euribor3m: euribor 3 month rate |
| 20 | nr.employed: number of employees |
| **Output variable (desired target):** |  |
| 21 | y has the client subscribed a term deposit? (binary: 'yes','no') |

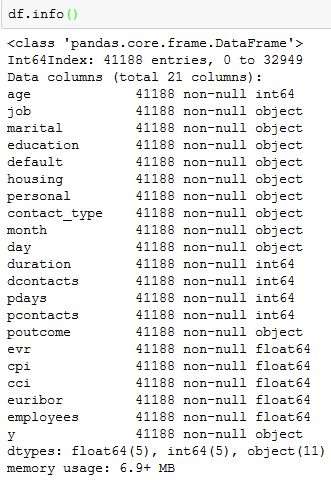
**2. Data Inspection**

**df.shape**



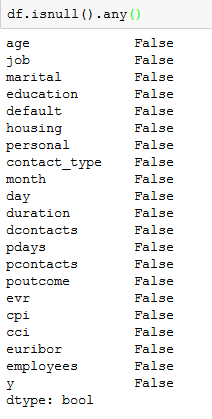
**Interpretation:** There are 41188 rows and 21 features.

**df.info()**

****

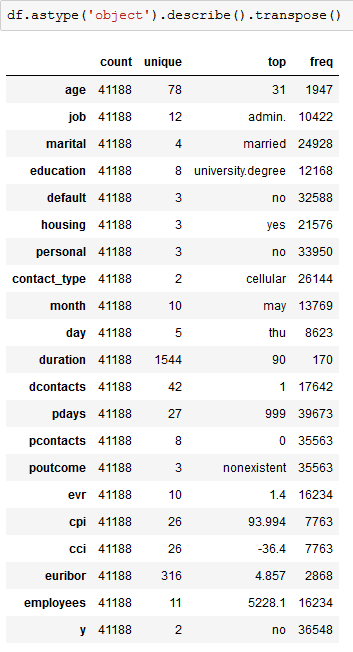
**Interpretation:** The dataset does not need any type casting operations since all the numeric, decimal and string attributes are have their respective data structures.

**df.isnull():**



**Interpretation:** Upon inspecting the dataframe, it is apparent that there are no null values.

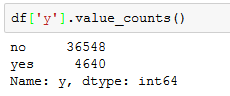
**df.astype('object').describe().transpose():**



**Interpretation:** Using the below code snippet, I can get an idea of most frequently occurring values in an attribute as well as their respective frequencies.

**df.response\_variable.value\_counts():**

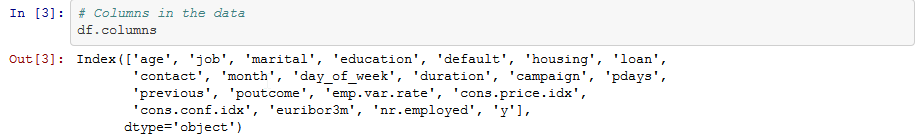
The number of positive responses (yes) is largely fewer than the negative responses (no) implying that the dataset is significantly imbalanced.



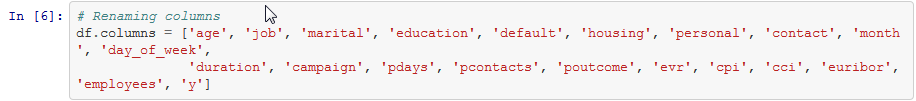
**Interpretation:** Business problems in financial, banking and healthcare industries often have datasets that are massively imbalanced. Considering the reality surrounding these problems, addressing the class balance anomaly is not a major priority, for now. However, later in this report, I use ‘upsampling’ and ‘downsampling’ to address class imbalance.

**3. Data Pre-Processing**

**df.columns**



Following code-snippet demonstrates the updated column names.



**Interpretation:** The attributes (column names) by default are self-explanatory. However, some of these are renamed to make it less confusing.

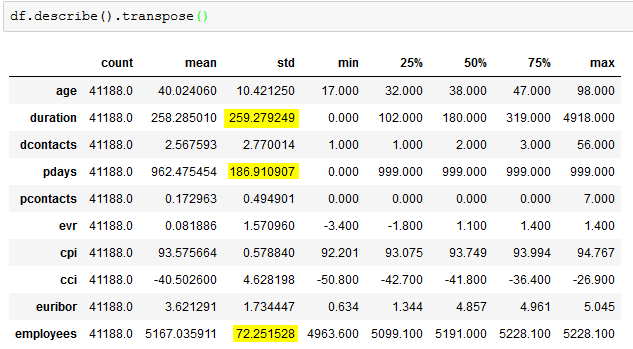
**4 - EDA**

It's always a good idea to get an idea about your data before creating an abstract of your model. Exploratory Data Analysis helps us with this. There are two goals of EDA:

1. Explain  
 2. Explore

***df.describe()***

Using describe() on the dataframe, Python returns the summary statistics of all the quantitative(numeric) variables.



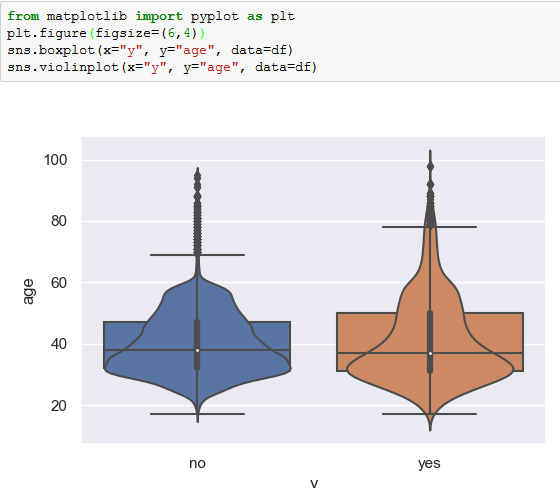
The standard deviations of 'duration', 'pdays' and 'employees' are tremendously large compared to that of other variables. These variables should be investigated to understand the underlying reason for this variability of the values.

As part of Graphical EDA, I plot two graphs for each numeric variable

* Histograms - to understand the distribution underlying the data
* Violin plot - to understand distribution of a variable with respect to the classes of the response variable)

**4.1 EDA with Numeric Variables:**

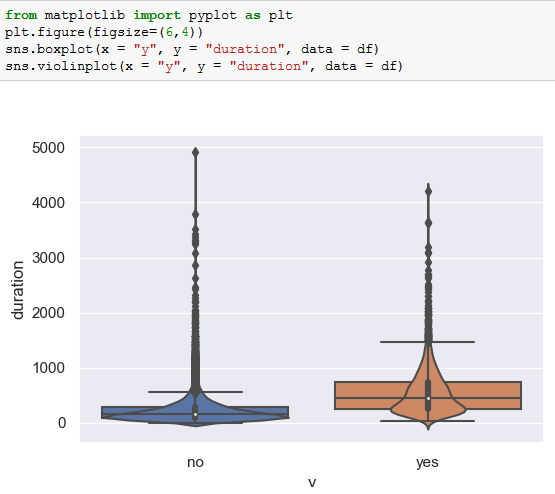
**‘age’:**

****

**Interpretation:** The variance of age of the customers who have rejected the offer is lower compared to that of the customers who have responded positively to the offer. Even though most observations are around early 30's, the mean has been recorded around late 30's for both the classes.

There are significant number of outliers for both classes. However, the outliers for 'no' are widespread. Binning the 'age' variable with respect to 'job' category might provide us better insights.

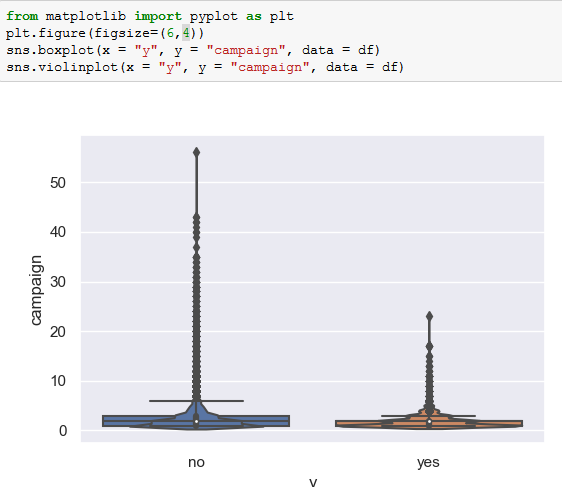
**‘duration’:**

****

**Interpretation:** The variance of class 'no' of the response variable is less compared to that of 'yes' class. Outliers for 'no' are widespread than the outliers of 'yes'.

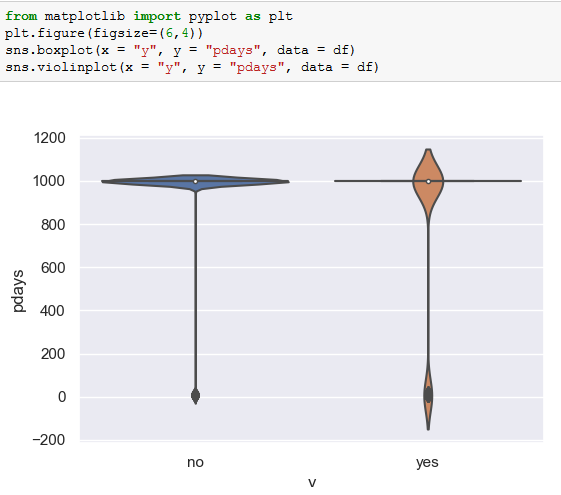
Since the data is widespread, it's a good idea to bin them and include upper bounds.

