Predicting Medical Appointment No-shows

Supervised Learning Capstone - Andrew Boho

Can we predict if a patient will skip their scheduled medical appointment?

Motivation:

- First line of defense: catch medical issues before they become medical emergencies
- Chronic illnesses require monitoring for optimal medical outcomes
- Efficiency: an unattended medical appointment wastes time for the medical practitioner, thus increases medical costs and waste

Data

- 110,527 rows containing with 14 columns regarding a scheduled medical appointment in Brazil
- No missing data points and no duplicate records
- Data sourced from Kaggle*, however the ultimate source is not given

*https://www.kaggle.com/joniarroba/noshowappointments/home

Data

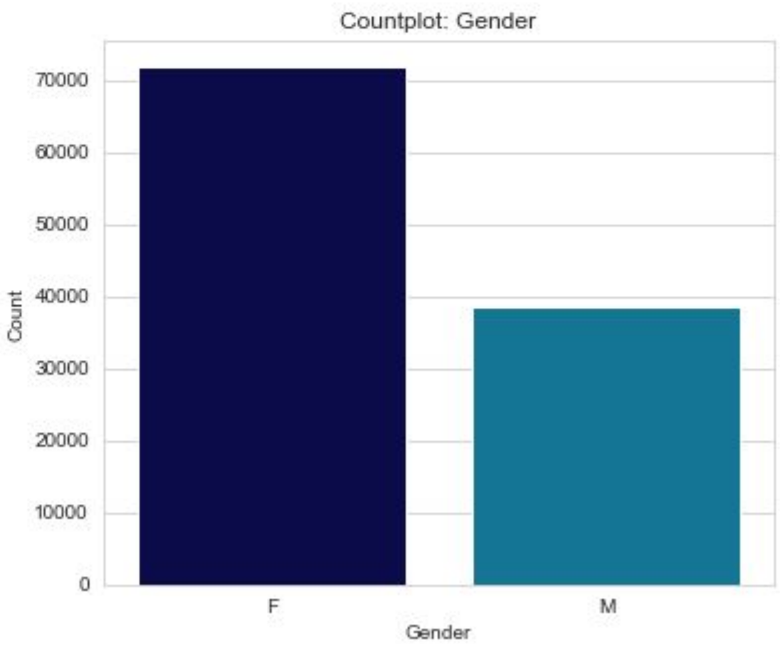
- Columns
    - **patient_id**: Unique identifier for patient
    - **appointment_id**: Unique identifier for appointment
    - **gender**: The patient’s gender
    - **scheduled_day**: The date and time that the appointment was scheduled
    - **appointment_day**: The date of the actual appointment.
    - **age**: The patient’s age
    - **neighborhood**: The neighborhood where the appointment took place
    - **scholarship**: Binary variable indicating if the patient receives medical public assistance
    - **hypertension**: Binary variable indicating if the appointment was for hypertension
    - **diabetes**: Binary variable indicating if the appointment was for diabetes
    - **alcoholism**: Binary variable indicating if the appointment was for alcoholism
    - **handicap**: A variable describing the handicap status of the patient
    - **sms_received**: Binary variable indicating if the patient received a text appointment reminder
    - **no_show**: Binary variable indicating if the patient was a no-show for the appointment

EDA and feature engineering

The two identifier variables dropped (patient_id and appointment_id), leaving 12 attributes (11 categorical and one quantitative)

Name	Unique	DType
gender	2	category
scheduled_day	103549	datetime64[ns]
appointment_day	27	datetime64[ns]
age	104	int64
neighborhood	81	category
scholarship	2	category
hypertension	2	category
diabetes	2	category
alcoholism	2	category
handicap	5	category
sms_received	2	category
no_show	2	category

EDA and feature engineering



EDA and feature engineering

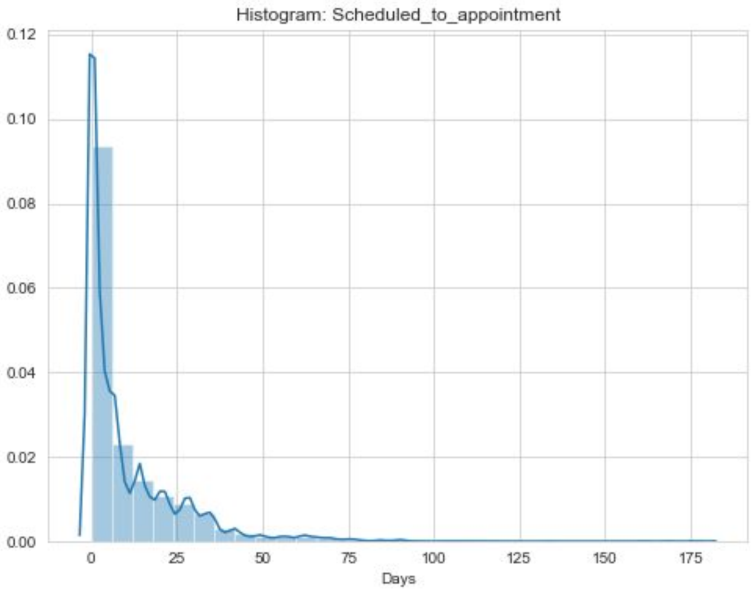
Time variables

```
##### scheduled_day #####
count                      110527
unique                     103549
top           2016-05-06 07:09:54
freq                           24
first         2015-11-10 07:13:56
last          2016-06-08 20:07:23
Name: scheduled_day, dtype: object

##### appointment_day #####
count                      110527
unique                         27
top           2016-06-06 00:00:00
freq                         4692
first         2016-04-29 00:00:00
last          2016-06-08 00:00:00
Name: appointment_day, dtype: object
```

