**1. Data Collection**

A perennial problem of any doctor’s office is the vexing no-show. A no-show is frustrating because it displaces a patient who would have shown up, perhaps even more importantly, medical office’s bottom line ends up missing out on revenue. What if there were some way to predict and thus mitigate the occurrence of no-shows? To this end I will examine 110,527 medical appointments recorded in Brazil over a period of about a year in 2016. The goal is to create a model that can predict with a certain degree of certainty the likelihood of a no-show. Afterwards, we will make some prescriptions that a medical office can follow that are likely to decrease their occurrence of no-shows. The initial features (before feature engineering) that I acquired from Kaggleis below:

01 – PatientId - Identification of a patient

02 – AppointmentID - Identification of each appointment

03 – Gender- Male or Female . Female is the greater proportion, woman takes way more care of they health in comparison to man.

04 –AppointmentDay - The day of the actuall appointment, when they have to visit the doctor.

05 –ScheduledDay- The day someone called or registered the appointment, this is before appointment of course.

06 – Age- How old is the patient.

07 – Neighborhood - Where the appointment takes place.

08 – Scholarship- True of False . Basically was the patient receiving subsidies.

09 – Hypertension- True or False

10 – Diabetes- True or False

Alcoholism- True or False

11. Handicap- True or False

12. SMS\_received - 1 or more messages sent to the patient.

13. No-show - True or False.

**2. Data Wrangling**

* I explored the data to see the type of data for each column, the size of the data, and checked the dataset for missing values. The dataset contains a total of 110,527 entries and 14 columns. Fortunately, that this dataset has no null values. The two columns AppointmentDay & ScheduledDay were read in as objects, I then converted to datetime.
* After the datetime conversion, I thought that our later modeling may gain from being able to determine the day of the week, as well as a categorical time of day. Four additional columns were made, the first two using dt.days attribute on the two datetime columns. A second set of two columns were derived from the datetime columns, these columns were made using the binning pd.cut function combined with the pd.hour attribute. The new categories: Very Early Morning (0-8), Morning (8-12), Afternoon (12-18), Night (18-24).