**‘campaign’:**



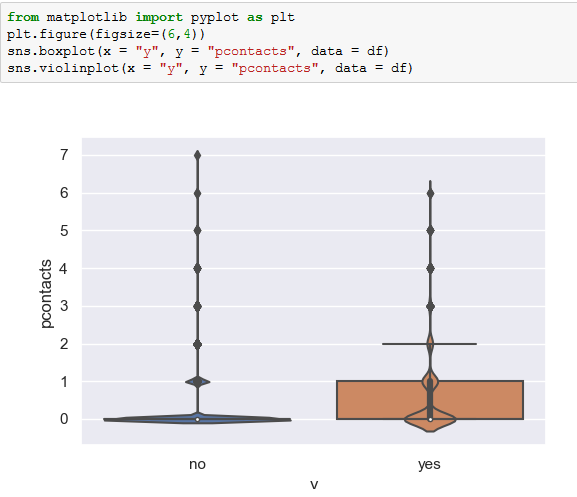
**Interpretation:** 95% of the data points are almost equally distributed for both the classes. However, the outliers are more loosely distributed for 'no' class than 'yes' class.

**‘pdays’:**



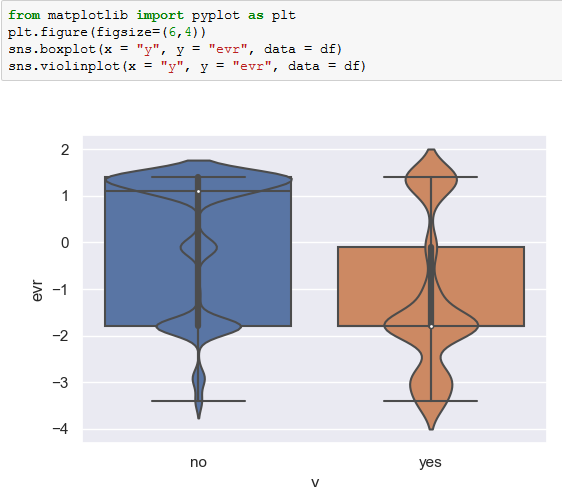
**Interpretation:** According to the data description (Section I), value ‘999’ denotes that the customer has not been contacted, it looks like majority of the customers from 'no' class have not been previously contacted. Same is the case for most of the customers in 'yes' class. However, a decent amount of customers have been contacted before this campaign which can be seen in the first peak of this bi-modal distribution.

**‘pcontacts’:**



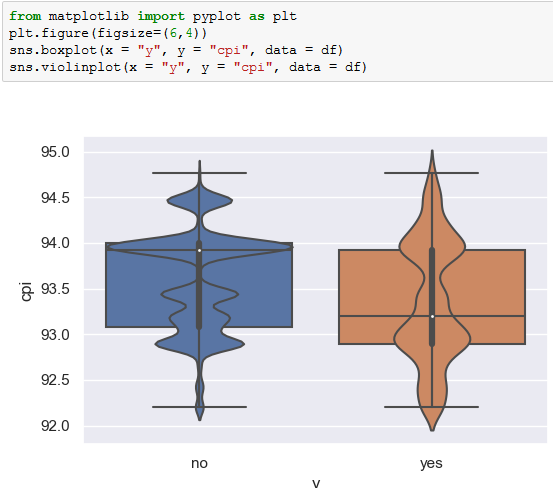
**Interpretation:** The 'no' class has a bi-modal while 'yes' class has a tri-modal distribution. There are few outliers for both classes which do not have significant effect on majority of the data.

**‘evr’:**



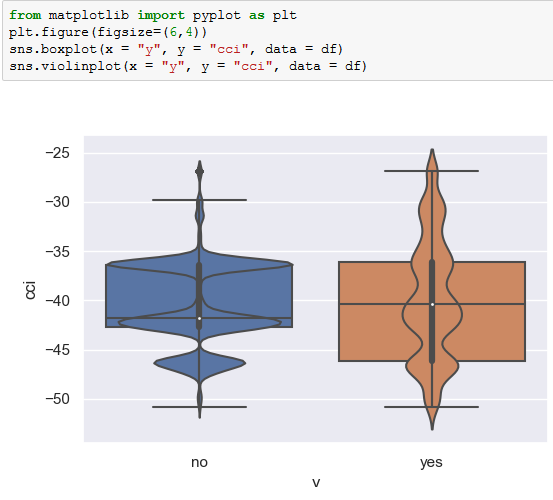
**Interpretation:** Most observations of EVR rate have been recorded between -2 and 2 for both classes. However, the observations of 'yes' class have recorded a significant amount of them with values between -4 and -2.

**‘cpi’:**



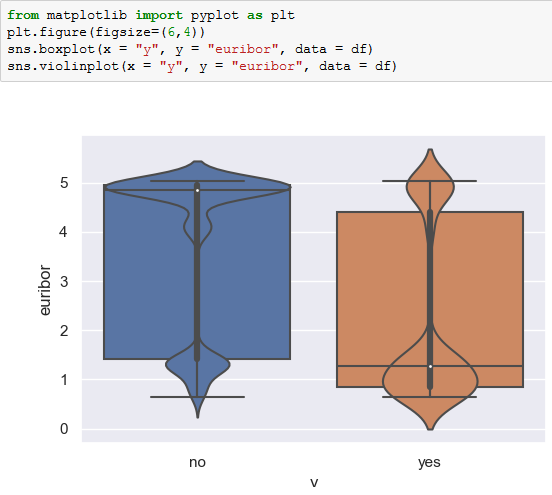
**Interpretation:** 80% of the observations have been recorded between 93 and 94. However, the emans for 'no' class is close to 94 while for 'yes' class is close to 93.

**‘cci’:**



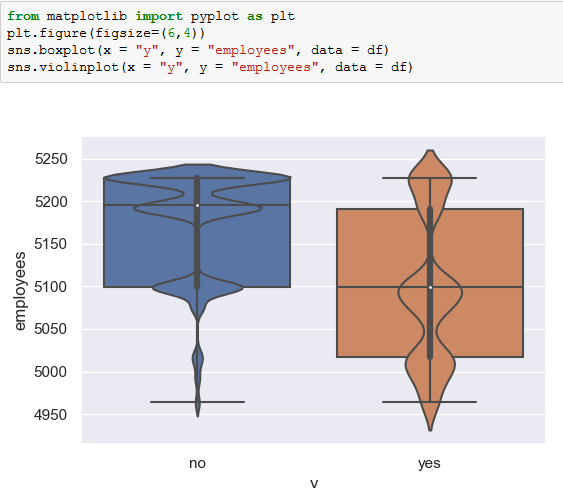
**Interpretation:** The means for both classes are almost equal. However, 'no' class has observations densely at -36 and -43 while 'yes' class has observations significantly spread from -52 to -30 in its multi-modal distribution.

**‘euribor’:**



**Interpretation:** Euribor mean for observations of 'yes' class is around 5 while that of 'no' class is close to 1. However, a decent number of observations have been recorded around 5 as well as 1 for both the classes.

**‘employees’:**



**Interpretation:** The mean number of employees for 'no' class is close to 5200 while the mean number of employees for 'yes' class is around 5100. Data is more evenly distributed for class 'yes' with most observations recorded less than 5150.

**4.2 Socio-Economic Factors Definitions:**

**Evr** - Employment Variation [(EVR)](https://www.quora.com/What-is-meant-by-employment-variation-rate-Does-it-affect-in-any-way-the-financial-decisions-that-an-individual-takes) is essentially the variation of how many people are being hired or fired due to the shifts in the conditions of the economy

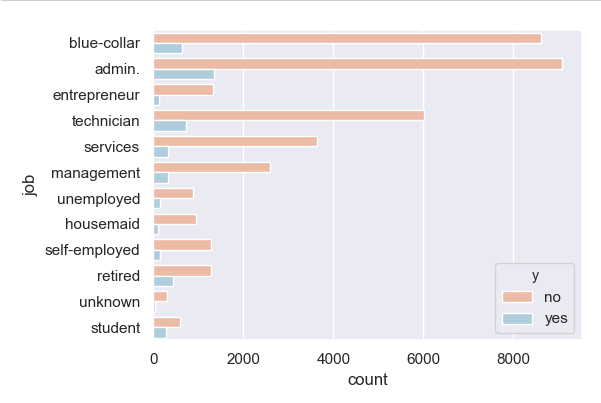
**Cpi** - The Consumer Price Index [(CPI)](https://www.bls.gov/cpi/) is a measure of the average change over time in the prices paid by urban consumers for a market basket of consumer goods and services.

**Cci** - The Commodity Channel Index [(CCI)](https://www.fidelity.com/learning-center/trading-investing/technical-analysis/technical-indicator-guide/cci) measures the current price level relative to an average price level over a given period of time. CCI is relatively high when prices are far above their average.

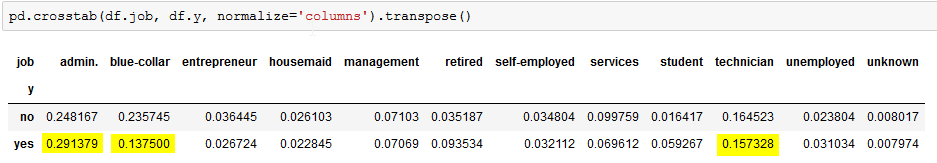
**Euribor** - [Euribor](https://www.euribor-rates.eu/what-is-euribor.asp) is short for Euro Interbank Offered Rate. The Euribor rates are based on the interest rates at which a panel of European banks borrow funds from one another.

