Capstone Project 2 Report

Predicting the Recurrence of Thyroid Cancer for Post-Treatment Patients

Alex Lai

Problem:

- The company I am working for has developed a treatment for thyroid cancer. However, some of those patients have had their thyroid cancer come back.
- I am tasked with making a prototype eligibility test AI, a predictive model that will input patient data and see if they are a right fit for this treatment and say if they have a high chance of success where post-treatment their thyroid cancer will not return.
- We can do this by using existing patient data that already took the treatment so the model can identify what factors led to cancer recurrence post-treatment.
- The stakeholders is the company developing the drugs, the doctors, and the patients.
- The model accuracy will need a minimum of 95% accuracy.

Data Wrangling

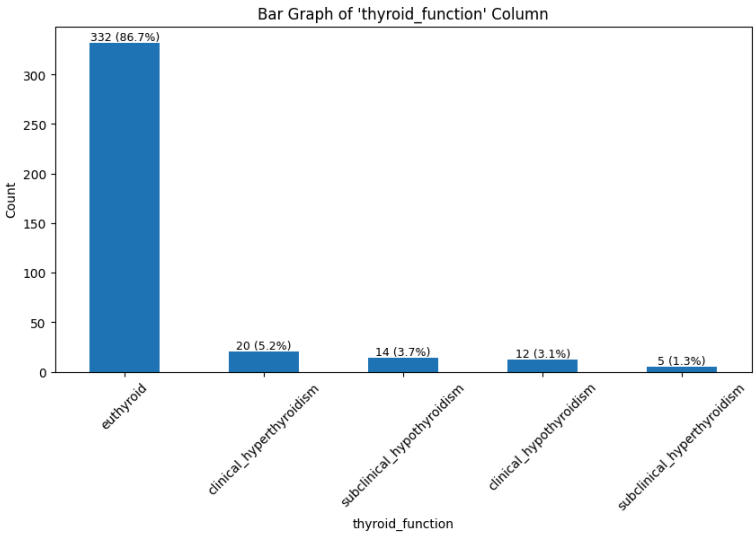
Each row represent a patient that has some form of Thyroid Cancer. We have 383 rows / patient data. This data is practically all categorical. As you can see up here, these are all the columns. I standardized the columns to make them all lowercase and replace spaces with an underscore. Also fixed the typo in hx_radiotherapy, and renamed the ‘recurred’ column to recurrence to make it more standardized for other medical professional terms as trying to remove as much ambiguity as possible. The data is already quite clean beforehand so I have to do minimal to no cleanup of missing values, and duplicate values, and rename misspelled values.

Then I quickly merge specific data together into new columns that EDA may want to use. Age is our only numeric column but to make it categorical with the rest I bin it into age groups. The other categories I merged had values that were somewhat duplicate and think maybe combining them could create better variance info for the model, then added those as new columns.

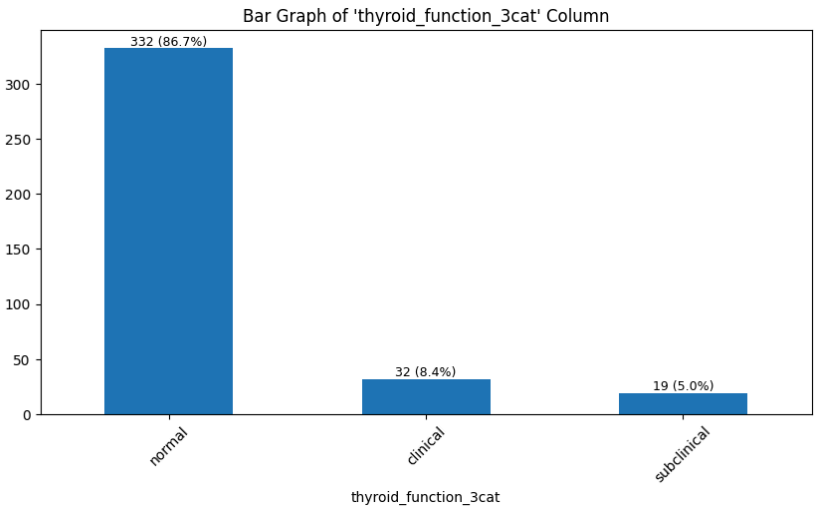
This is an example of how we merge the categories in a column into new columns which EDA and feature engineering can decide which to use. Of course, feature engineering will make sure to select the column they want and prevent duplicates.

