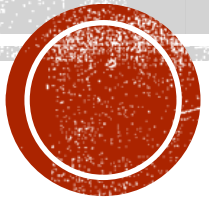
NOSHOW PREDICTION PROJECT

Nouf Almutairi



NOSHOW PREDICTION PROJECT

- Hospital appointment NoShow is a comment issue that cause issues to healthcare organization.

- The predication of the patient who will not attend their appointment or the number of the no show would help the organization in the decision making, plan and to reduce the no show rate.

- The goal of this project is to use classification models for Noshow predication in order to help in this issue.

DATA DESCRIPTION

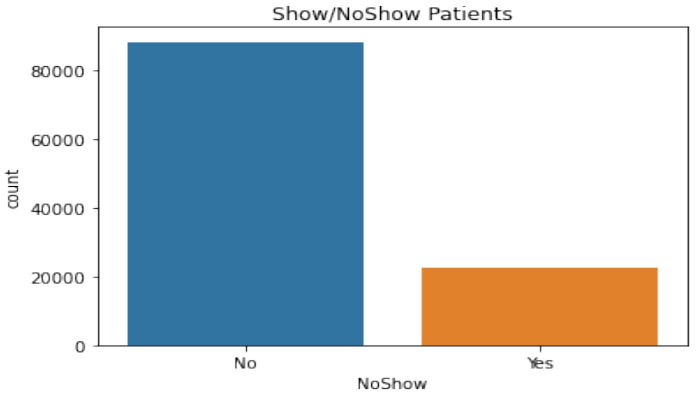
- Patient appointment data with the status of the patient as Show or Noshow
- The source of the data is from kaggle
- The data consist of 110,527 records with 13 features.
- One label feature.

DATA PREPROCESSING

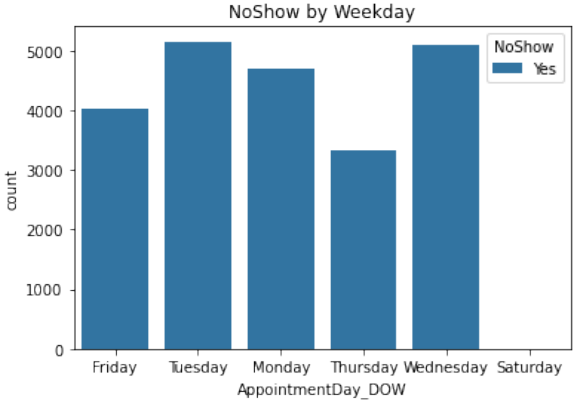
- Convert date columns "ScheduledDay" and "AppointmentDay" form float data type to date date type.
- Remove Age value = -1
- Adding Waiting time and Appointment Week day to the data
- Waiting time contains minus data, this is happening when ScheduledDay grater than t AppointmentDay. As data cleansing for these cases the ScheduledDay replaced by the AppointmentDay and vice versa. Assuming this is data entry mistake.
- Remove Patient ID and Appointment ID since they are a database column and no meaning to have them in the data
- Remove date columns: "ScheduledDay" and "AppointmentDay" after convert them to Day, month and year