**4.2 EDA on Categorical Variables:**

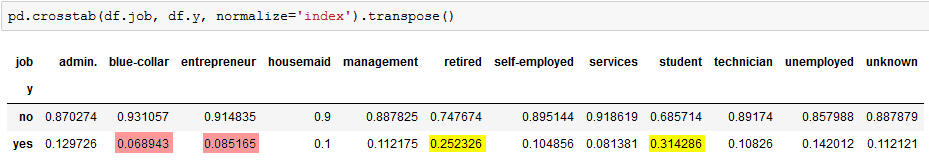
**‘job’:**



**Interpretation:** Admin category has the highest number of positive and negative responses while 'unknown' has the lowest for the both.



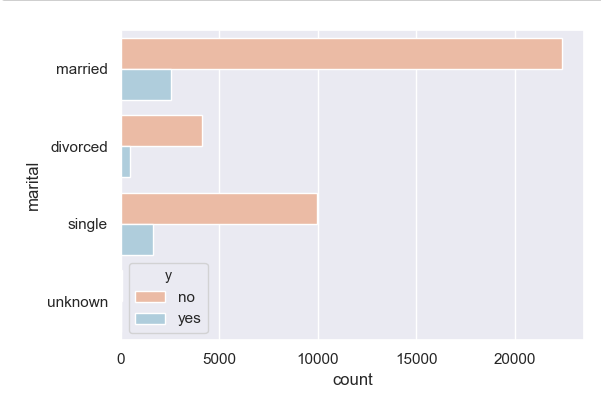
**Interpretation:** At category-level, Admin, Blue-Collar and Technicians contributed the highest percentage of positive response rate.



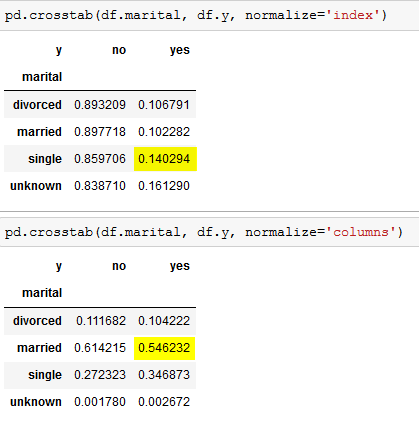
**Interpretation:** At class-level, ‘blue-collar’ and ‘entrepreneur’ (6% and 8%) had the lowest positive response rate while retired and students had high positive response rate (25% and 31%).

Thismeans that ‘admin’ and ‘blue-collar’ jobs were contacted frequently than any other job. However, the highest positive response rate, is among ‘retired’ and ‘student’, rather not ‘blue-collar’ and ‘entrepreneur’.

**‘marital’:**

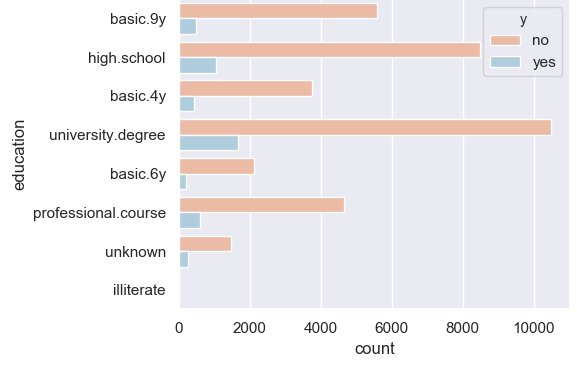


**Interpretation:** Married category has the highest number of positive responses. Around 60% of the people considered for this survey belong to either 'married' category.

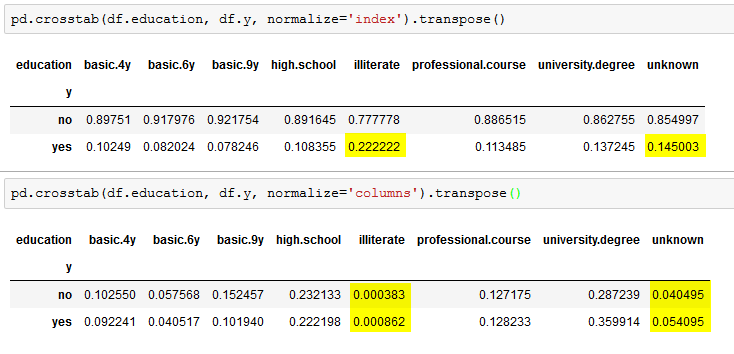


**Interpretation:** At class-level, 'unknown' has the highest positive response rate. At category-level, with 55% 'married' contributed the highest percentage of positive response rate.

**‘education’:**

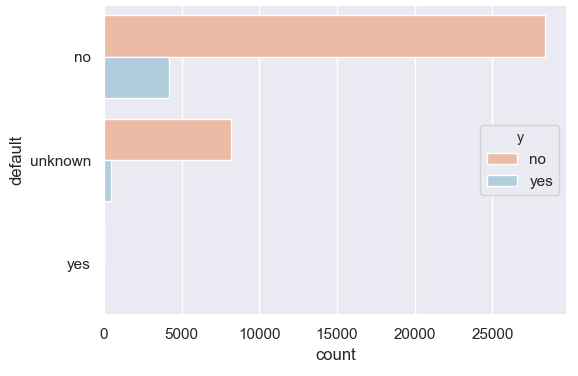


**Interpretation:** Around 5% of the datapoints has an education level of either 'illiterate' or 'unknown'. Though 'illiterate' and 'unknown' contribute to only 5% of the total datapoints, they have the highest positive response rate within categories.

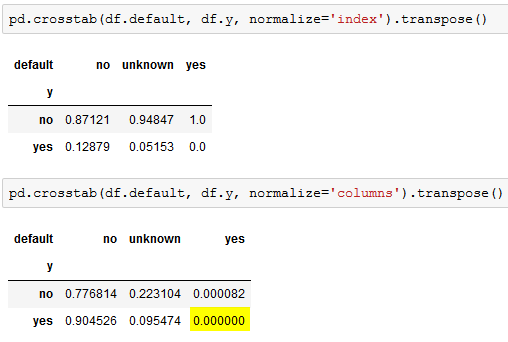


**Interpretation:** The distinction between the education descriptions is very minimal which makes it hard to combine similar classes in the category.

**'default':**

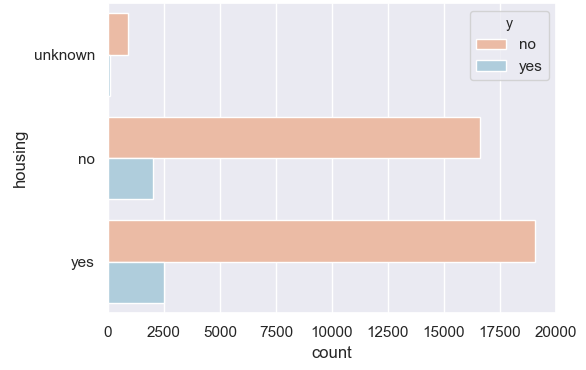


**Interpretation:** Most number of people who were approached do not have a default. Almost 20% of the datapoints are 'unknown'.

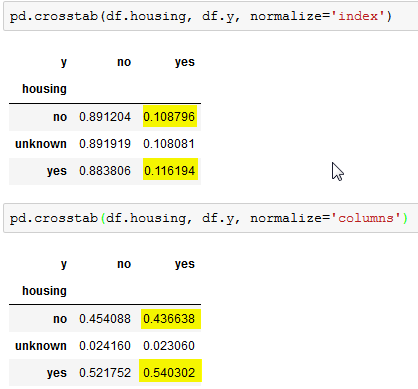


**Interpretation:** None of the customers with loan default have responded positively to the offer.

**‘housing’:**

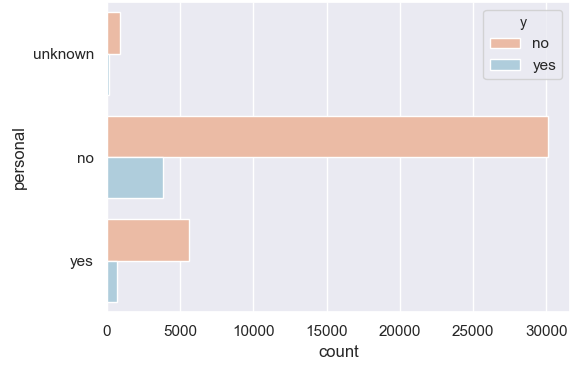


**Interpretation:** Two major classes (people with/without housing loan) are almost equally distributed.

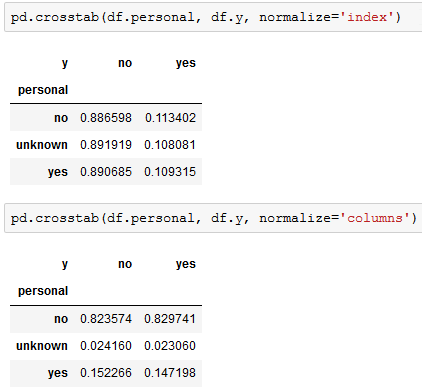


**Interpretation:** Customers with no housing loan have less positive response rate compared to the ones with housing loan.