**3. EDA**

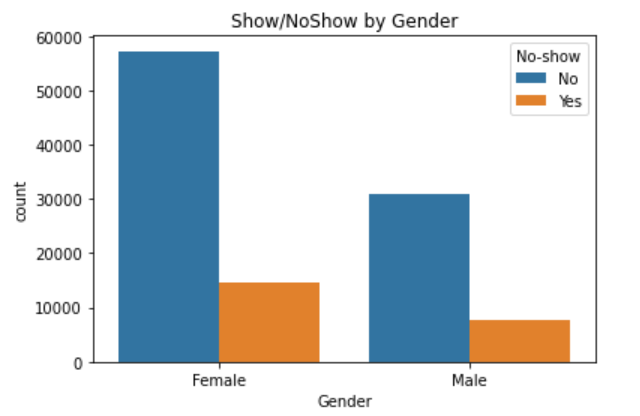
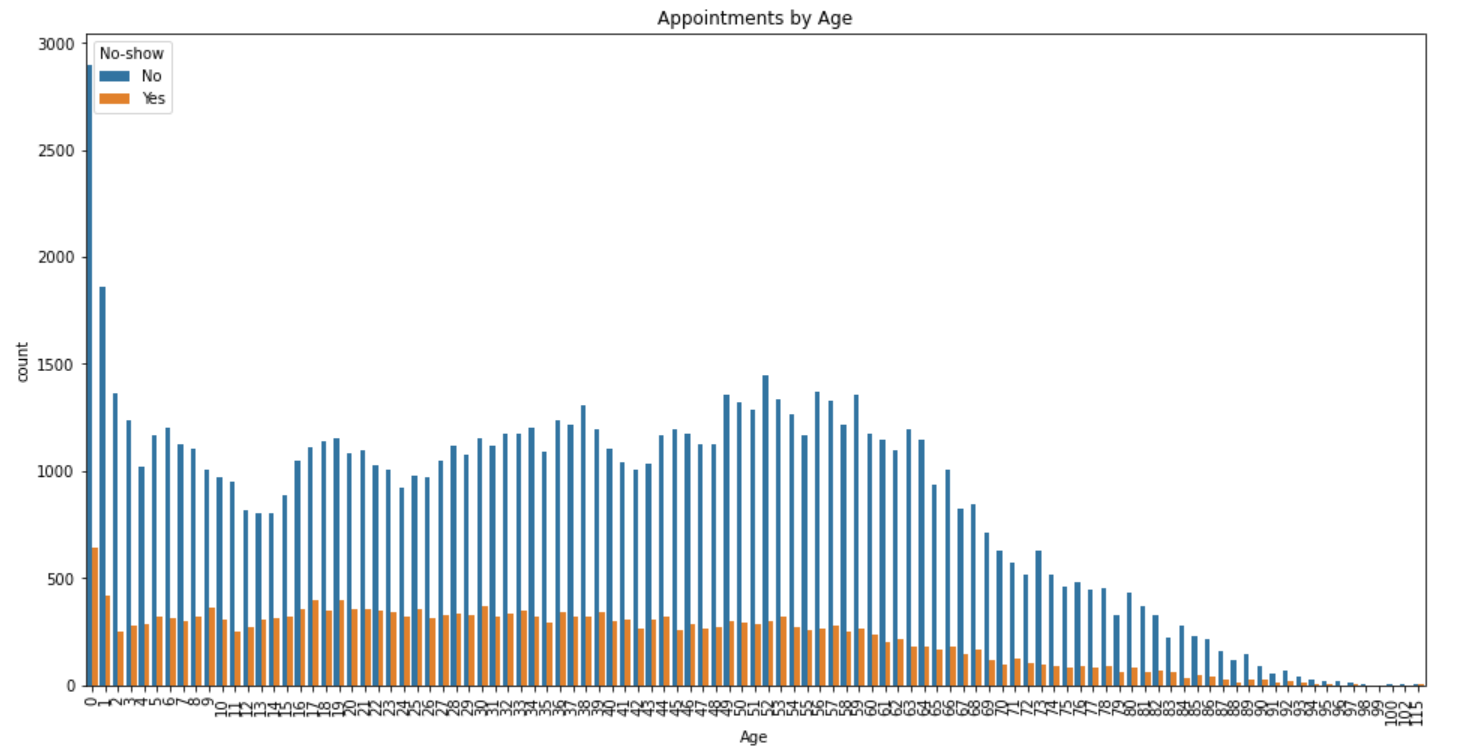
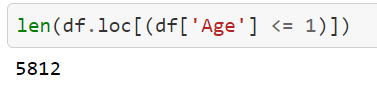
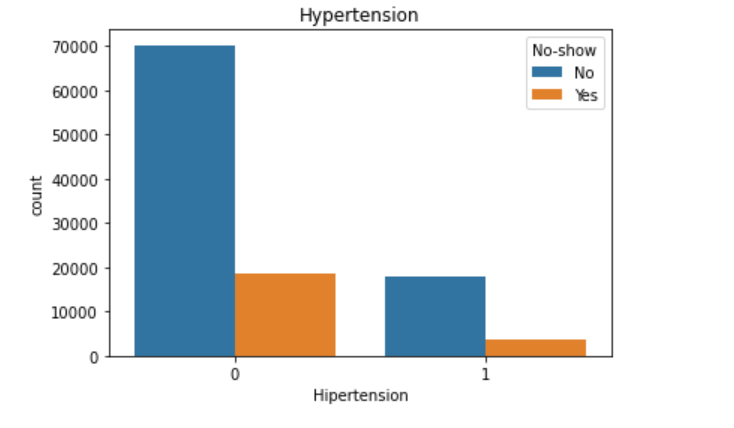
* Determine if natural outliers or erroneous data entry
  + With Age there appears to have been a patient with a listed age just over 110. That is very old but not impossible, I will decide to keep it.
  + Chart, box and whisker chart