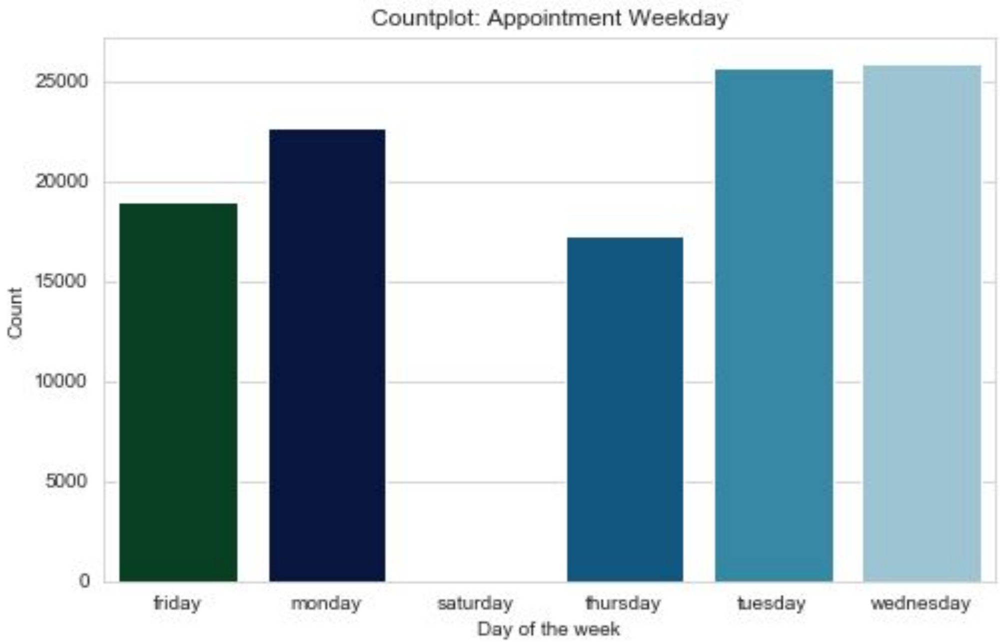
EDA and feature engineering

Calculated the time in days between date scheduled to the actual appointment

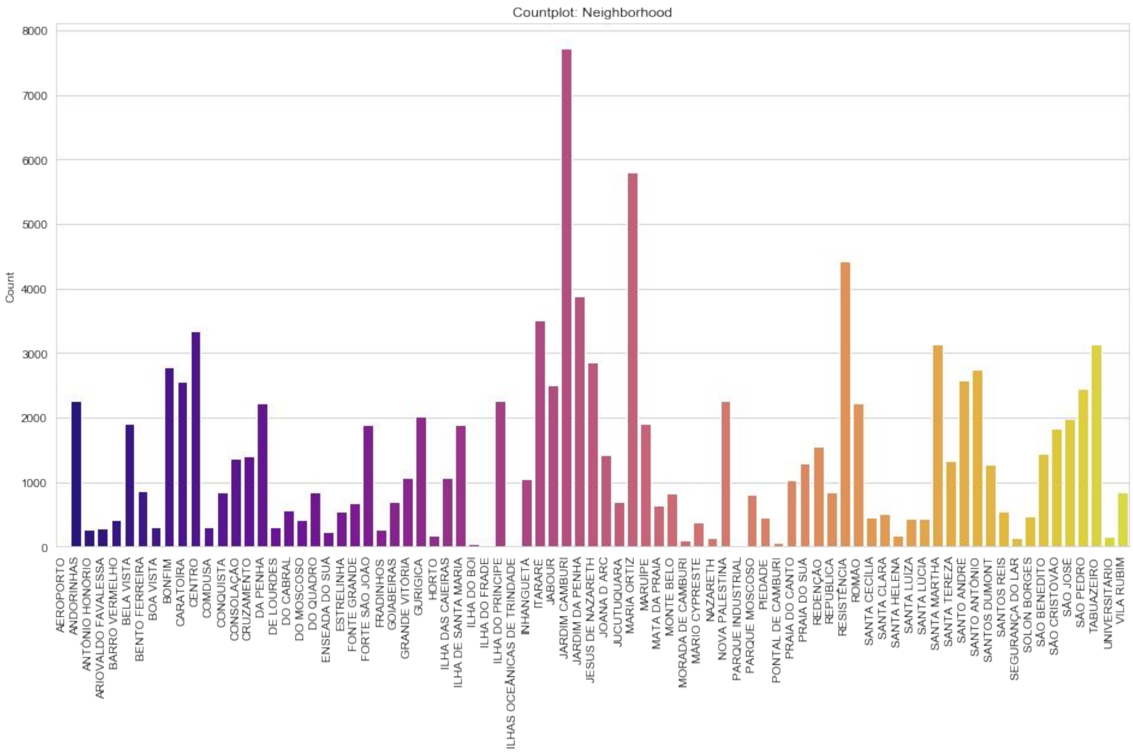


EDA and feature engineering

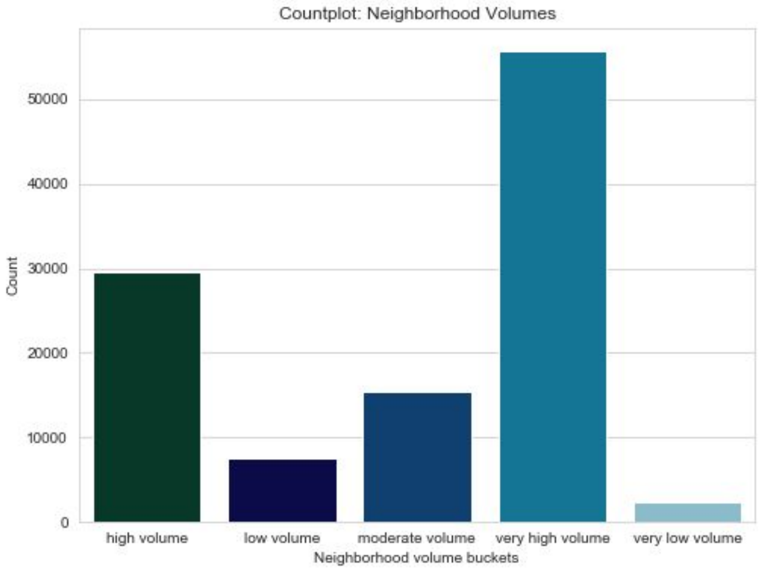
Added feature for the day of the week



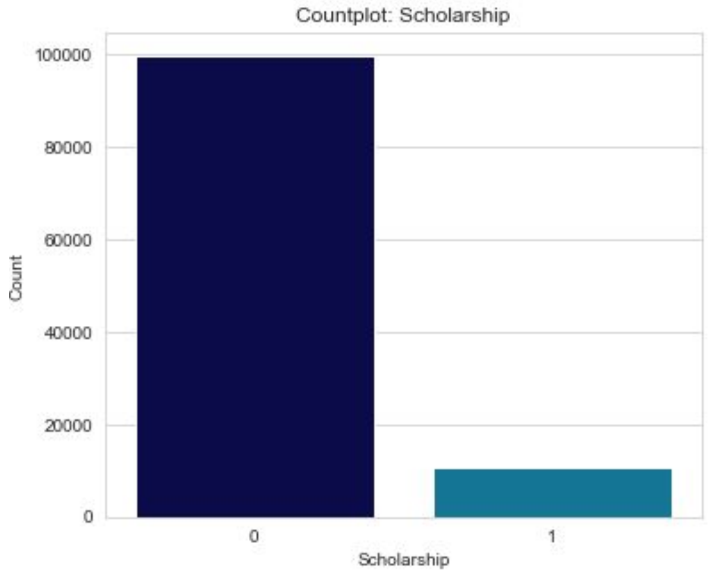
EDA and feature engineering



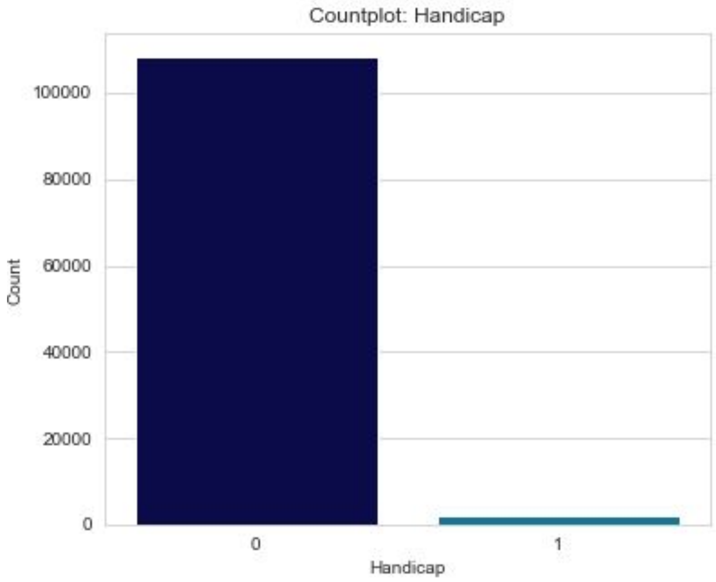
EDA and feature engineering