Example of Column Merging in data wrangling:

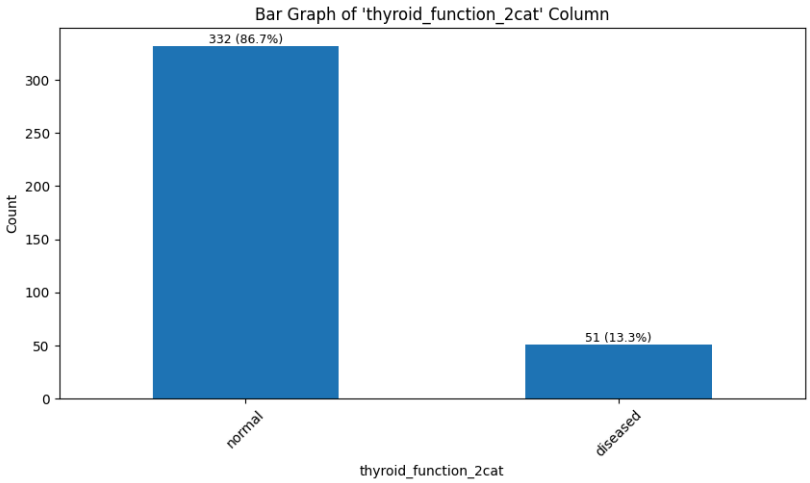
Original Column



Merge the clinical and subclinical columns



Merge the all the columns with thyroid issue into one ‘diseased’ column.

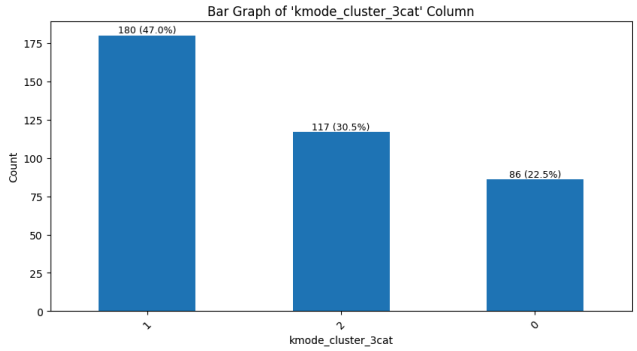


So back to our problem identification, we are trying to make a model that can input new patient data and predict if thyroid cancer come back will occur after treatment. Thus the column we would be focusing on is ‘recurrence’.

Exploratory Data Analysis (EDA)

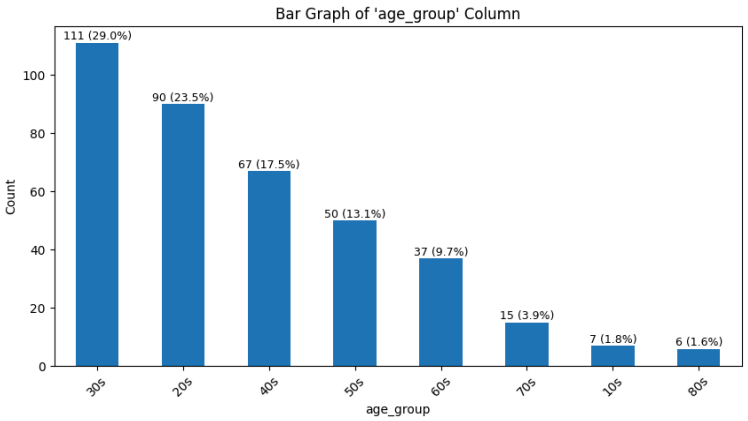
K Mode Clustering

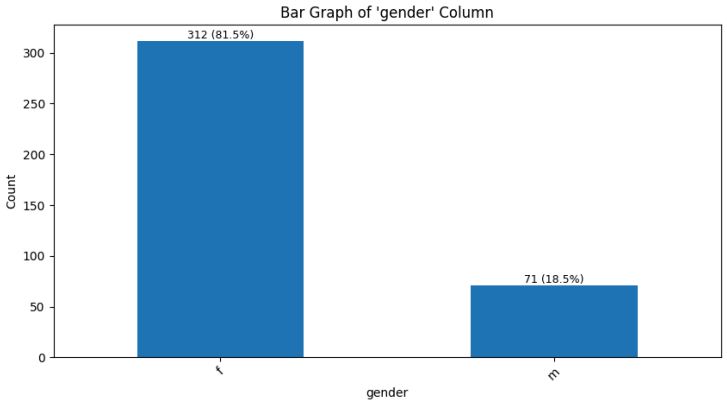
We also perform K mode clustering to see if there are specific patterns of groups, that may hold promise so we included as a new column.