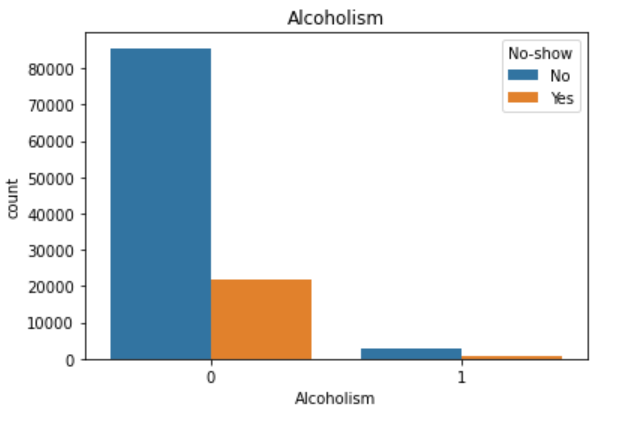
    Description automatically generated
  + In Age column there was one entry that had the age -1. Since it occurred only once we though it safe to delete the whole row.
* Interesting Means
  + Age mean is approx. 37; Std Dev is 23
  + 9.5 mean Interval (# of days b/w ScheduledDay & AppointmentDay); Std Dev is approx. 15 days
* Are any values in a feature disproportionately represented in no shows?
  + Table

    Description automatically generated
  + Table

    Description automatically generated
  + Saturday is the high7est % of attended visits, but there was only 24 visits on that day over the time frame. Friday comes out slightly on top for highest % of no shows. Tuesday had the most scheduled appointments and Thursday and Friday (the end of the week) have the least amount of scheduled visits.
* Let's look to at the neighborhoods that make up the extremes in our % of no-shows over all visits within that neighborhood. We'll do loc list search by creating lists out of the top/bottom 5 Neighborhoods.
  + The least amount of confirmed no-show as a % is: Table

    Description automatically generated
  + The most amount of confirmed no-show as a % is: Table

    Description automatically generated
  + Our top 3 best-attended %-wise are extreme outliers in terms of total number of scheduled visits. Our worst-attened %-wise is an extreme outlier. These could be good candidates for deletion before modeling.
* Overall, 20.19% of all scheduled appointments resulted in a no-show.
* Gender was investigated to determine whether it may play any factor into no-showing. Women comprise approx. 65% of those who no-showed and they make up about 65% of the total scheduled visits. It appears that gender plays no measurable role in no-shows.¶ 
* Age & Distribution 
  + 
  + There are an extremely high number of visits for age 0 and a lot for age 1. 0 could be an entry for when they don't know the age thus inaccurately pumping up that value's count. But the likelihood of that possibility seems diminished when you take into account that there are a very high amount of 1s, the value immediately above 0. Presumptively, it would make sense that these two age groups predominate Age count because young children often visit doctors more often than most age ranges up until late life where the elderly may visit doctor more often.¶
* Hypertension & No-shows 
  + Those with hypertension were not more likely to no-show than those who did not, or put differently, patients marked as having not having hypertension were slightly more likely to no-show up for their scheduled appointment. (21% vs. 17%)
* Diabetes & No-Show Chart, bar chart

  Description automatically generated
  + Those with diabetes were about 2 percentage points less likely to no-show. (18% vs 20%)
* Alcoholism & No-shows
  + 
  + Virtually no difference between alcoholism diagnosis and no-showing (20% v 20%)
* Handicap & No-shows
  + Chart

    Description automatically generated
  + People that have at least one handicap are about 2% less likely to no-show. (18% v 20%).
* SMS & No-shows
  + Chart, bar chart