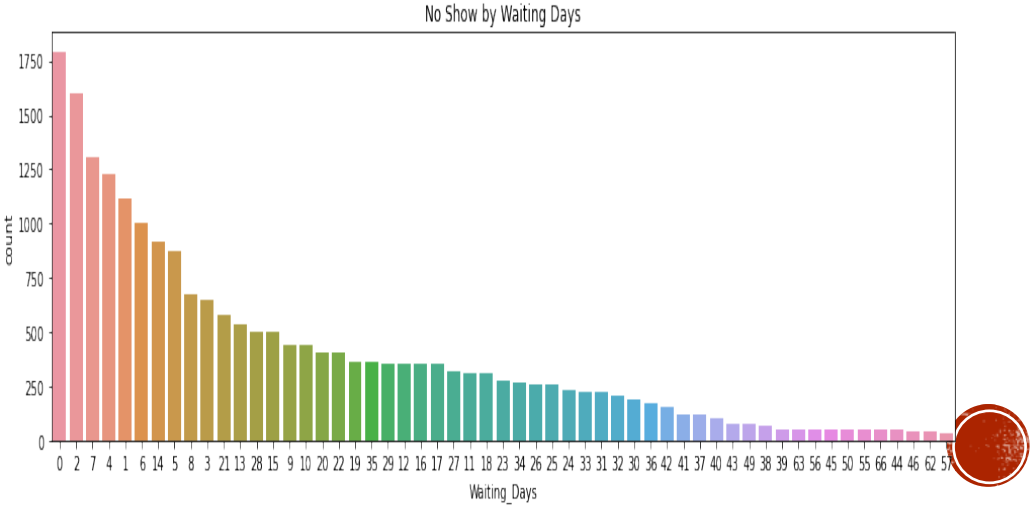
EDA

- A graph for total Show & No Show cases

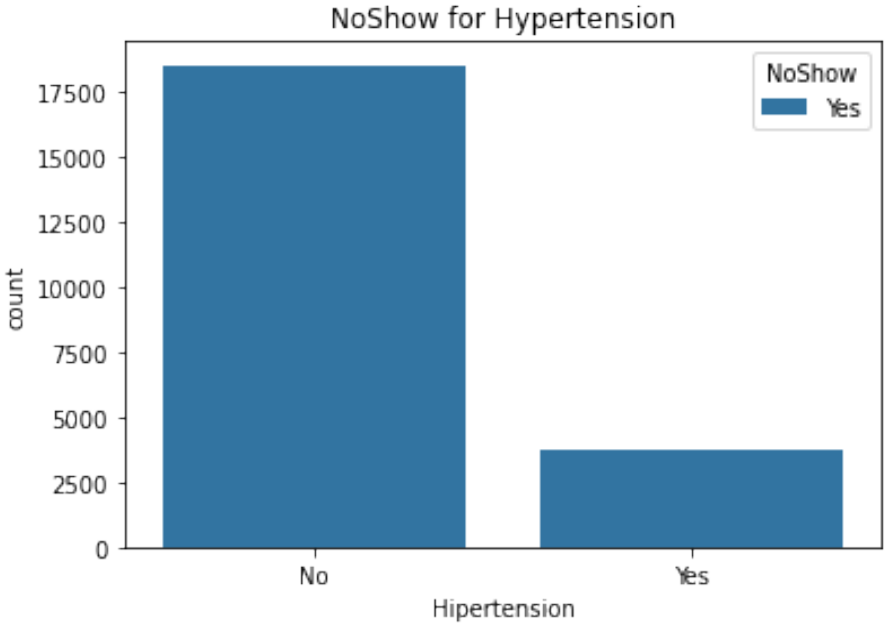


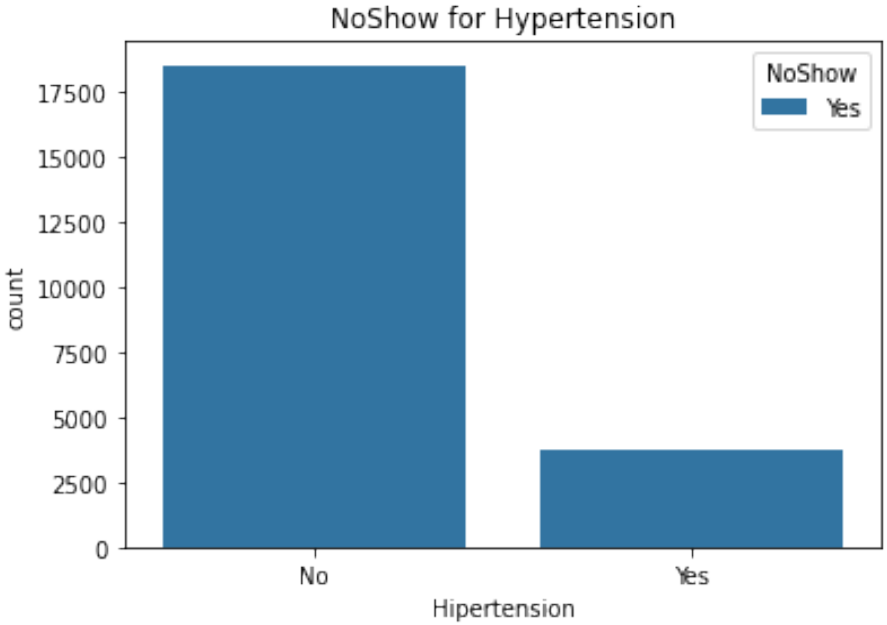
- A graph for No show cases by Diabetes, hypertension, day of the week and waiting time to find the relation between NoShow and these features.





EDA


NoShow for Hypertension
NoShow
Yes
17500
15000
12500
10000
7500
5000
2500
0
count
No
Yes
Hipertension