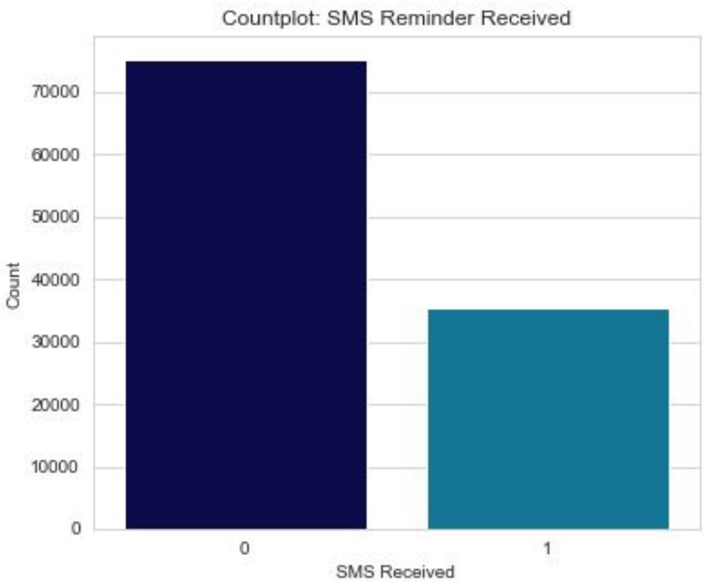
EDA and feature engineering



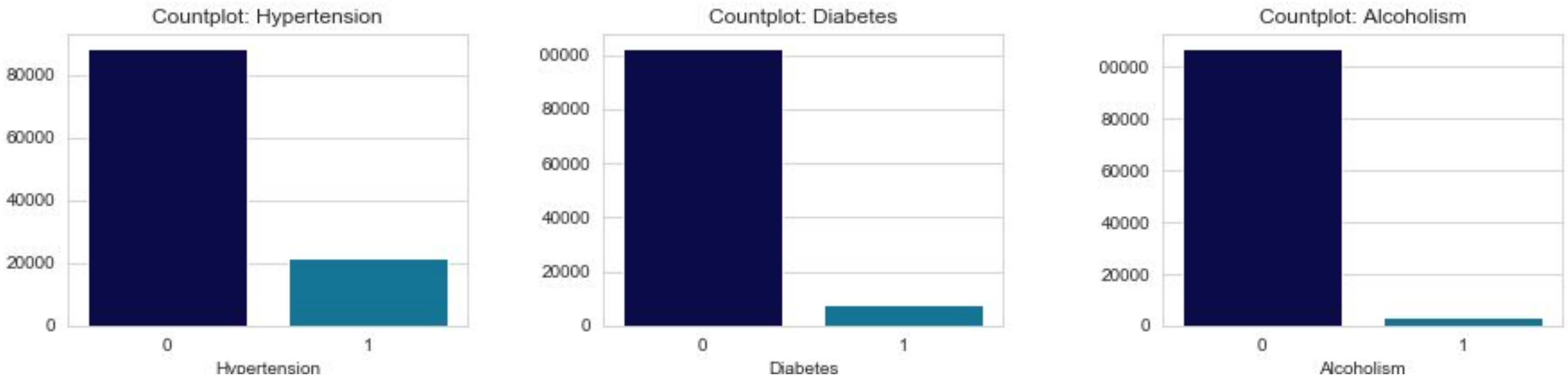
EDA and feature engineering



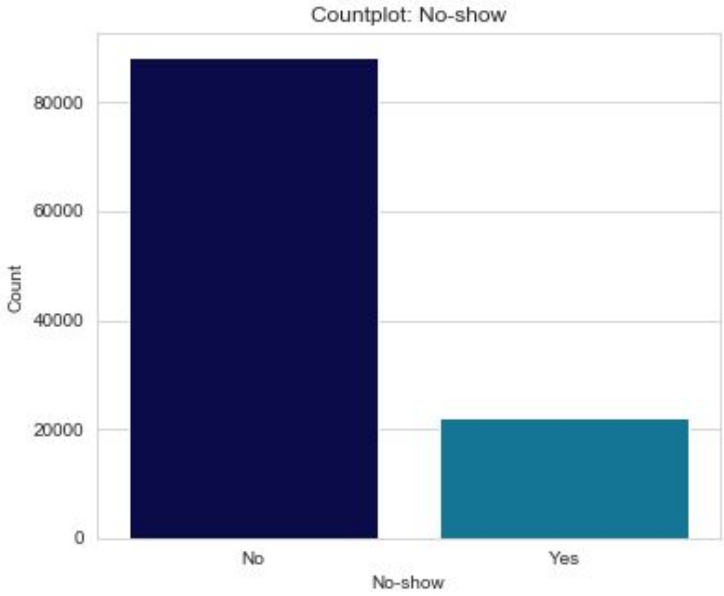
EDA and feature engineering



EDA and feature engineering

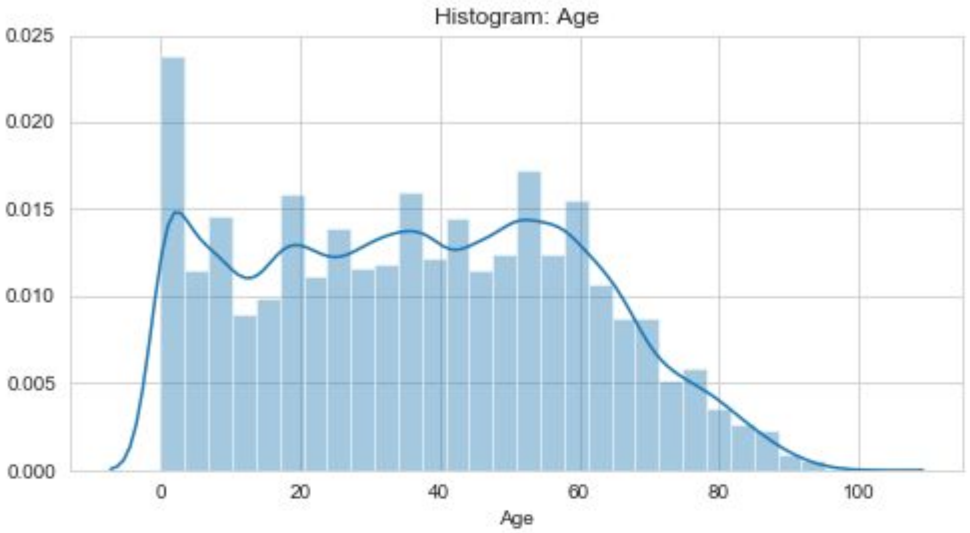


EDA and feature engineering

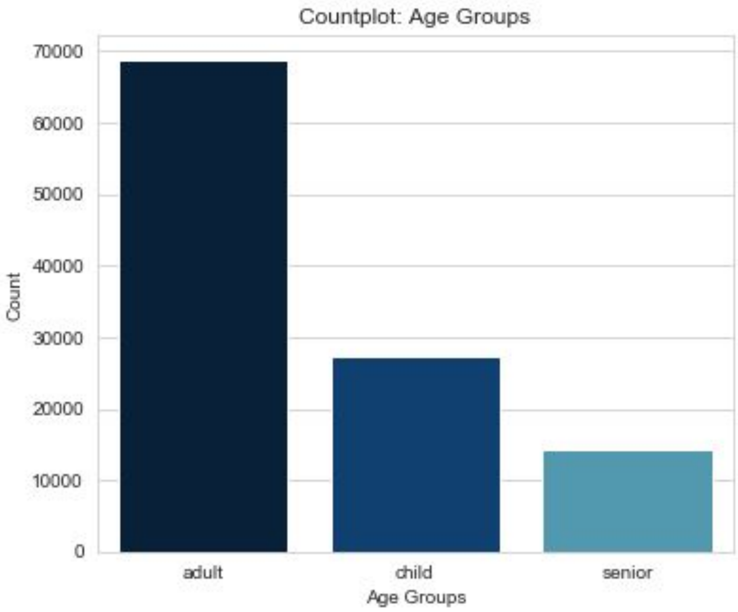



```
No      0.798104
Yes     0.201896
Name: no_show, dtype: float64
```