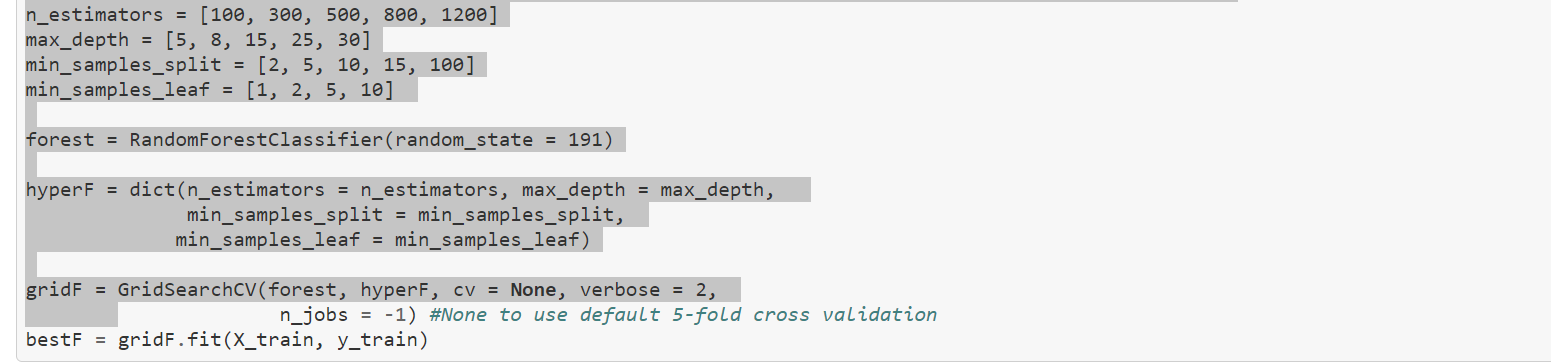
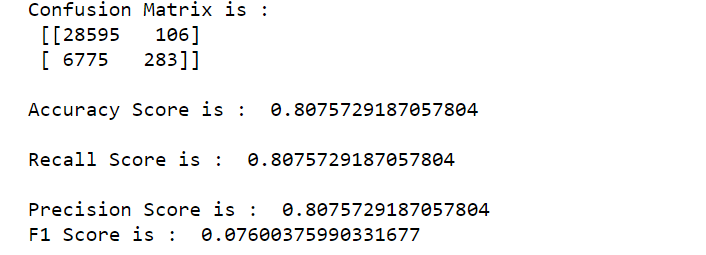
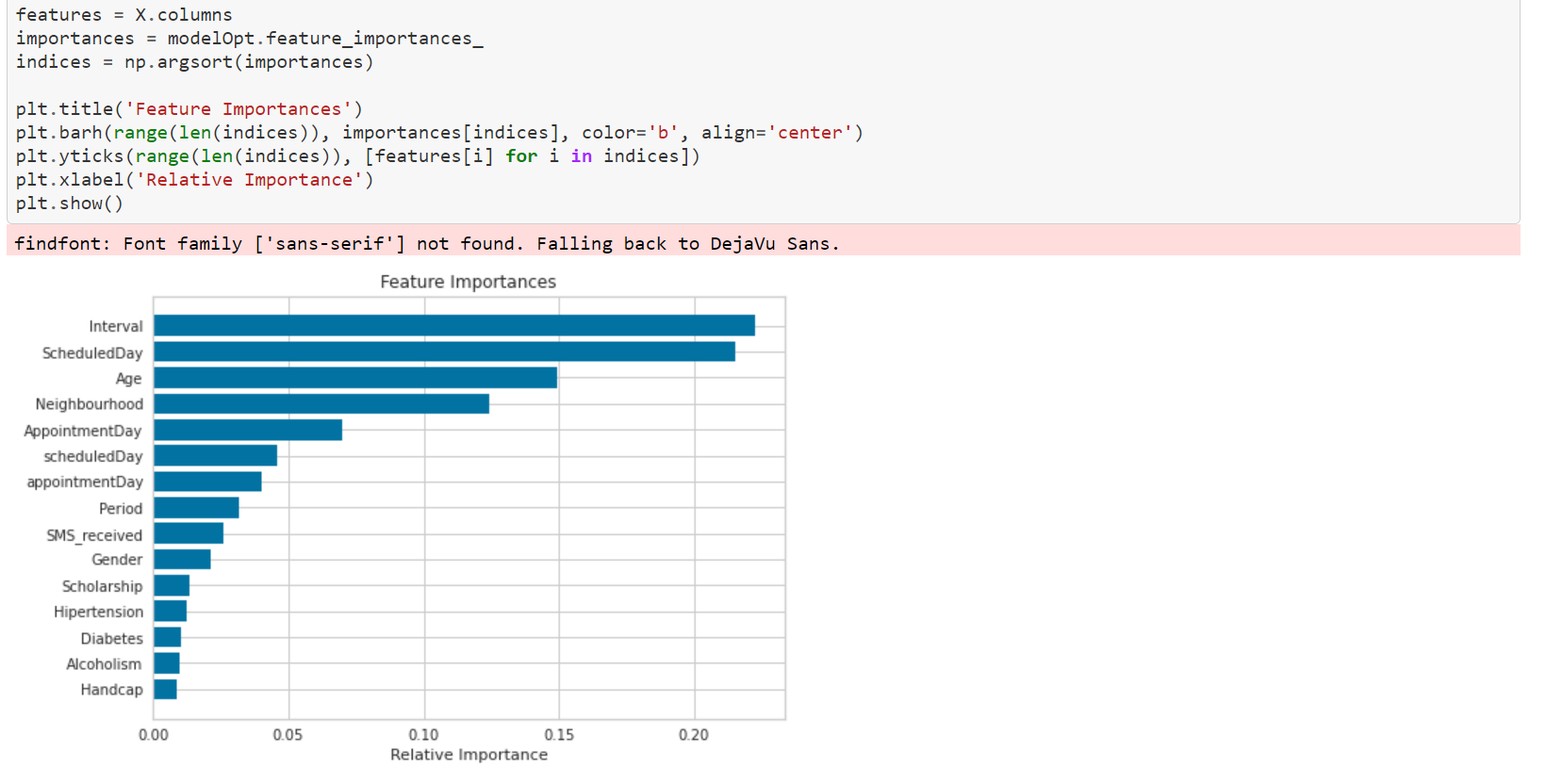
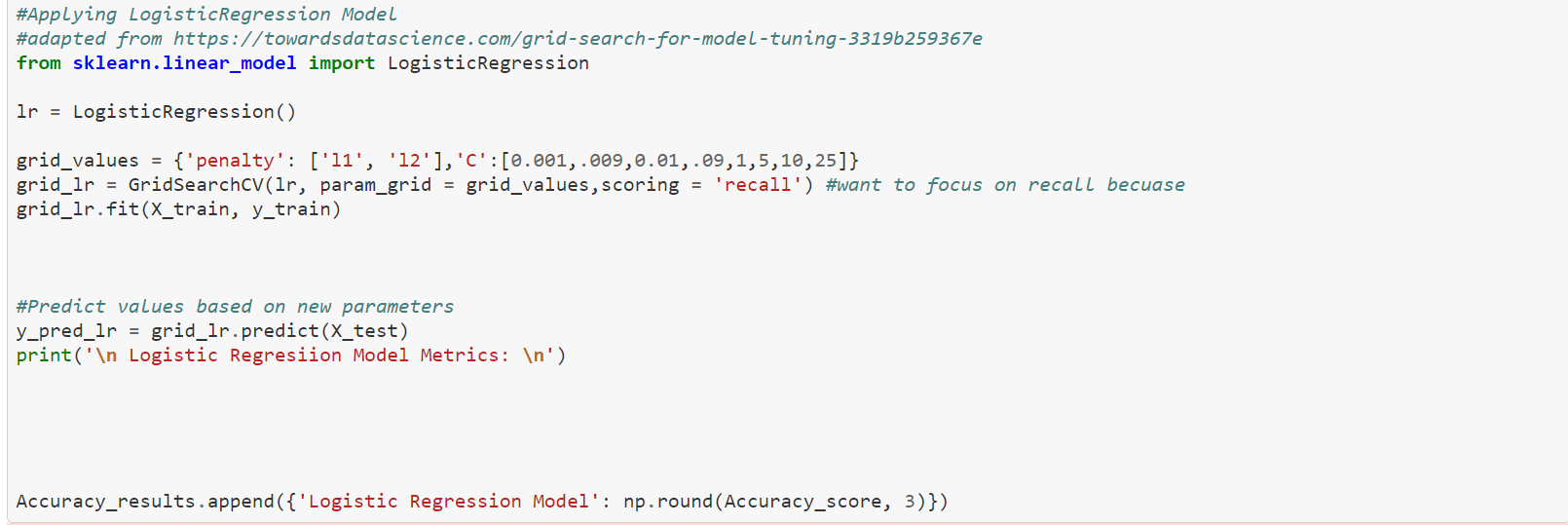
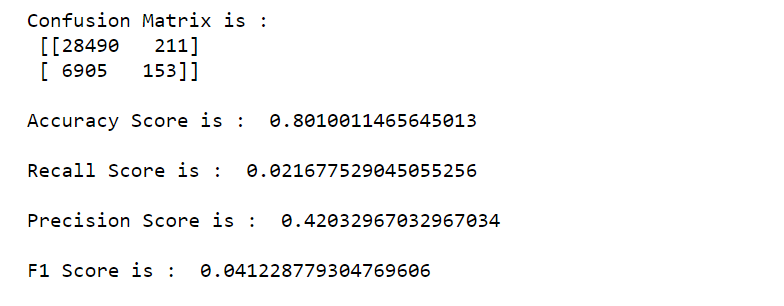
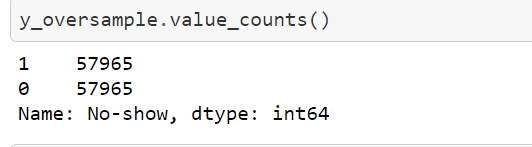
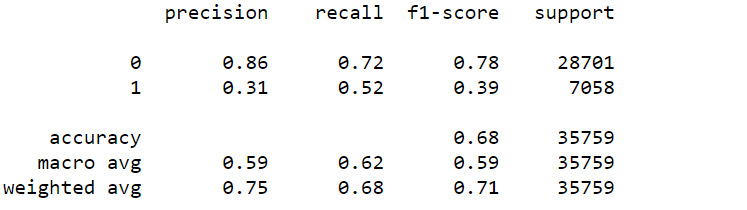
    Description automatically generated
  + Those who received an SMS were far less likely to no-show. (28% vs 17%)
* Intervals
  + There were many unique values created in when we differenced the appointment date from the scheduled date. 127 unique values to be precise. There was at least one occurrence of a vlue up 97 days, then there were about 30 unique occurrences after that with max interval value being 178. With such a high number here we may want to think about segmenting these into blocks/bins.