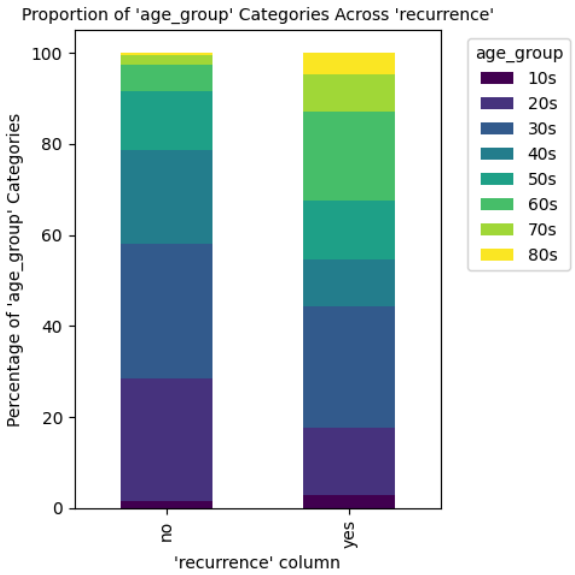


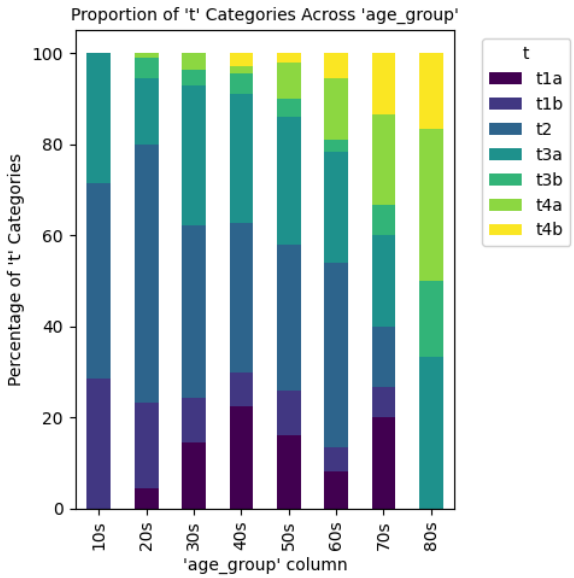
Explore the data count and stacked bar graphs

I have made bar graph and stacked bar graph for each column but will show a few that are interesting. One of the main things that I have noticed is that some of these data are imbalanced.