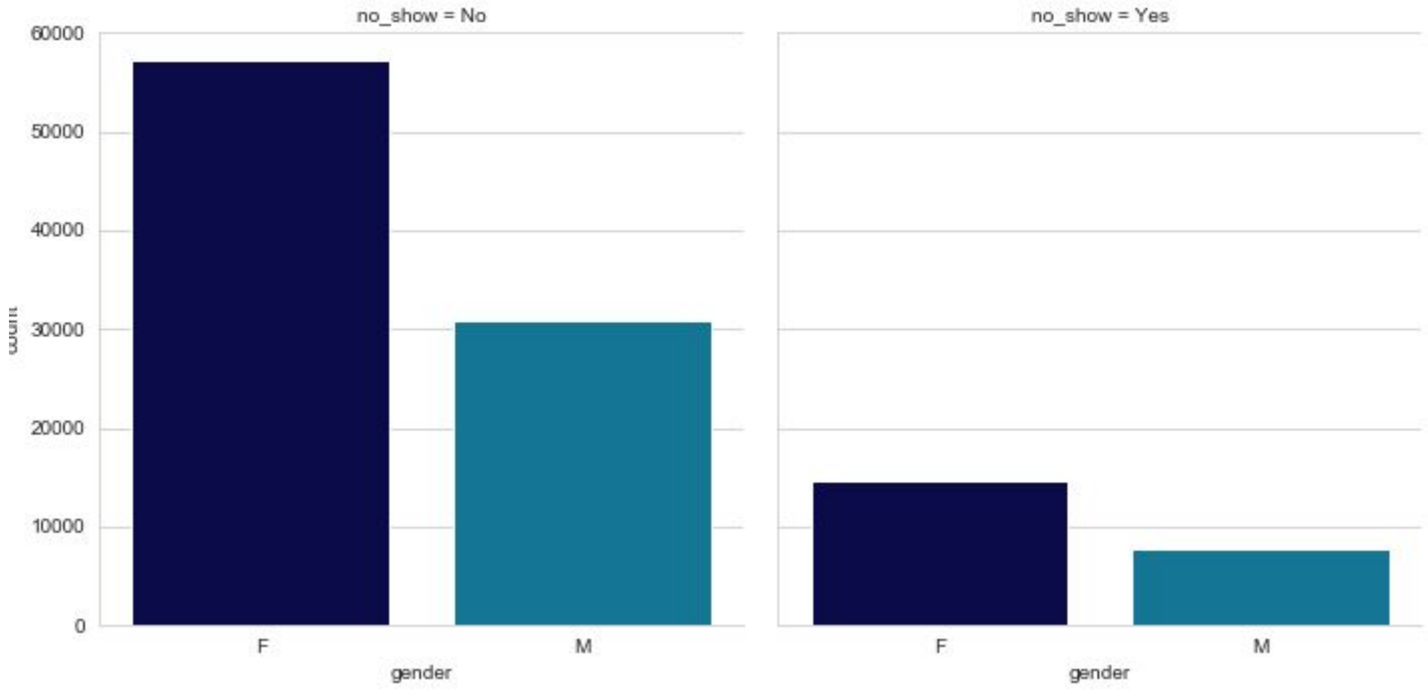
EDA and feature engineering



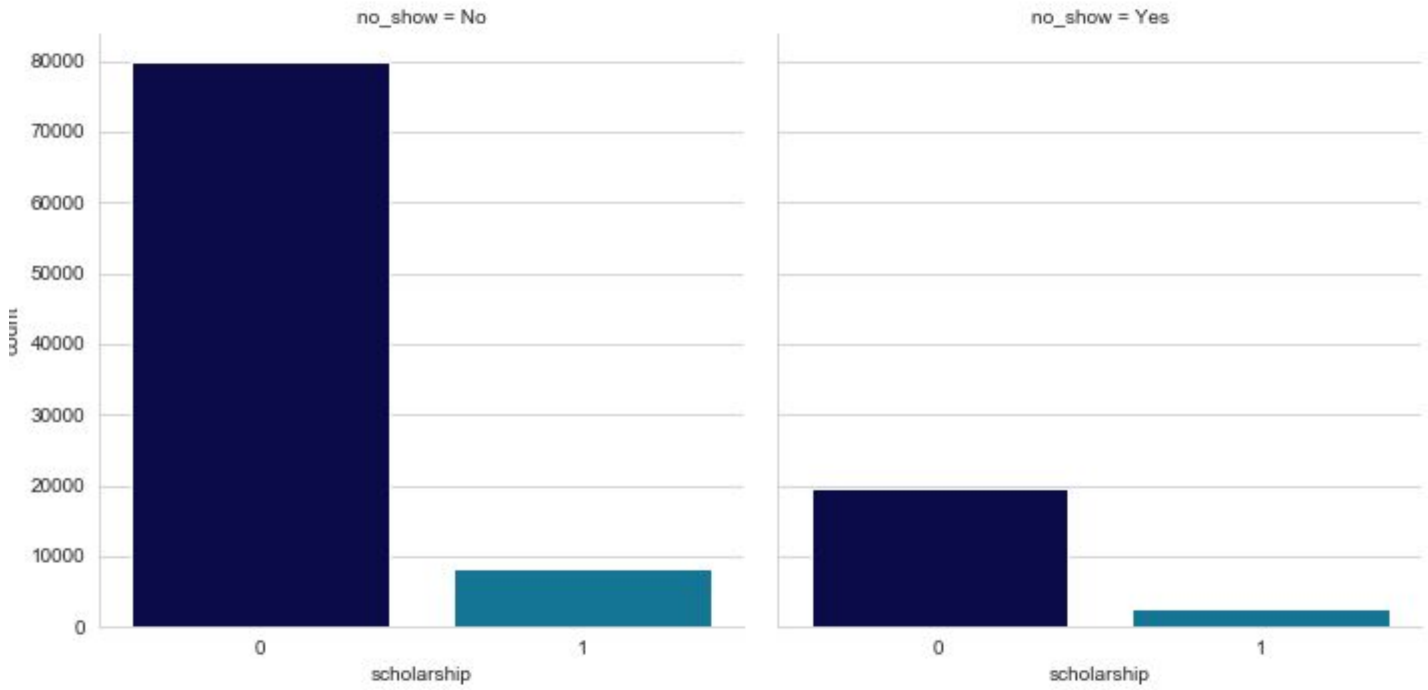
EDA and feature engineering



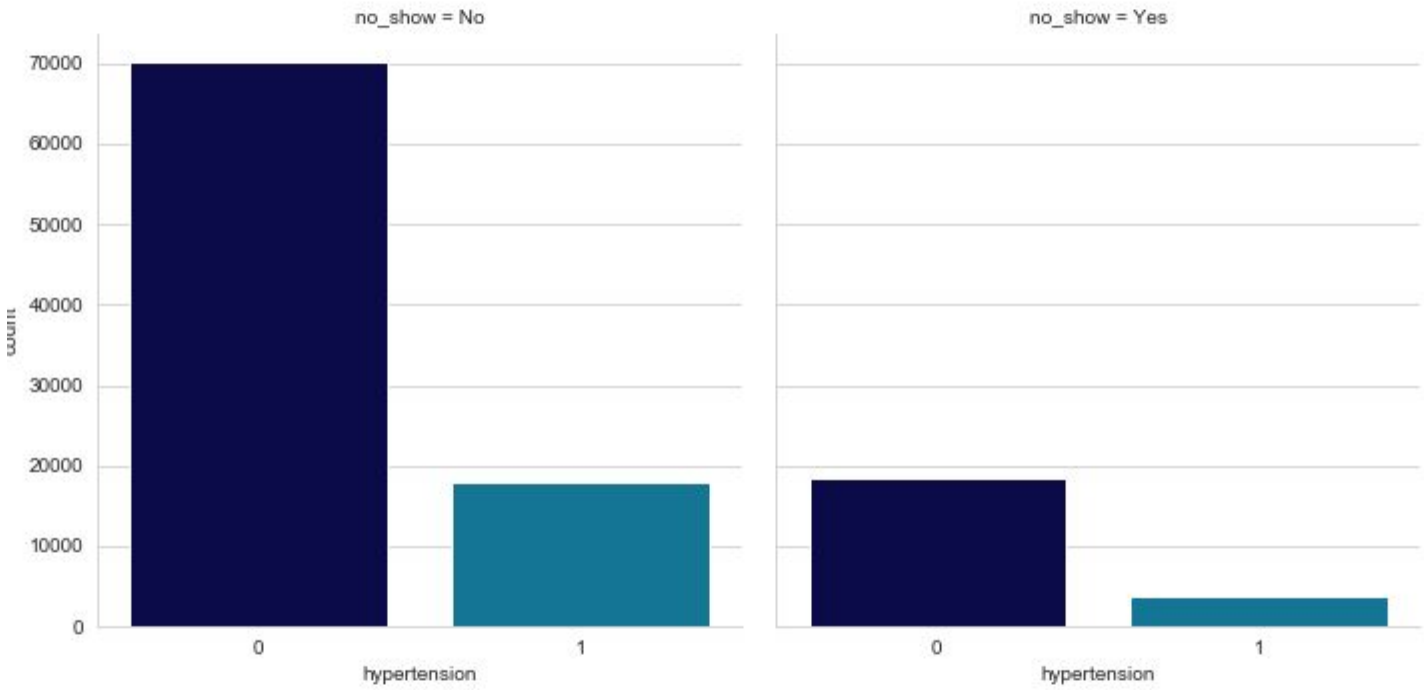
EDA and feature engineering



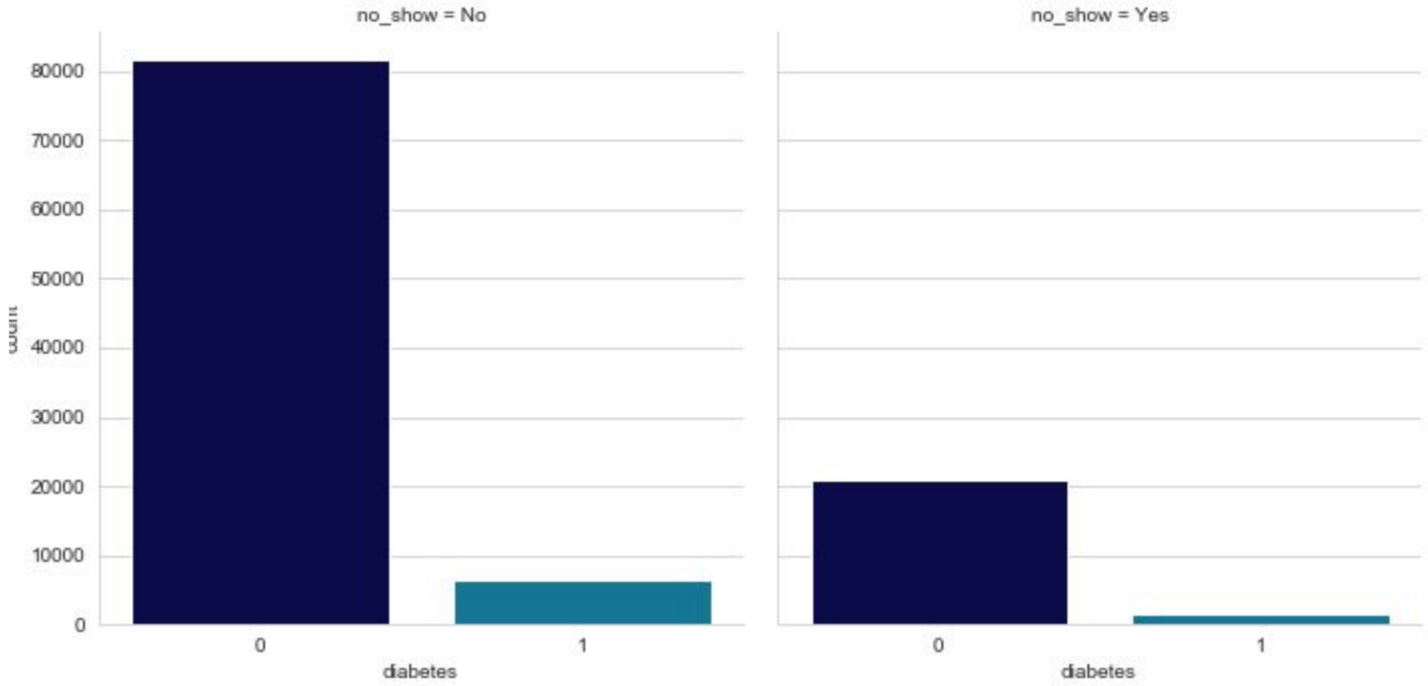
EDA and feature engineering



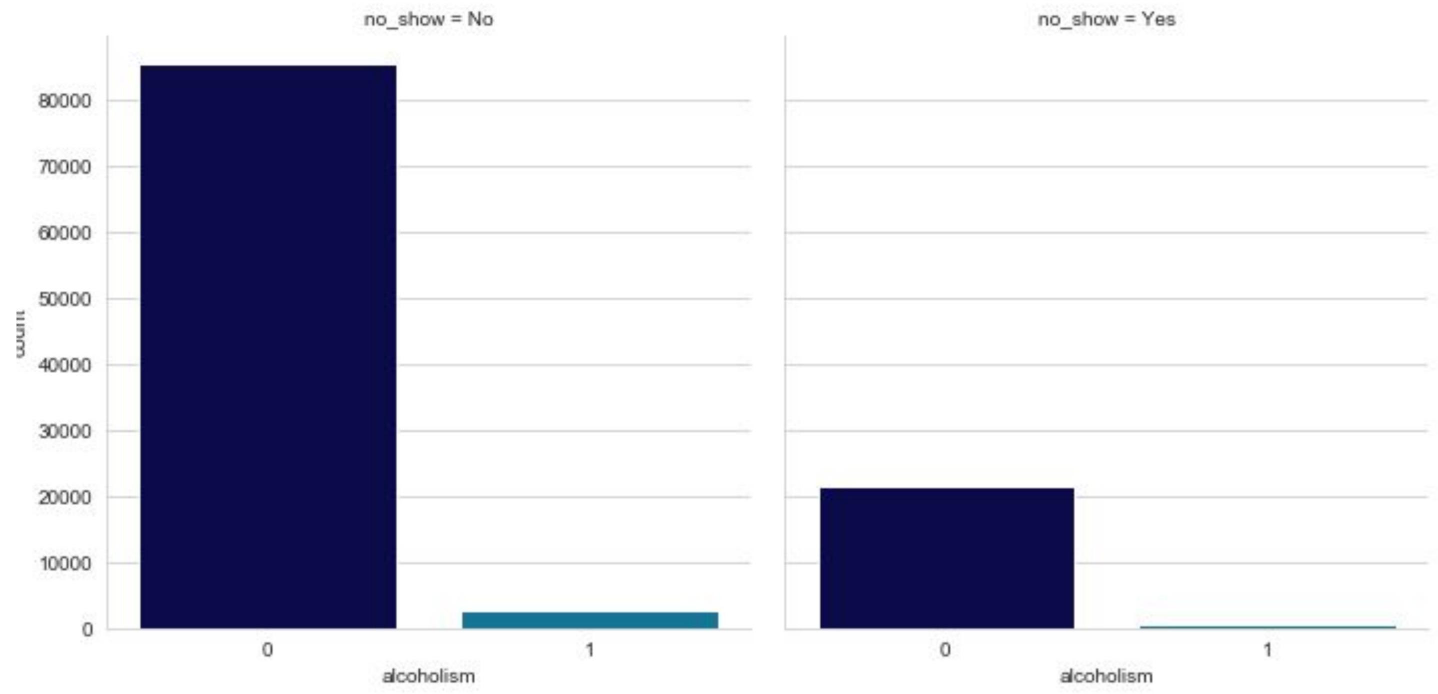
EDA and feature engineering



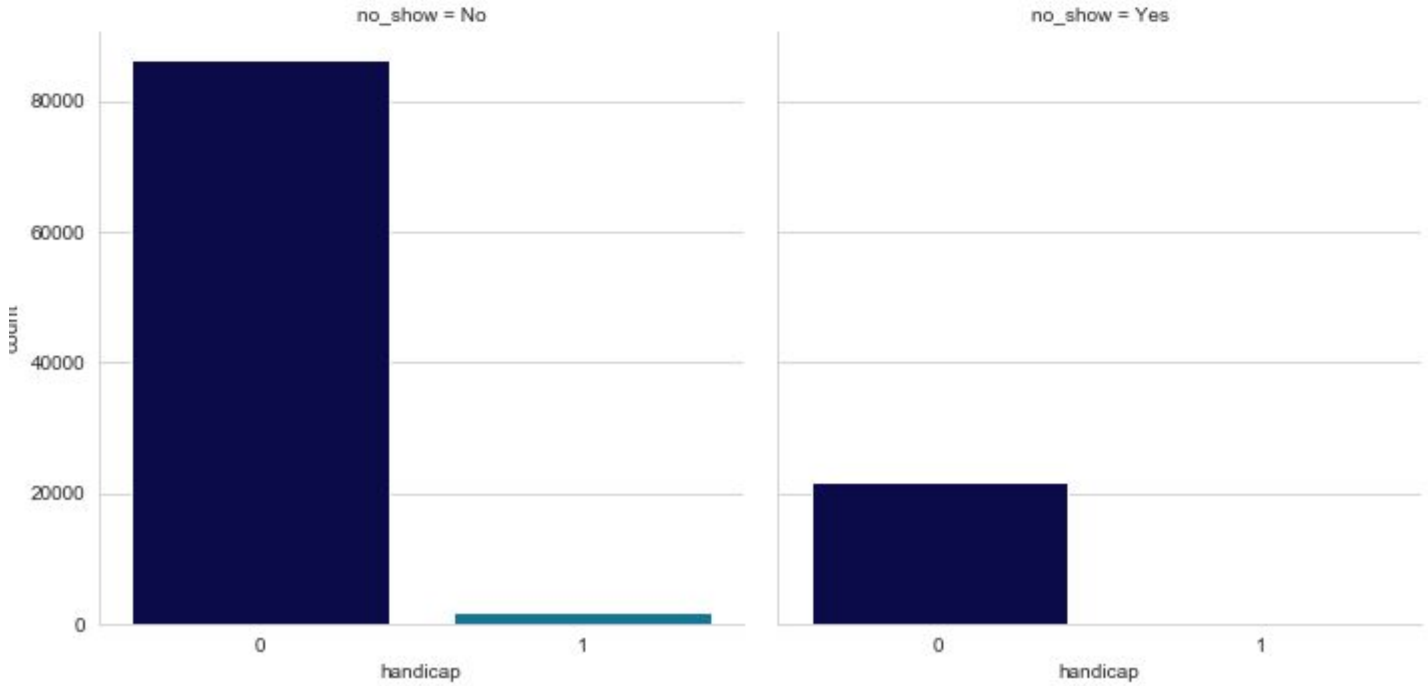
EDA and feature engineering



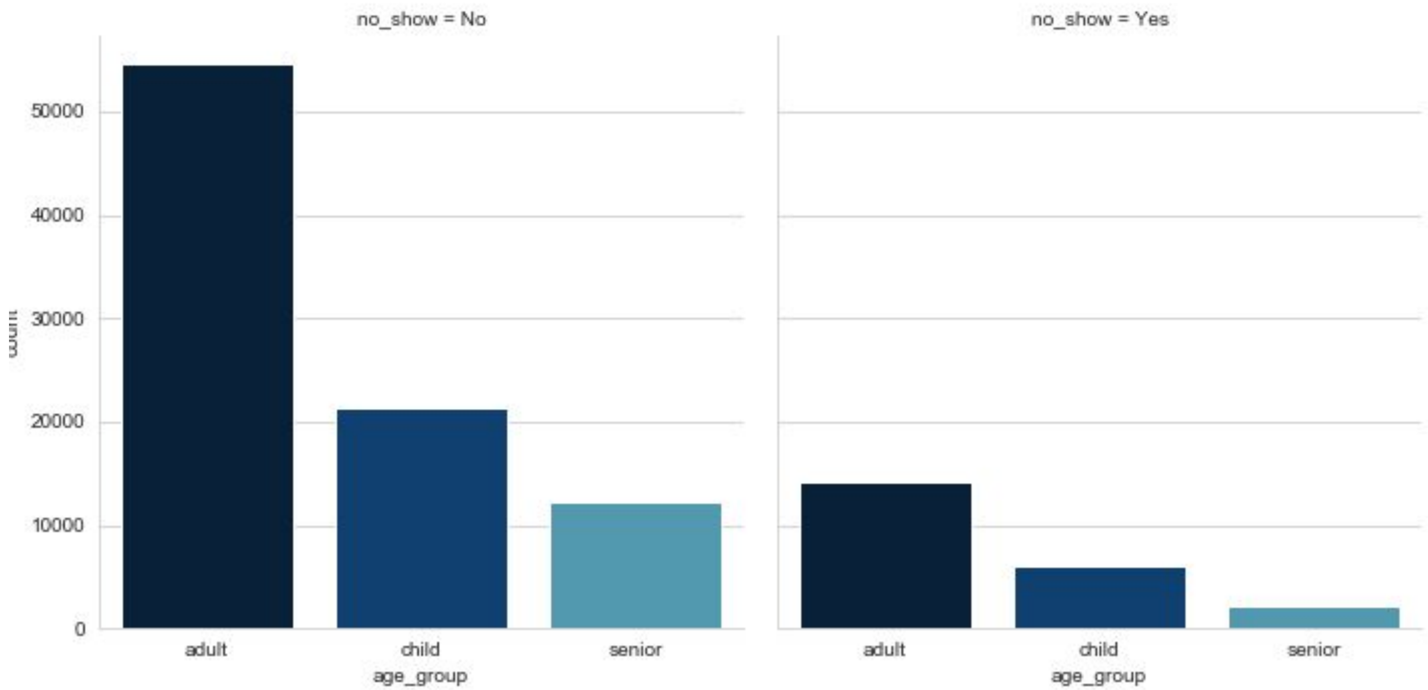
EDA and feature engineering



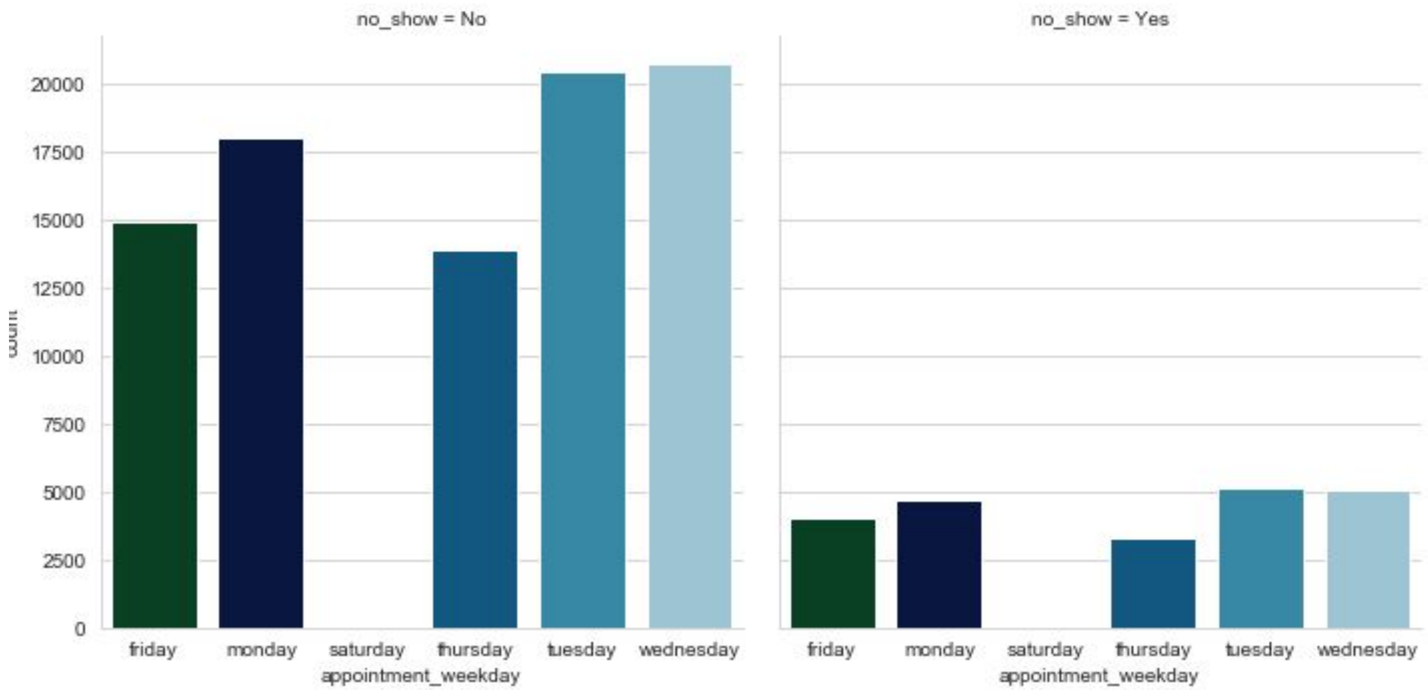
EDA and feature engineering



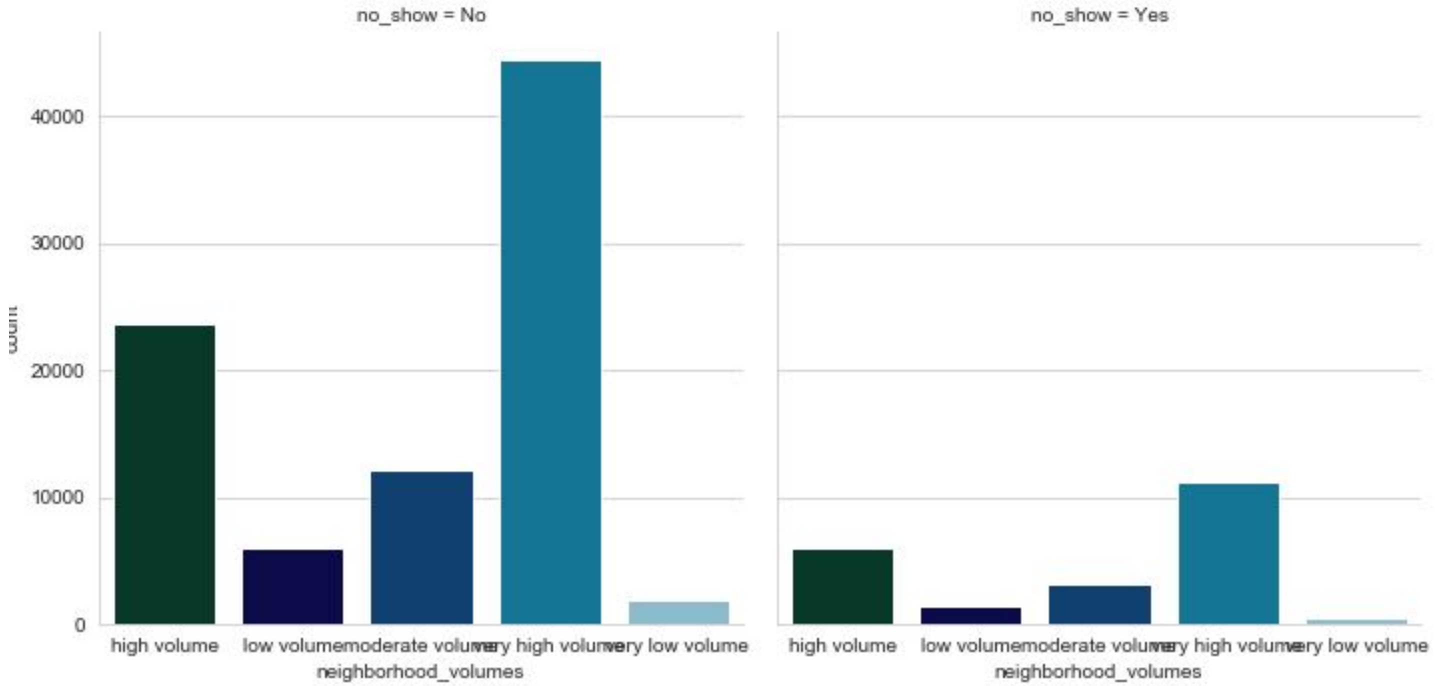
EDA and feature engineering



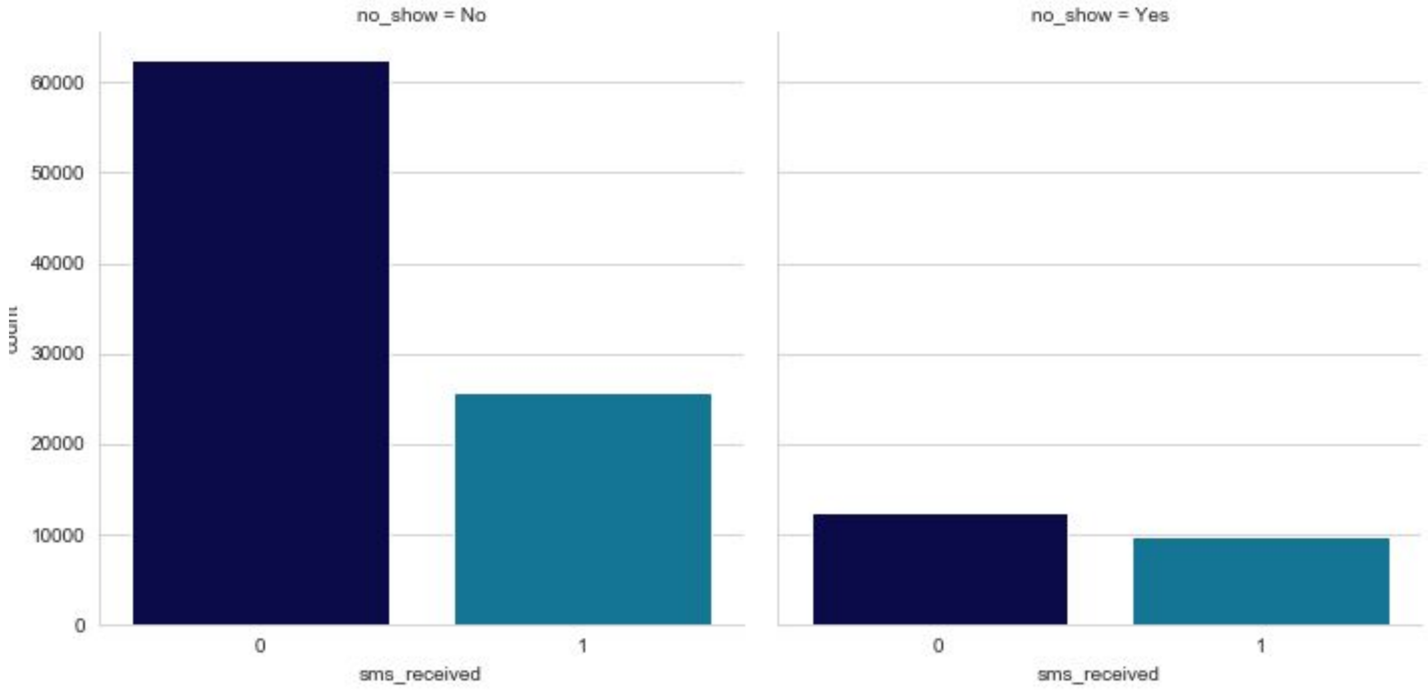
EDA and feature engineering



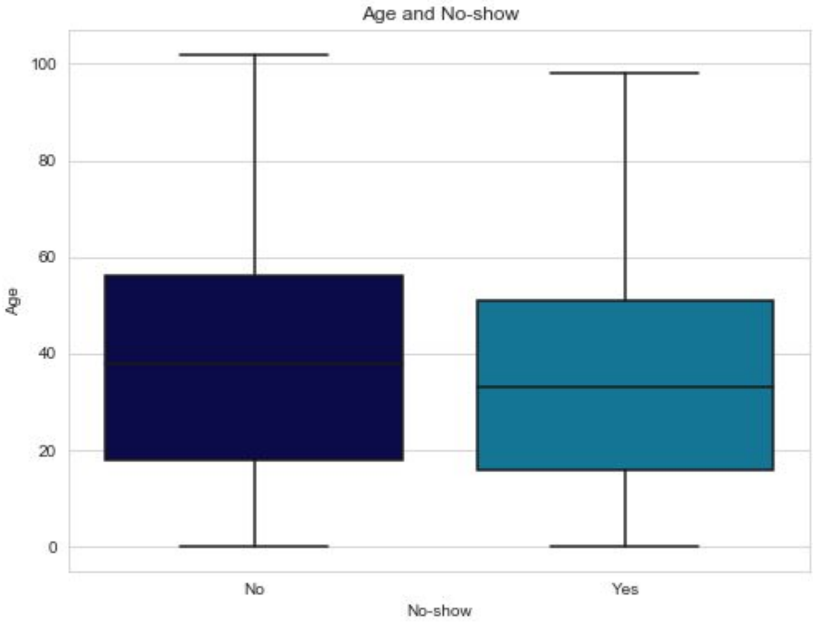
EDA and feature engineering



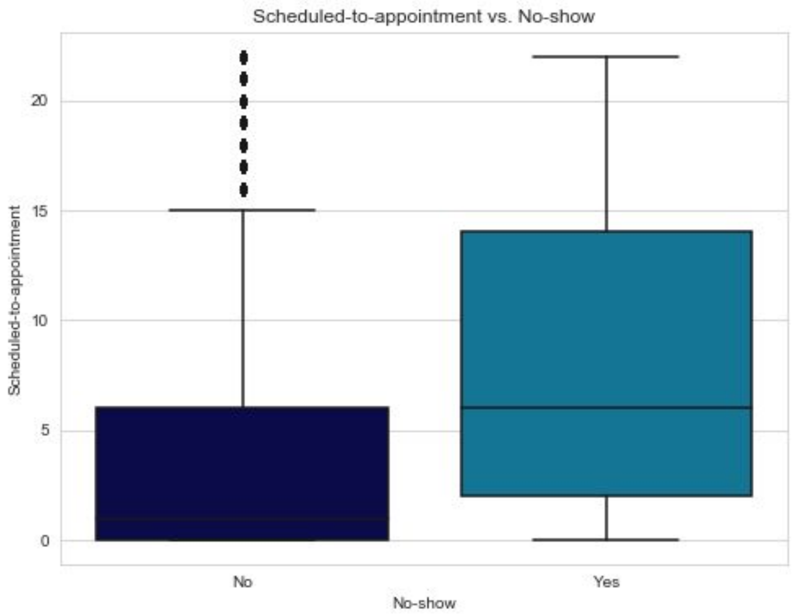
EDA and feature engineering



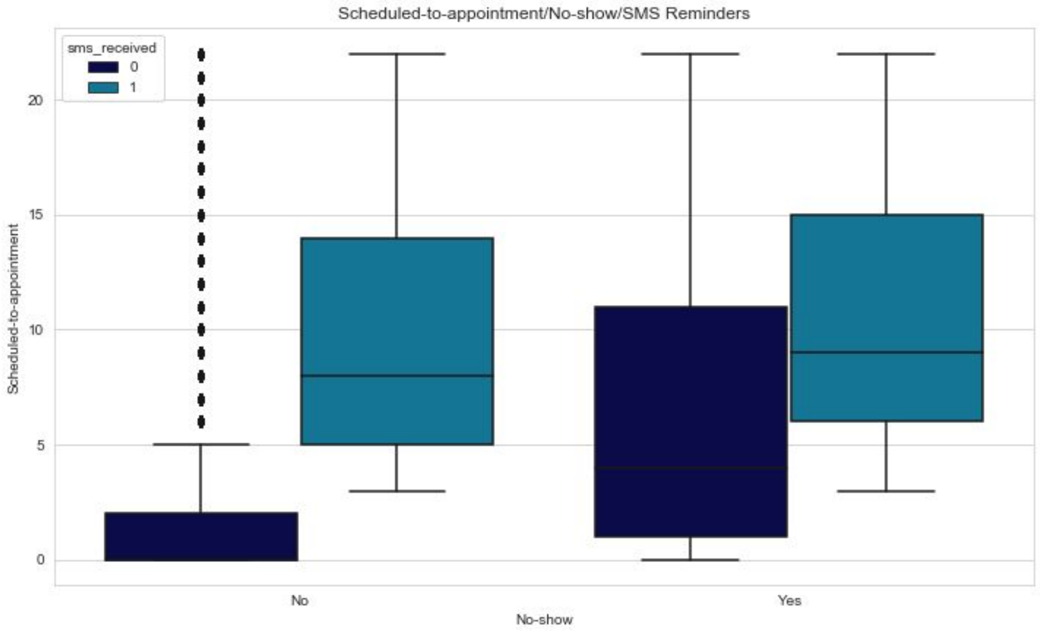
EDA and feature engineering



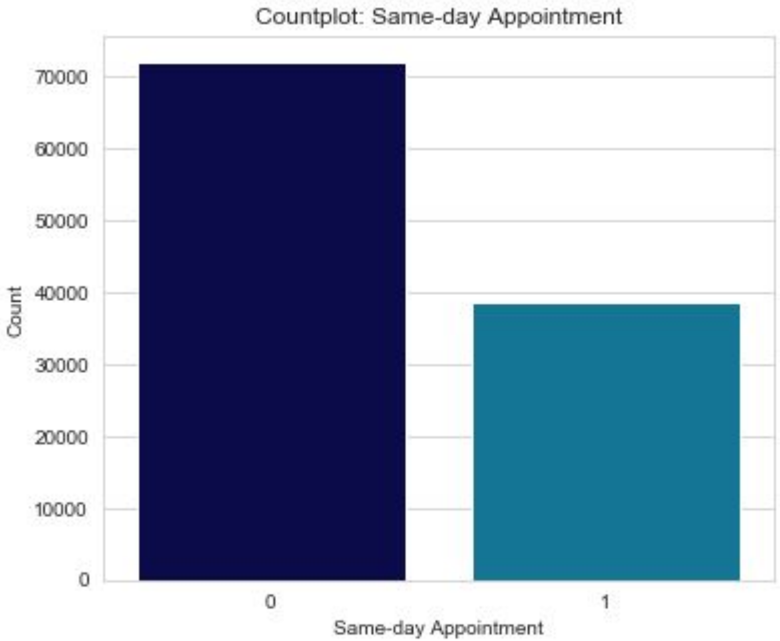
EDA and feature engineering