**‘personal’:**

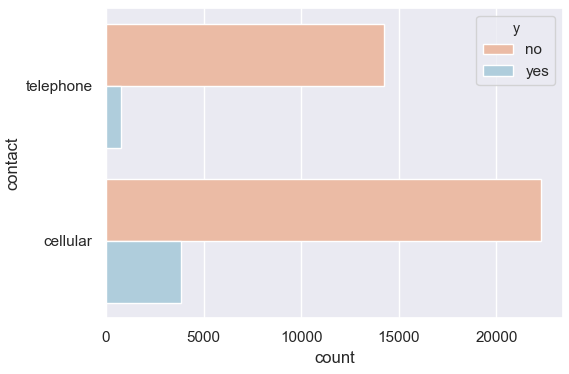


**Interpretation:** Around 82% of datapoints do not have a personal loan on them.

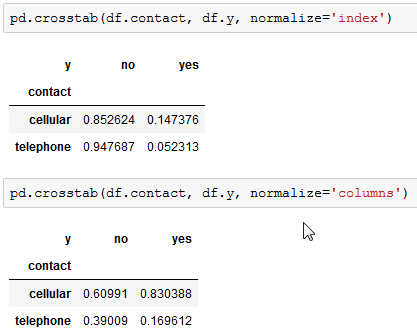


**Interpretation:** Within each category, each of them contributed to the same percentage for a positive response.

**‘contact’:**

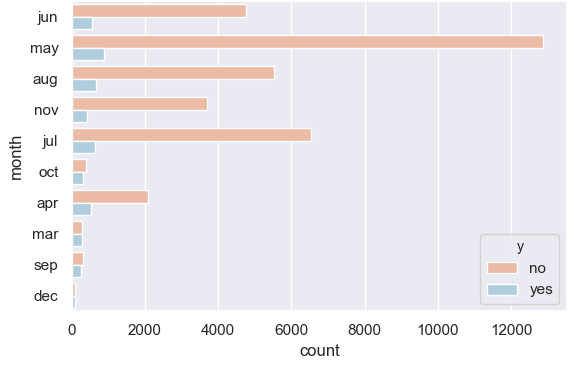


**Interpretation:** Customers with cellular phone are more likely to respond positively.

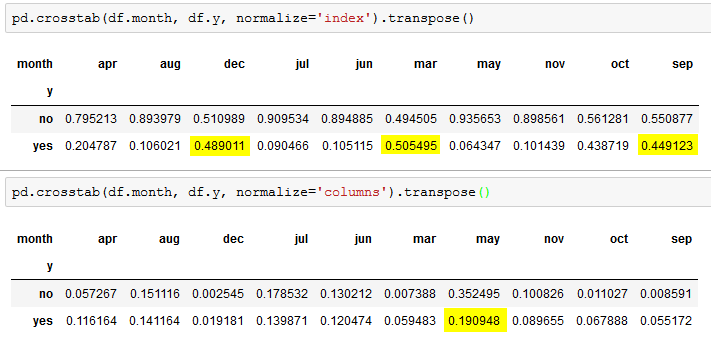


**Interpretation:** At category-level, customers with telephone contributed least to the positive response rate.

**‘month’:**

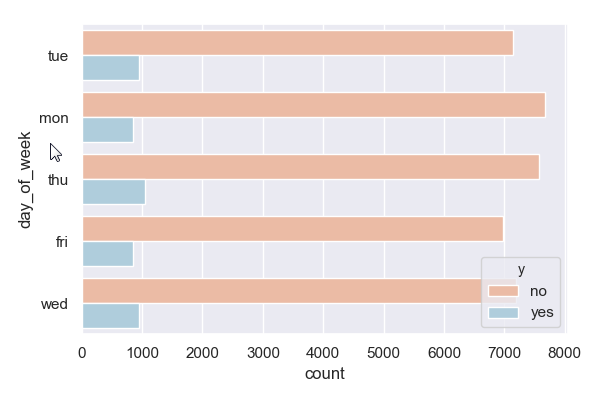


**Interpretation:** Most customers were contacted during the second quarter of the calendar year.

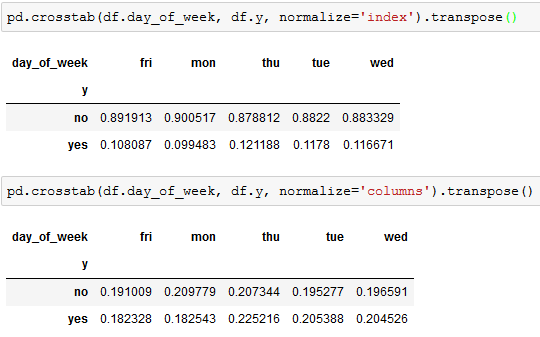


**Interpretation:** Highest positive response percentage was recorded in March, September and December. Highest percentage of positive response was recorded in the final quarter while the lowest percentage was recorded in 'November' which also fall in the final quarter of the year.

**‘day’:**

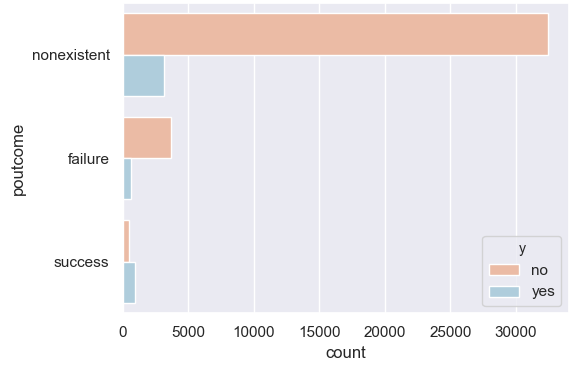


**Interpretation:** All classes are almost equally distributed both in terms of numbers.

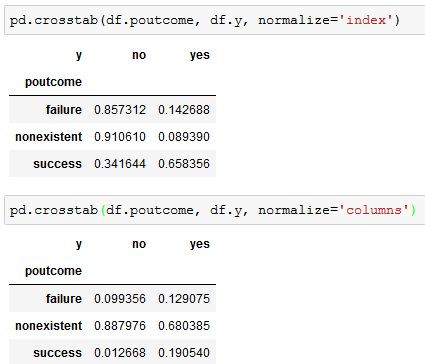


**Interpretation:** All classes are almost equally distributed both in terms of percentages as well.

**‘poutcome’:**

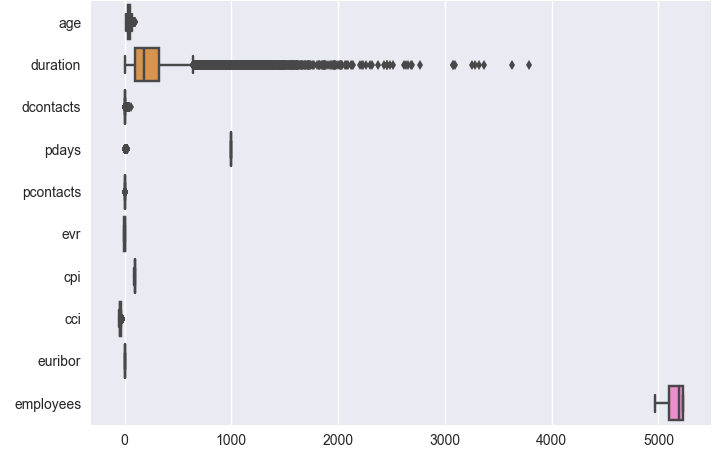


**Interpretation:** Most of the customers have not been contacted for the previous campaign.



**Interpretation:** Of the customers who have not accepted the previous offer, only 14% of them have accepted the current offer. About 65% of customers who have accepted the offer previously, have also accepted current offer.

**4.3 Box plot of all the numeric variables:**



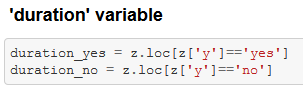
**Interpretation:** We notice that there is a huge disparity among the scale which requires us to bring all the variables to a common scale.

**4.4 Effect of classes of response variable with respect to other variables:**

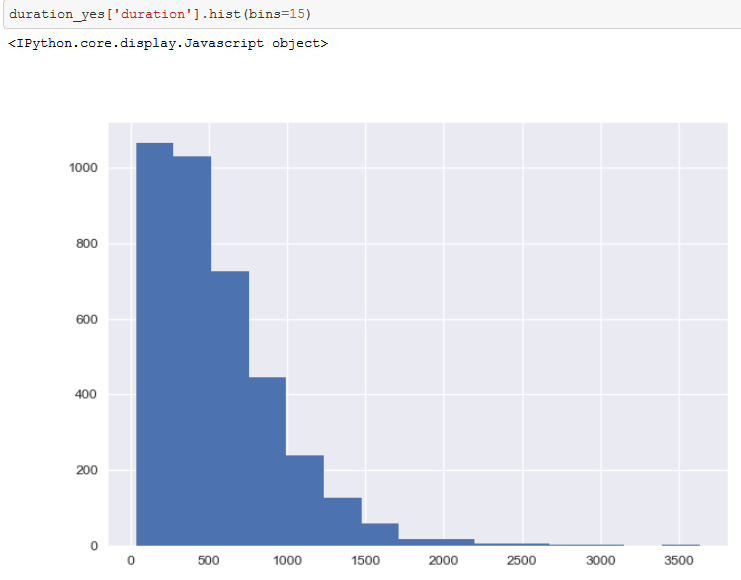
**From the problem statement:**

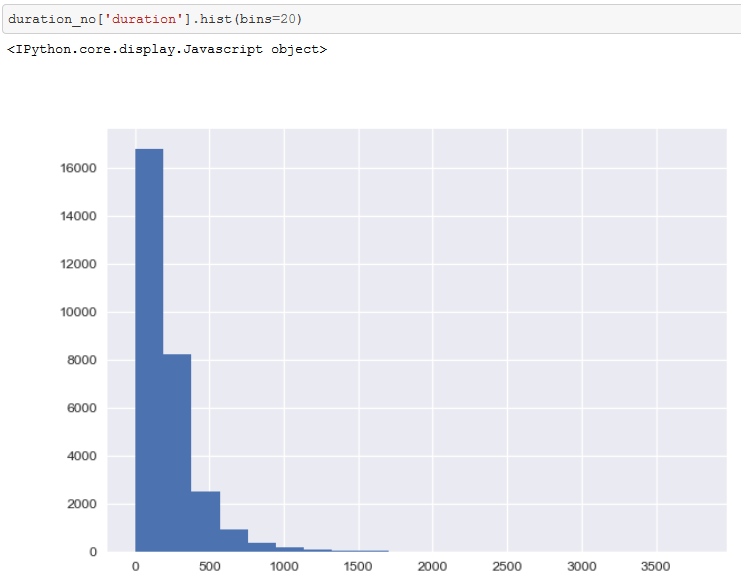
Important note: this attribute highly affects the output target (e.g., if duration=0 then y='no'). Yet, the duration is not known before a call is performed. Also, after the end of the call y is obviously known. Thus, this input should only be included for benchmark purposes and should be discarded if the intention is to have a realistic predictive model.