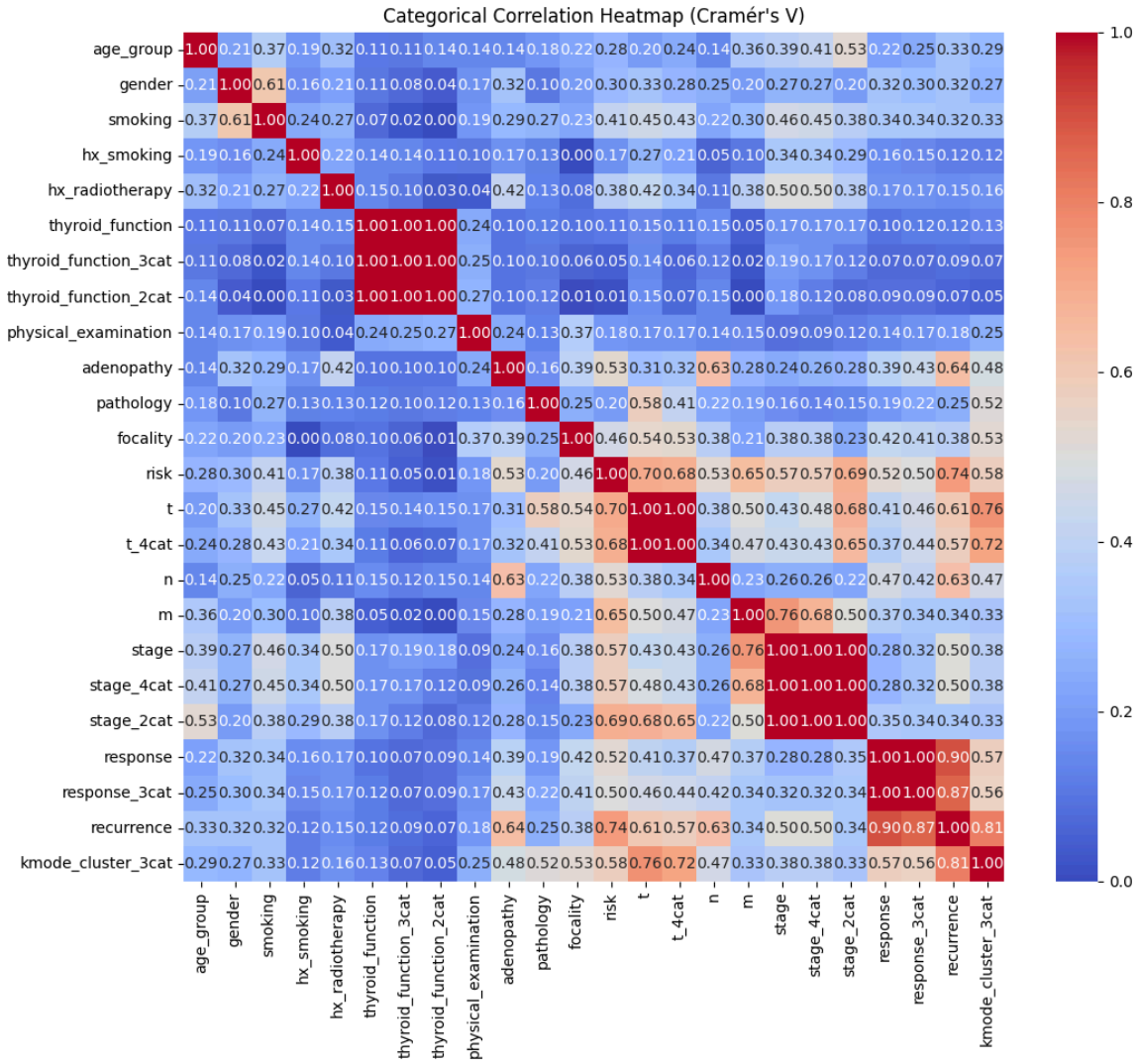






Correlation Heat Map

To show which columns have a correlation with each other, I made a Cramer V correlation heat map. Just to give note the limitation, all the numbers represent association but does not say that if the association is direct or inverse. This is because all our columns are categorical and not numeric.

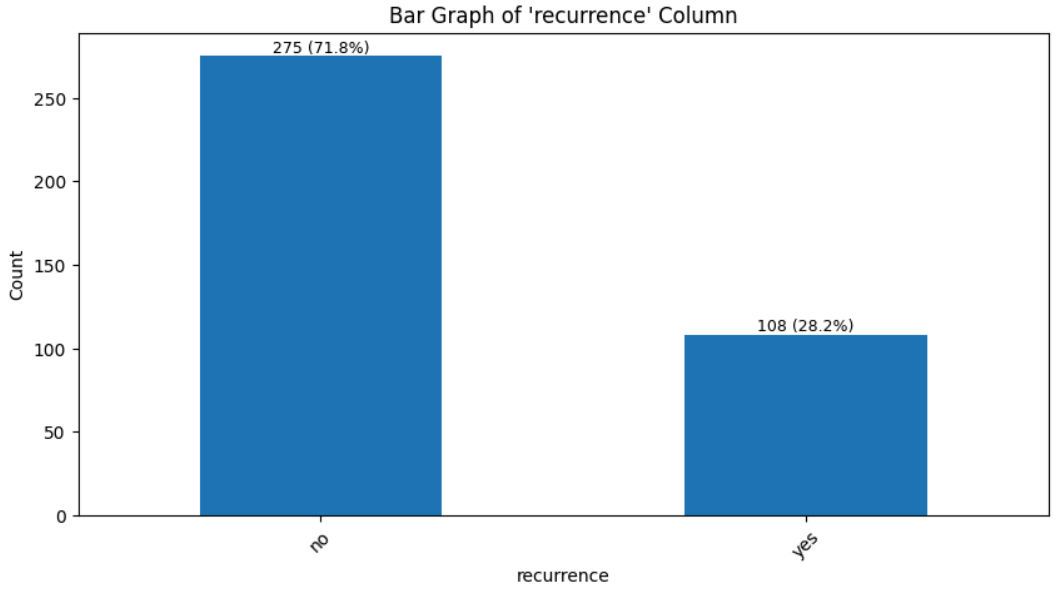


- I will be checking to see which features have strong associations with each other, but mainly focusing on 'recurrence' as that is what we will be using for labels in training.
- Merging the 'response' column into 'response_3cat' was a good move as combining both negative categories together gave it more variance info. However, that column is related to post-treatment and the goal of this project is to make a predictive model that is pretreatment based on info from examination.
- The 'Risk' column has a high association, but that feature is inputted vaguely by the doctor.