EDA and feature engineering



EDA and feature engineering




```
Frequency of No-shows for Same-day-appointments
No     0.953528
Yes    0.046472
Name: no_show, dtype: float64
```

Feature selection and sampling

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 110516 entries, 0 to 110526
Data columns (total 20 columns):
scheduled_to_appointment                  110516 non-null int64
gender_M                                  110516 non-null uint8
age_group_child                           110516 non-null uint8
age_group_senior                          110516 non-null uint8
scholarship_1                             110516 non-null uint8
hypertension_1                            110516 non-null uint8
diabetes_1                                110516 non-null uint8
alcoholism_1                              110516 non-null uint8
handicap_1                                110516 non-null uint8
sms_received_1                            110516 non-null uint8
neighborhood_volumes_low volume           110516 non-null uint8
neighborhood_volumes_moderate volume      110516 non-null uint8
neighborhood_volumes_very high volume     110516 non-null uint8
neighborhood_volumes_very low volume      110516 non-null uint8
appointment_weekday_monday                110516 non-null uint8
appointment_weekday_saturday              110516 non-null uint8
appointment_weekday_thursday              110516 non-null uint8
appointment_weekday_tuesday               110516 non-null uint8
appointment_weekday_wednesday             110516 non-null uint8
same_day_appointment_1                    110516 non-null uint8
dtypes: int64(1), uint8(19)
memory usage: 8.7 MB
```

Feature selection and sampling

Sampling

- High class imbalance for the target variable (no_show)
- 79.8% of patients represented in the data show up for their appointment
- Data was split into a training-set and a test-set (80% / 20%)
- Three sampling techniques used to correct for the imbalance in the training-set*
  - Up-sample
  - Down-sample