I have created two dataframes, which hold records that have response variable as ‘yes’ and ‘no’ separately. Idea is to analyze how ‘duration’ variable



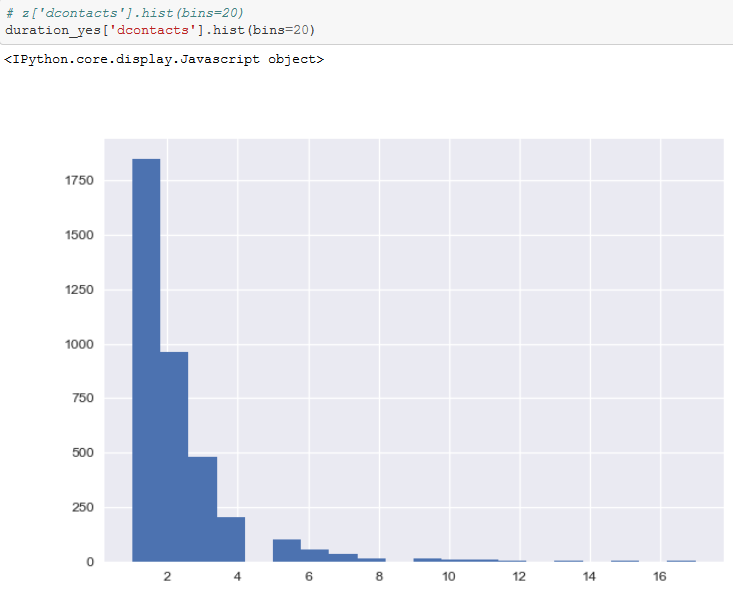
**Classes of response variable vs distribution of ‘duration’ variable:**

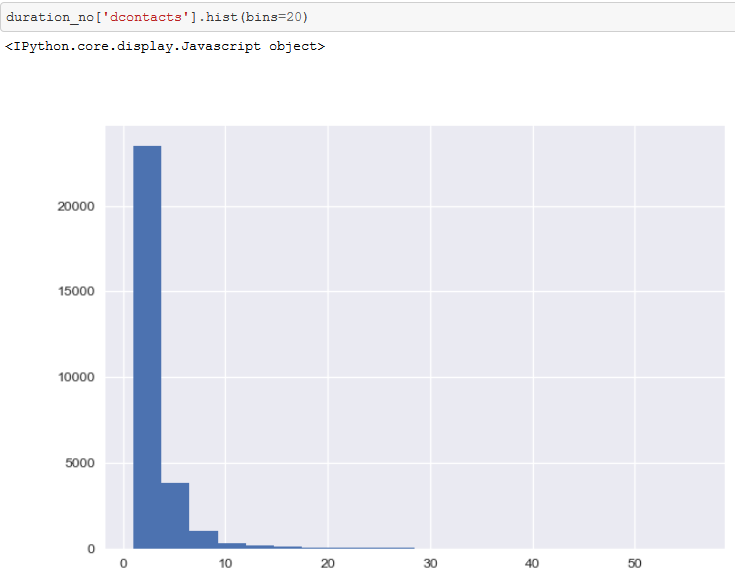




**Interpretation:** From the above two graphs, it is evident that if the call duration is less than 300 seconds, the customer is more likely to says 'NO' and if the call lasts for more than 300 seconds, the customer is more likely to say 'Yes'

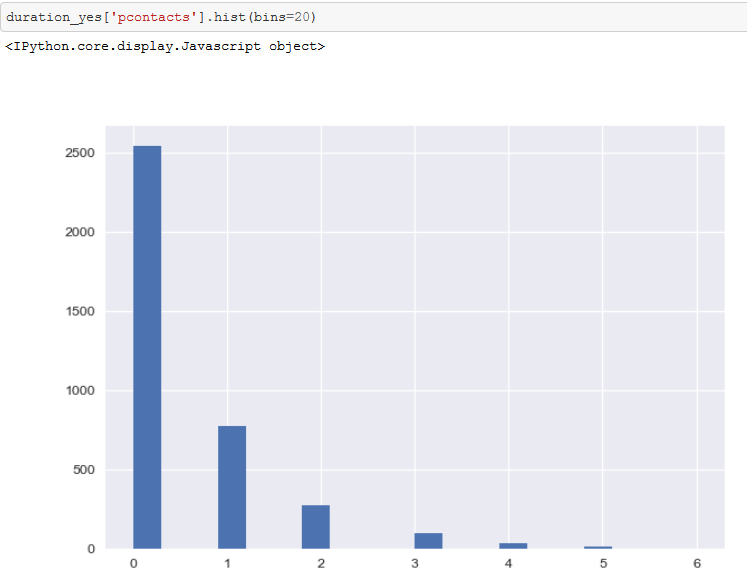
**Classes of response variable vs distribution of ‘dcontacts’ variable:**

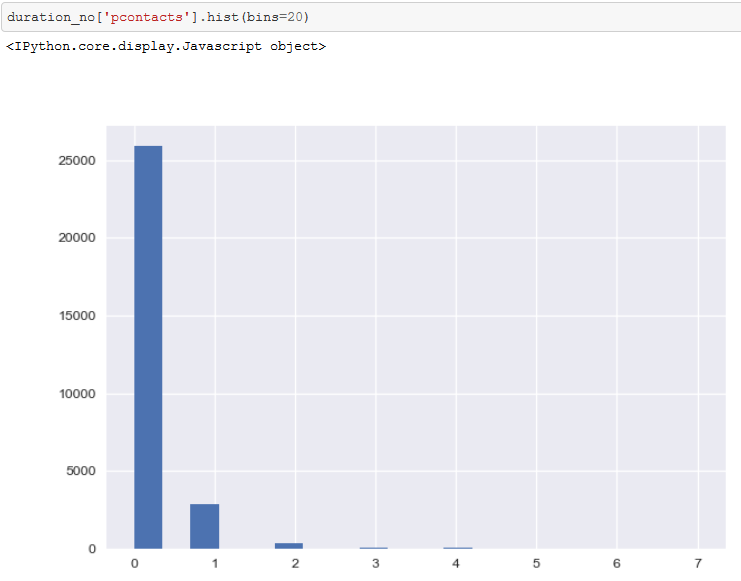




**Interpretation:** The above two histograms are drawn differently. Look at the size of X-Labels. It looks like both the graphs have the similar behavior. It's hard to differentiate. But we can infer that if a customer is made more than two contacts, they are highly likely to say 'yes' than 'no'

**Classes of response variable vs distribution of ‘pcontacts’ variable:**





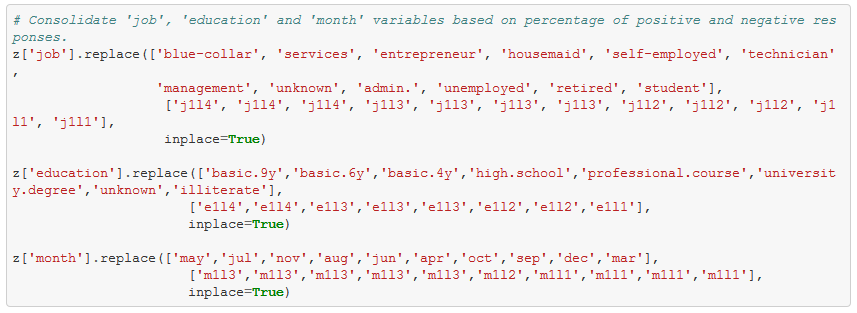
**Interpretation:** The difference is very little between the two dataframes.

**5. Feature Engineering**

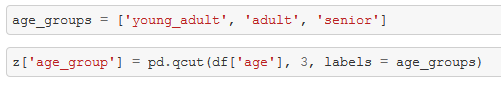
Considering the findings from EDA, the following feature engineering techniques are implemented.

**5.1 Consolidate category classes:**

I consolidate category classes into various levels based on the percentages of ‘yes’ class.