- 'Adenopathy' and 'n' columns have a high association and can work as they are less vague and clearer observation-based. Both are similar in that they are related to the observations of the appearance of the disease.
- Merging the 't' column into 4 categories made it have a slightly weaker association with recurrence. This means that the different subcategories in the 't' column do contain important variance info.
- Merging the 'stage' column into 4 categories has no difference.
- Interesting experimenting with k mode clustering the data into 3 groups that have high recurrence, will have to look into it further if meaningful or relies heavily on the recurrence column.

Recurrence column

Here is the recurrence column which will be out model's labels. Exploring noticed alot of data imbalance with this data set. Which we will have to address later.



Feature Engineering

We encode all binary data using dummy encoding. For non-binary categorical data we tried both One hot encoding and label encoding. After some time thinking we decided to use label encoding, as research shows it is better for the tree-based modeling systems we are about to use in modeling. And we split the data set into 80 20 training test splits.

Modeling

Due to all our features being categorical data, we find it best to use tree-based models. We made 3 models (basic decision tree, Random Forest, Gradient Boosting), and for each of them, I tried another version using methods to balance out the test data. So 6 models in total.

Native with imbalance data:

Decision Tree:

	precision	recall	f1-score	support
False	0.95	0.97	0.96	58
True	0.89	0.84	0.86	19
accuracy			0.94	77
macro avg	0.92	0.90	0.91	77
weighted avg	0.93	0.94	0.93	77

Random Forest:

	precision	recall	f1-score	support
False	0.98	0.98	0.98	58
True	0.95	0.95	0.95	19
accuracy			0.97	77
macro avg	0.97	0.97	0.97	77
weighted avg	0.97	0.97	0.97	77

Gradient Boosting:

	precision	recall	f1-score	support
False	0.98	1.00	0.99	58
True	1.00	0.95	0.97	19
accuracy			0.99	77
macro avg	0.99	0.97	0.98	77
weighted avg	0.99	0.99	0.99	77

Balanced Data Modifications:

Decision Tree with SMOTE:

	precision	recall	f1-score	support
False	0.95	0.93	0.94	58
True	0.80	0.84	0.82	19
accuracy			0.91	77
macro avg	0.87	0.89	0.88	77
weighted avg	0.91	0.91	0.91	77

Random Forest with class_weight balanced:

	precision	recall	f1-score	support
False	0.98	0.98	0.98	58
True	0.95	0.95	0.95	19
accuracy			0.97	77
macro avg	0.97	0.97	0.97	77
weighted avg	0.97	0.97	0.97	77

Gradient Boosting with SMOTE:

	precision	recall	f1-score	support
False	0.98	0.98	0.98	58
True	0.95	0.95	0.95	19
accuracy			0.97	77
macro avg	0.97	0.97	0.97	77
weighted avg	0.97	0.97	0.97	77

The main thing we were focusing on would be the F1 scores for both True and False and accuracy metrics. The higher the score for both the better. As you can see, almost all of the 6 models (except for the decision tree with SMOTE) reach the required minimal accuracy of 95% from the problem identification. Both ensemble decision tree methods work great and can natively handle imbalanced data well. However, the random forest method seems suspiciously high in precision and recall. Balancing the data increases all stats in all models except for the decision tree for some reason, will need to look further into that.

Conclusion

Our original goal was to make an eligibility test model for doctors to predict if our company's drug will prevent thyroid cancer from recurring post-treatment.

After comparing all the models, we feel that the 'Random Forest Method with class weight balanced' model is the best due to it's high F1 scores and accuracy, while also being less suspicious of overfitting.

Future Scope:

- Currently, this prototype is trained on a limited amount of data. And I would love for our company to partner with hospitals to get more consented user data.
- Also, I would like to do more experimentation on all the models to see if they are overfitting or not
- The current model also can use more optimization like cross-validation
- I would love to research more on the Gradient Booster model to see what is up with its high precision and recall.
- Another reason why I chose the random forest model was because I love its parallel computing concept, which means we can easily scale up the model with more data (like when we do partner with hospitals). This feature effectively makes this model future-proof.
- We can further expand on this model to include cross-reaction chance with other drugs or stats

Possible Issues:

I do not know if management will find using the CC 4.0 Attribution license data an issue, as if we release the product with this data set then we would have to say somewhere on the product this data set is included. If needed when we do get the hospital data we can remove this dataset and its label.

Overall I think we are on track to fine-tuning and finishing up this model for production.