  - SMOTE: Synthetic Minority Over-sampling

*All models where run on un-sampled data as well to get a baseline. Thus, each model was run on four sample-sets

Performance metrics

Since the goal of this research is to identify patients who are most likely to miss medical appointments with an eye towards ultimately reducing this behaviour, it is imperative that any predictive model be able to correctly identify true positives.

- Models initially ranked by precision
- Other metrics tracked
  - Accuracy!
  - Recall
  - F1-Score
  - Model runtime
- Balance!

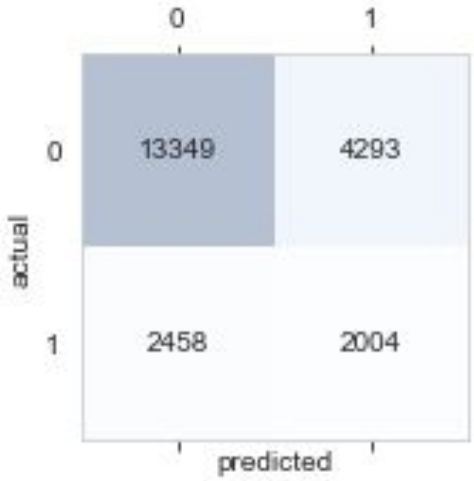
Models

- BernoulliNB
- KNeighborsClassifier
- DecisionTreeClassifier
- RandomForestClassifier
- LogisticRegression
- RidgeClassifier
- LassoClassifier
- SVC
- GradientBoostingClassifier
- XGBClassifier

Best model (but not really a winner)

KNeighborsClassifier

```
###### Results ######
Model: KNeighborsClassifier
Sampling: smote
Mean Training Accuracy: 0.727
Std Training Accuracy: 0.040
Test Accuracy: 0.695
Recall: 0.449
Precision: 0.318
F1 Score: 0.373
Runtime: 162.54548
```



Can we predict if a patient will skip their scheduled medical appointment?

No

Can we predict if a patient will skip their scheduled medical appointment?

Okay, maybe not with this dataset. Ideas for future research:

- Track better attributes
  - Time of day for the appointment
  - More details on the location of the clinic
  - Travel distance between patient and clinic
  - More diagnostic categories
    - 23% of the current dataset doesn’t fall into any of the diagnostic buckets (hypertension, diabetes, and alcoholism)
    - More details regarding the criticality (emergency vs. routine check-up) of the appointment may boost the signal
  - Track over a longer time period

Can we predict if a patient will skip their scheduled medical appointment?

Ideas for future research continued:

- Run an experiment!
  - A/B testing on different types of reminders (i.e. SMS text vs. in-person phone call)

Thank you!