**5.2 Binning the age:**

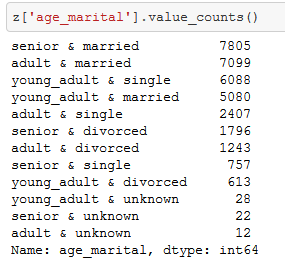


Given the data is highly imbalanced, 'age' is categorised into bins based using 'qcut' rather than 'cut'

**5.3. Categorize 'day' with 'weekday\_1', 'weekday\_2' and 'weekend' classes:**



**5.4 Merging 'marital' and 'age' variable:** 



**5.5 Inclusion and Exclusion of 'duration' column:**

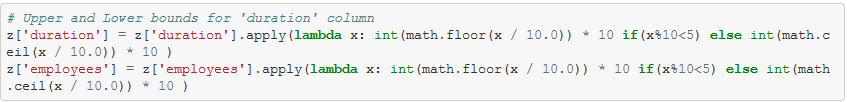
Since this attribute highly affects the output target, I create two dataframes (one with 'duration' column another without).

**5.6 Treating Outliers:**

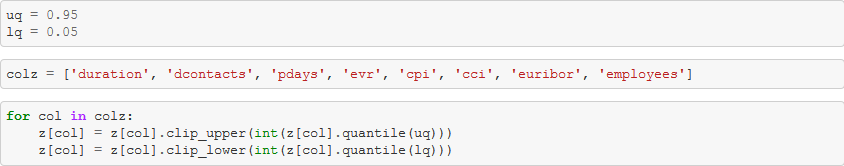
**Idea:**

(a). Replace valid outliers with logarithmic transformation  
 (b). Replace invalid outliers (human-error) with 90th percentile or upper bounds.

**5.6.1. Applying Upper and Lower bounds to 'duration' and 'employees' variable**



**5.6.2. Applying 90 percentiles and 5 percentiles for the lower and upper outliers**

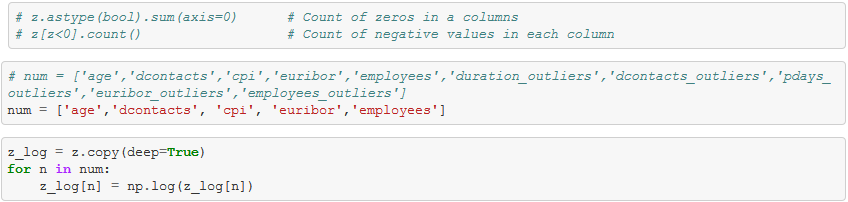


**5.6.3. Apply Logarithmic transformations to invalid outliers**

Creating a new dataframe to apply logarithm transformations. From all the numerical columns, logarithmic transformations is applied to only a few

Excluded attributes

cci - (Negative values)  
 evr (negative)   
 pcontacts - (Zeros)  
 duration – (Zeros)  
 pdays – (Zeros)  
 pcontacts – (Zeros)  
 pdays - (value 999 - means client was not contacted previously)



Above code snippet applies logarithmic transformation to the numerical variables.

**6. Ready for Machine Learning**

**6.1 Standardization and Normalization**

Preprocessed data may contain attributes with a mixtures of scales for various quantities such as dollars, kilograms and sales volume. Many machine learning methods expect or are more effective if the data attributes have the same scale.

Two popular data scaling methods are normalization and standardization.

1. Data Normalization
2. Data Standardization

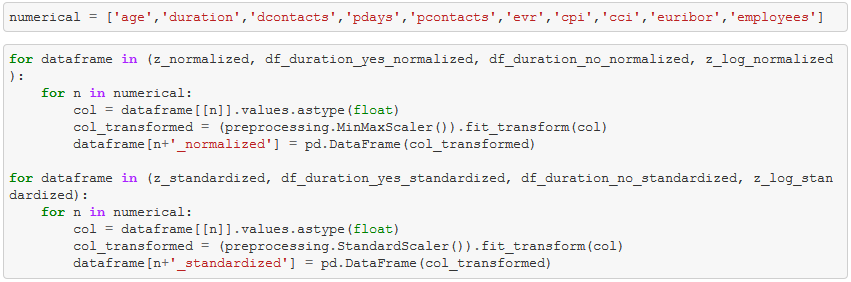
**Normalization:** It refers to rescaling real valued numeric attributes into the range 0 and 1. It is useful to scale the input attributes for a model that relies on the magnitude of values, such as distance measures used in k-nearest neighbors and in the preparation of coefficients in regression.

ML algorithms such as Linear Regression and SVM perform faster on normalized data.

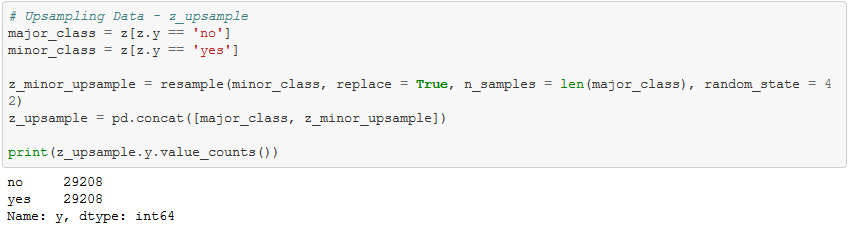
**Standardization:** Standardization refers to shifting the distribution of each attribute to have a mean of zero and a standard deviation of one (unit variance). It is useful to standardize attributes for a model that relies on the distribution of attributes such as Gaussian processes.

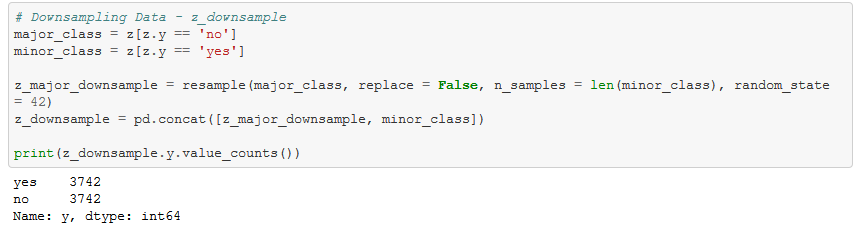
**Which Method to Use**: It is hard to know whether rescaling your data will improve the performance of your algorithms before you apply them. If often can, but not always.

A good tip is to create rescaled copies of your dataset and race them against each other using your test harness and a handful of algorithms you want to spot check. This can quickly highlight the benefits (or lack thereof) of rescaling your data with given models, and which rescaling method may be worthy of further investigation.

****

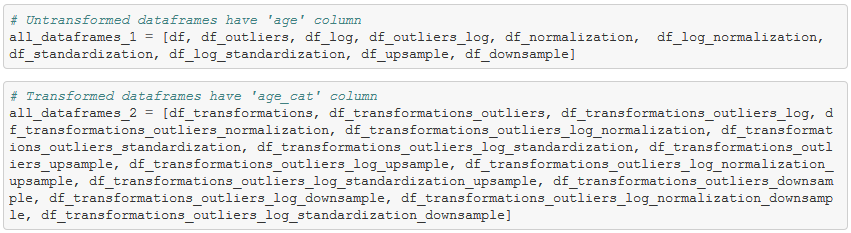
**6.2 Upsampling and Downsampling:**

****

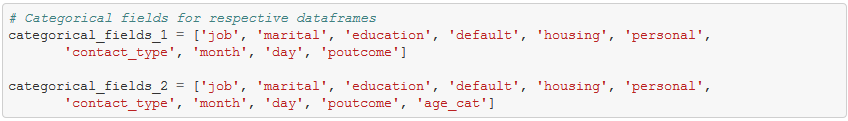
****

**6.3 Dummy Variables:**

Since each dataframe has different categorical columns, all dataframes are divided into two lists.



Categorical columns for each dataframe are divided as two different lists so that they can be efficiently looped and dummied.

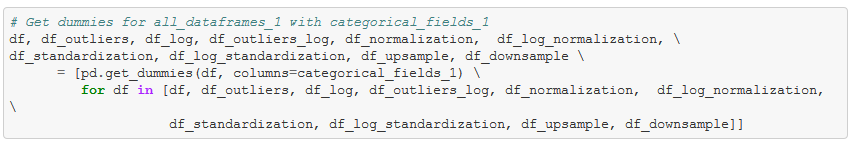


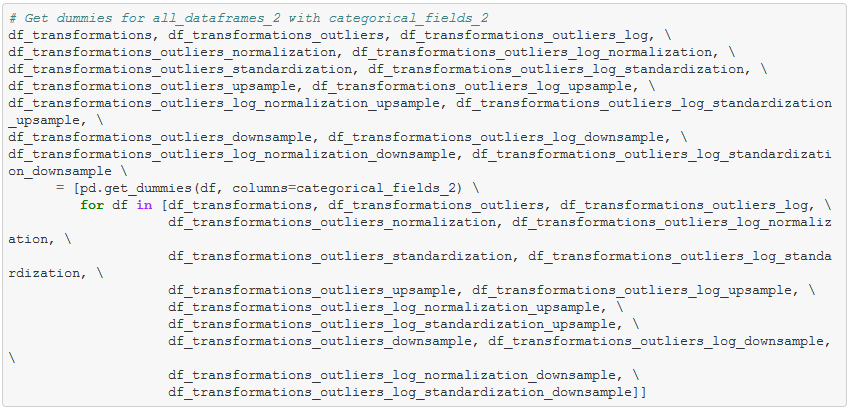
The following code snippet does not replace the categorical columns with their respective dummies in place.





Hence, I had to dummy each dataframe individually.