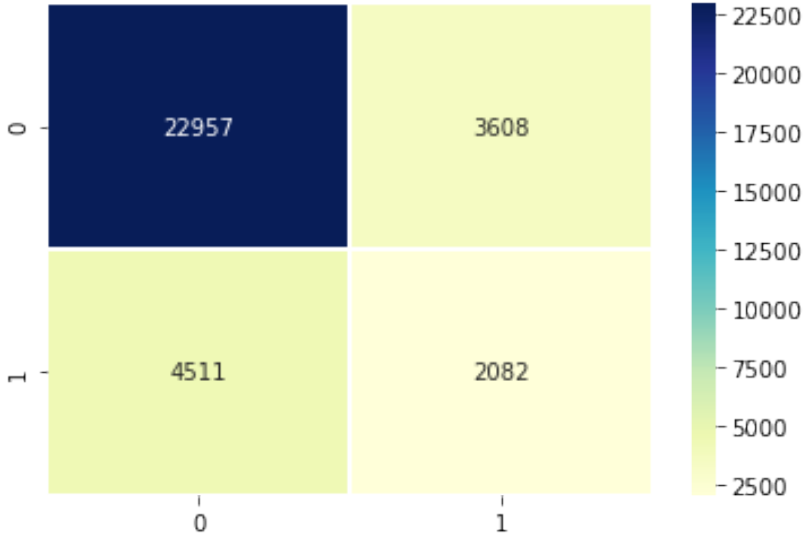

NoShow for Hypertension
NoShow
Yes
17500
15000
12500
10000
7500
5000
2500
0
count
No
Yes
Hipertension

MODELING

- Data is imbalance and oversampling technique has been applied
- Random forest , Decision tree and Logistic regression were used.

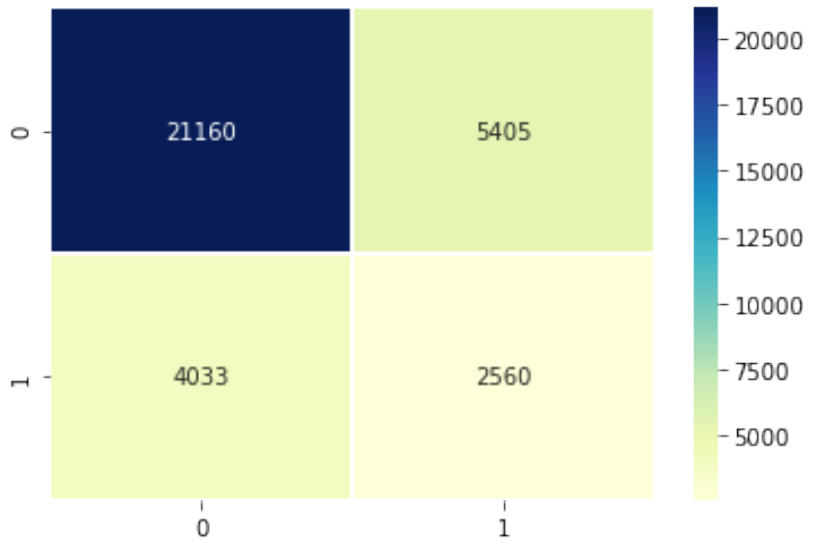
MODELING

- Random forest classifiers
  - **F1 Score:** 0.339
  - **Test Precision:** 0.366
  - **Test Recall:** 0.316
  - **Confusion_matrix:**



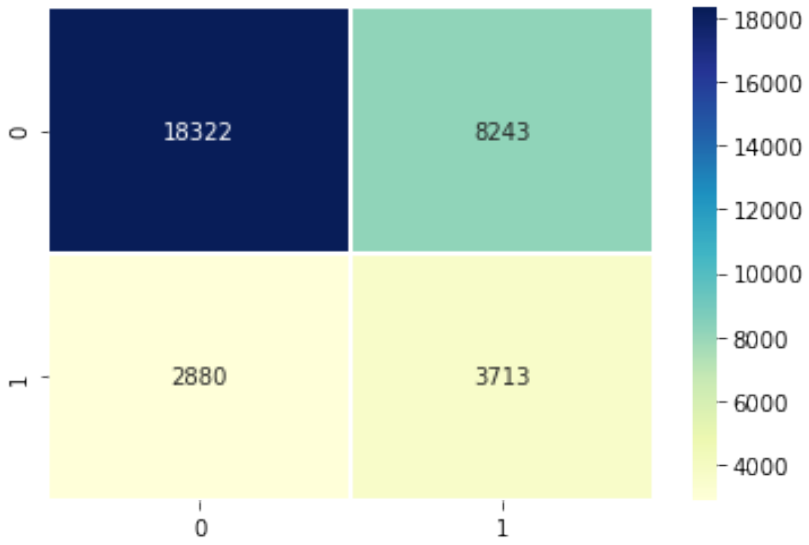
MODELING

- Decision tree
  - **F1 Score:** 0.352
  - **Test Precision:** 0.321
  - **Test Recall:** 0.389
  - **Confusion_matrix:**



MODELING

- Logistic regression
  - **F1 Score:** 0.400
  - **Test Precision:** 0.310
  - **Test Recall:** 0.563
  - **Confusion_matrix:**



CONCLUSION

- Logistic regression was best performance than other techniques.
- The same process will applied on real data.
- This project will be utilized to help in decision making.

Thank you