**4. Preprocessing**

* Intervals
  + In order to not have our models afflicted by outliers that provide little explanatory value we excised all rows whose Interval value was greater than 58 using a mask filtering.
  + 58 was chosen because it resided at the 98% quantile and to up 99% was a massive jump when considering the jump from 97% - 98%.
  + Partially inspired by a post at <https://machinelearningmastery.com/how-to-prepare-categorical-data-for-deep-learning-in-python/>, I developed a function that could prepare a dataset amenable to RandomForest modeling.
    - Graphical user interface, text, application, email

      Description automatically generated
    - Data types are all int64
    - df\_out argument to DataFramerMapper. This indicates that a dataframe should be returned instead of a numpy array when you transform your data, which we’ll do after we fit it

**5. Modeling**

* RandomForest
  + Hyperparameters determined using exhaustive grid search
  + 
  + Best params found to be:
    - n\_estimators: 800
    - max\_depth: 15
    - min\_samples\_split: 2
    - min\_samples\_leaf: 1
    - Mean cross-validated score of the best\_estimator: 0.8016363700874904
  + Confusion Matrix for the RandomForestClassifier is as follows: 
  + The feature importance for the optimal rfc is: 
* LogRegression
  + Our grid search looks like this:
  + 
  + This model’s confusion matrix is markedly worse on almost all fronts: 
* Improving our modeling: We have very imbalanced classes 80 (negative)/20 (positive) so we employ SMOTENC, but first we will undersample the majority class before the oversampling of the minority class
  + SMOTENC will create new synthetic data for both our continous and categorical data. It uses nearest neighbors to choose the value of the syntehotic sample. We in essence are going to add non-existent, but “realistic” no-shows in order to even our target variable ratio.
  + New ratio:
  + 
* The results of the modeling results post-SMOTENC:
  + Log Regression: 
  + New post\_SMOTENC rfc best params:
  + 
    - With a best\_score of: 0.8170965237643404
    - Classification report:
    - Table

      Description automatically generated

Recommendations:

* When an interval gets large enough it is paramount that an SMS be sent (It would have been huge if we could have had the date the message(s) was/were sent.)
* We do now that receiving at least one SMS message shows a positive correlation to the prevalence of showing. If feasible, there should always be at least one SMS message sent reminding of appointment.
* If costs of SMS are low then it would seem to be worth the cost of automatically messaging all individuals with an appointment date. The only additional consideration is how many days out the message should be sent.
* If the costs to auto-message are too high then I’d recommend messaging once a interval becomes large enough. Further data exploration should break down the no-show ration in a dataframe grouped by interval and ranked in descending order. Try to see if there’s a pattern where there is a sizeable jump in no-show percentage, at that point I’d make the recommendation that any interval at that point and larger be required to have SMS message.