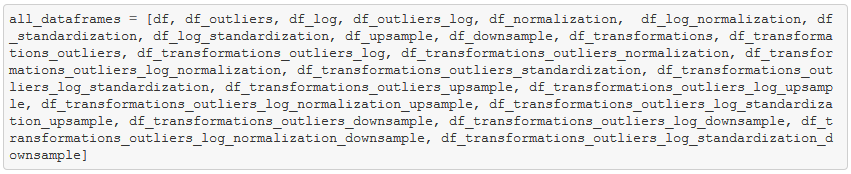




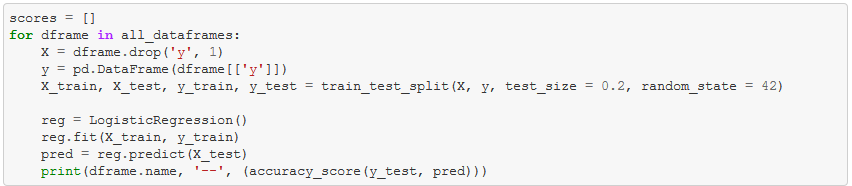
**7. Machine Learning**

**7.1 Logistic Regression:**

All dataframes consolidated into one list.

****

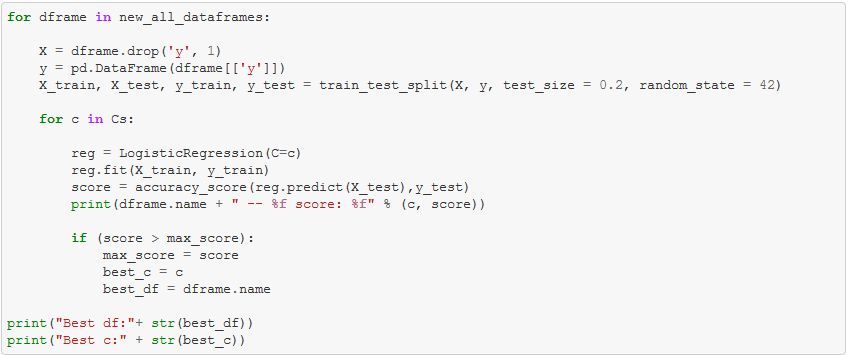
Training all models to find the best model that is more accurate.

****

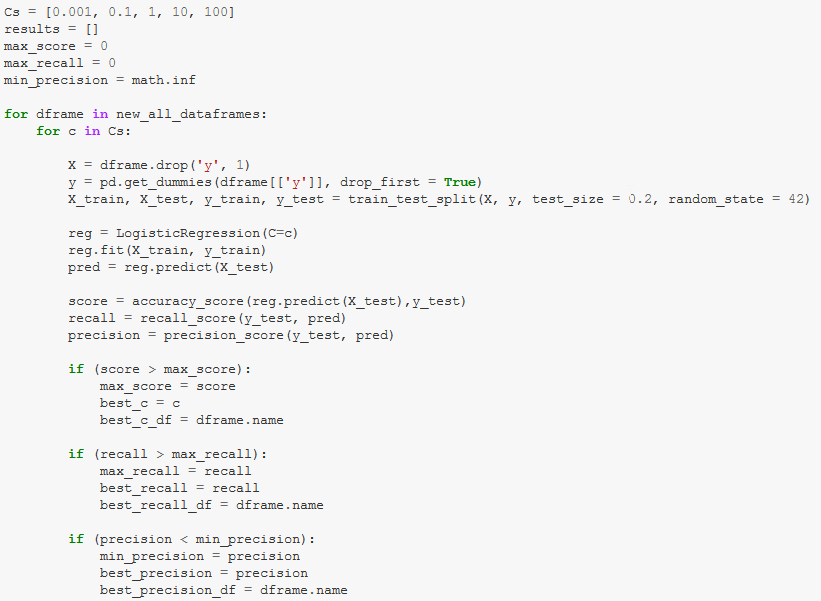
Best df: df\_outliers\_log -- 0.907259043457

**Hyperparamter tuning for model with best accuracy.**

****

****

**Training models to find the best hyper parameters and metrics**

****

Best df: df\_outliers

Best c: 0.1

Best df: df\_upsample

Best recall: 0.889485273878

Best df: df\_log\_standardization

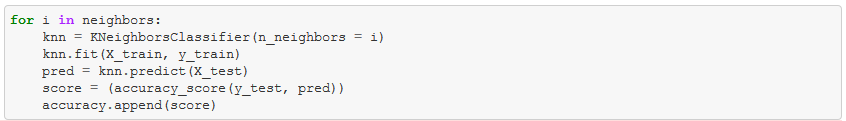
Best precision: 0.586419753086

**7.2 KNN**

Trying multiple neighbors to build a k-nearest neighbor model.

****

Training a knn model to find the best parameters.

****

1 ---- 0.887351298859

3 ---- 0.896941005098

5 ---- 0.902403495994

7 ---- 0.902767662054

9 ---- 0.903131828114

11 ---- 0.904952658412

13 ---- 0.904588492353

15 ---- 0.905438213159

17 ---- 0.906894877397

19 ---- 0.907623209517

21 ---- 0.906409322651

23 ---- 0.905559601845

25 ---- 0.906287933965

27 ---- 0.907987375577

29 ---- 0.907259043457

31 ---- 0.907623209517

33 ---- 0.907380432144

35 ---- 0.907380432144

37 ---- 0.907016266084

39 ---- 0.906894877397

41 ---- 0.906409322651

43 ---- 0.907137654771

45 ---- 0.906652100024

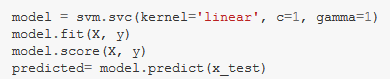
47 ---- 0.907016266084

49 ---- 0.906773488711

All neighbors yield almost a similar accuracy score. Upon further analysis, it is apparent the best model is df\_upsample.

**7.3 SVM**

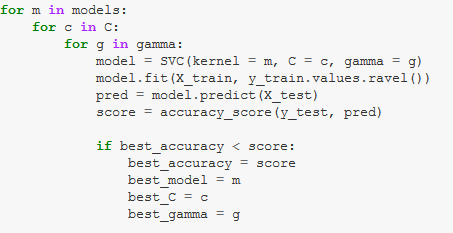
An SVM model with base parameters

****

Various hyper parameters

****

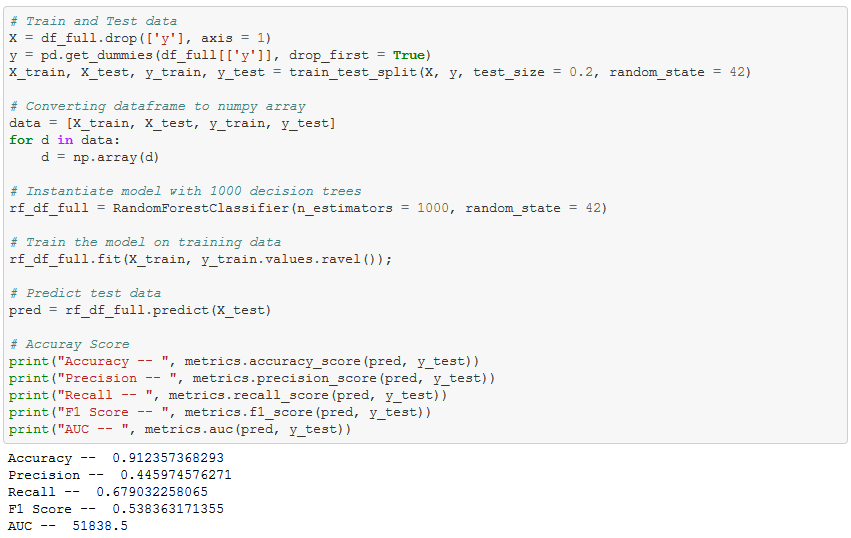
Trying various hyper parameters

****

The best model is df\_outliers.

**7.4 Random Forests with untransformed data**

Initially the data is trained on the base model with no transformations

****

**7.4.1 Randomized Search CV:**

Random Forests has the following hyper parameters:

n\_estimators = number of trees in the foreset  
 max\_features = max number of features considered for splitting a node  
 max\_depth = max number of levels in each decision tree  
 min\_samples\_split = min number of data points placed in a node before the node is split  
 min\_samples\_leaf = min number of data points allowed in a leaf node  
 bootstrap = method for sampling data points (with or without replacement)

Default parameters used in the Random Search:

{'bootstrap': True,

'criterion': 'mse',

'max\_depth': None,

'max\_features': 'auto',

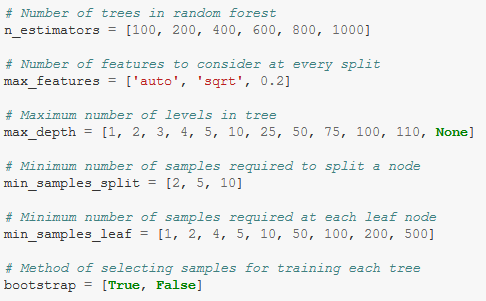
'min\_samples\_leaf': 1,

'min\_samples\_split': 2,

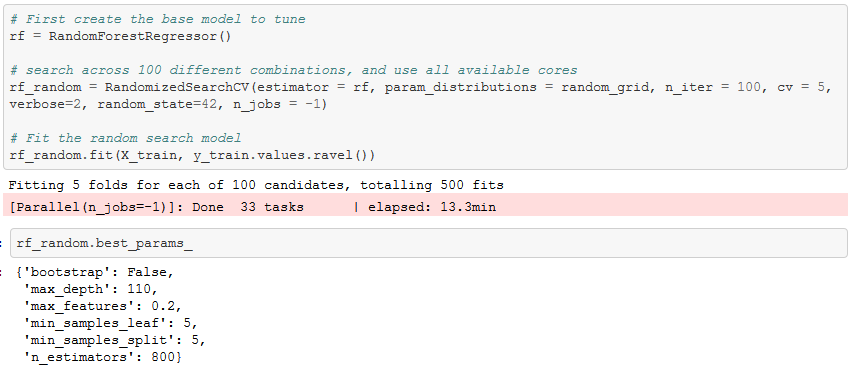
'n\_estimators': 10,

}

Using Randomized Search CV to pick the best parameters. Best parameters can take any values. Efficient approach is to narrow our search to evaluate a wide range of values for each hyperparameter.

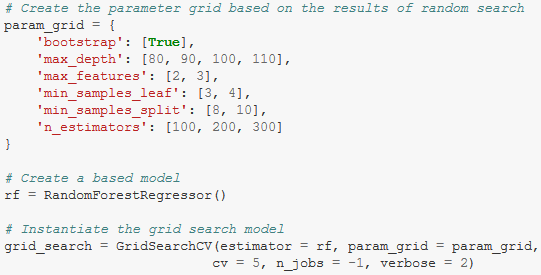


Training the base model with different sets of parameters to find the best set.

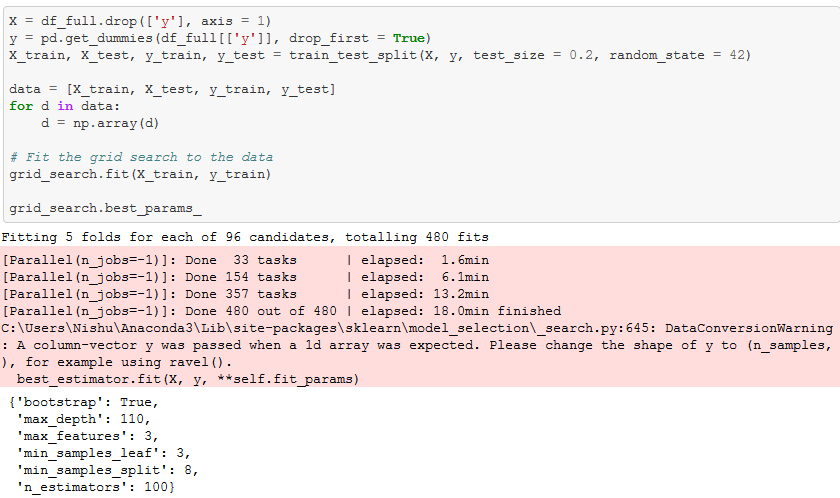


**7.4.2 Grid Search CV:**

Using Grid Search CV to pick the best parameters. This gives us an idea where to concentrate our search.



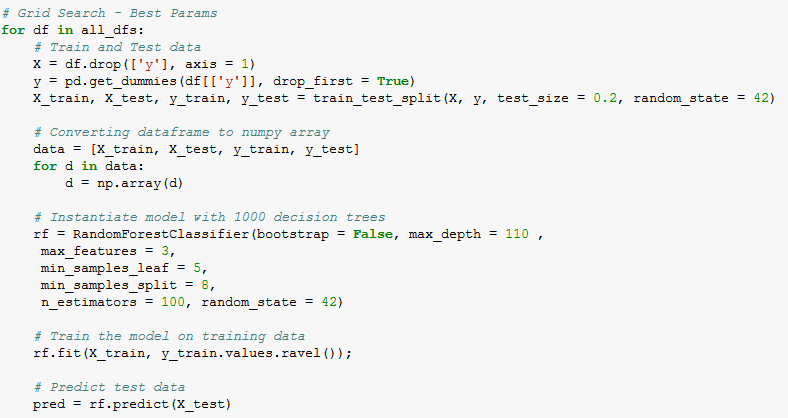
Training the base model with different sets of parameters to find the best set.



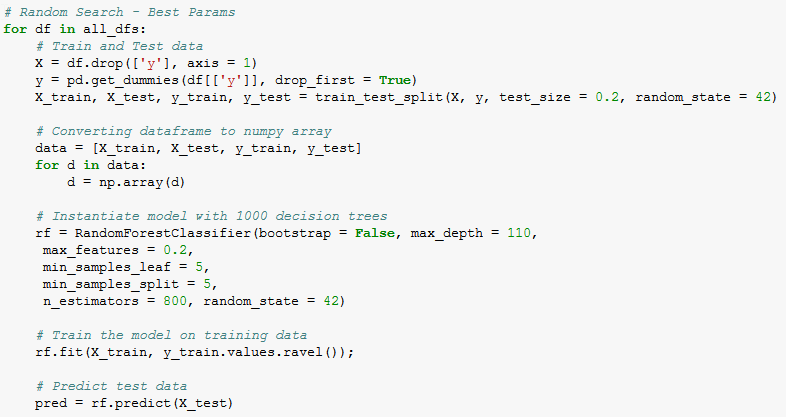
**Note:** Since the goal of this project is to minimize False Negatives (How many did we miss), we focus on getting a recall value close to 100% with a less bad precision value

**7.4.3 Training all the models with grid search CV best parameters:**

Training all models with best parameters of Grid Search CV

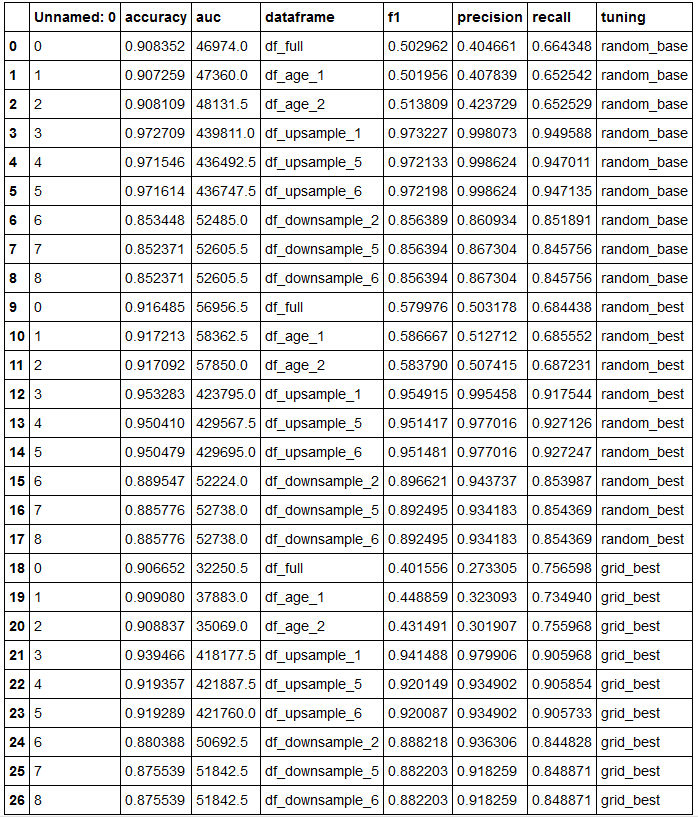


Training all models with best parameters of Random Search CV

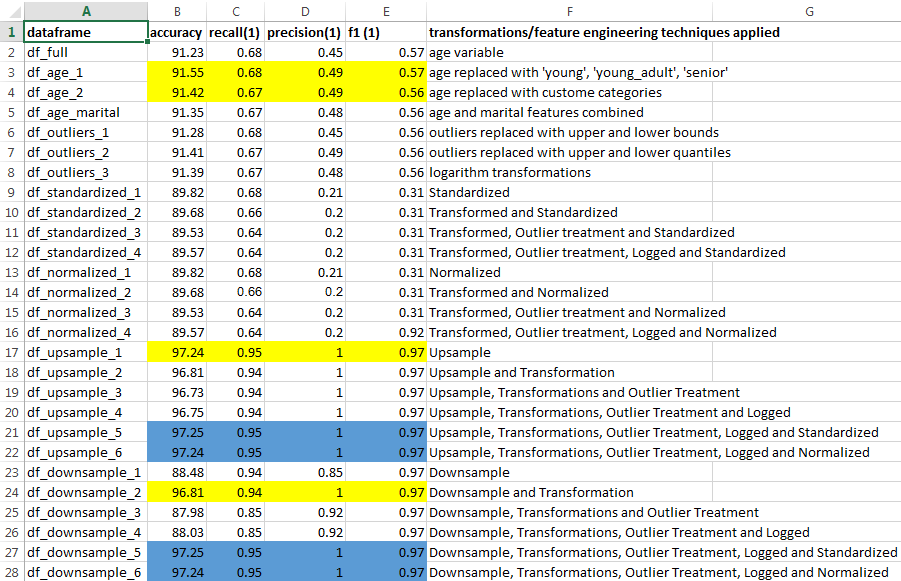


Best parameters of Grid Search CV are chosen over the best parameters Random Search CV considering the computational resources I have.

Below is the table with models are their respective metrics



Here is the list of dataframes tested on the best parameters of Grid Search CV and their respective metrics.



Highlighted are the models with optimal metrics.

**8. Choosing the best model**

From the models highlighted in the above screenshot, df\_age\_2 is the model that yields a better results on the test data.

Other models are not chosen (upsample and downsample) considering the weights each classes are given when the data is either upsampled/downsampled.

**9. Other potential data sets I could use**

The data provided could actually be considered very rich in terms of predicting the client’s behavior for a given campaign. However, given additional data pertaining to client’s financial spending such as income disposal, large credit purchases, demographic of the client.