Identifying patients missing HS diagnosis

Mentors - Professor Lynn Petukhova
Austin Bell

Students - Arusha Kelkar
Tanvi Pareek
Karthik Rajaraman Iyer
Kanak Singh
Manas Dresswala

Introduction

Electronic Health Records (**EHR**) and electronic biorepositories **(EB)** that link EHR to genetic data offer a cost- and time-efficient solution to building cohorts for **well-powered** clinical and research studies of **HS**.

Care providers often fail to recognize that patients presenting with boils have a chronic inflammatory skin disease, such that up to **30% of people receiving healthcare to help manage HS symptoms are missing an HS diagnosis code** in their medical records[1,2].

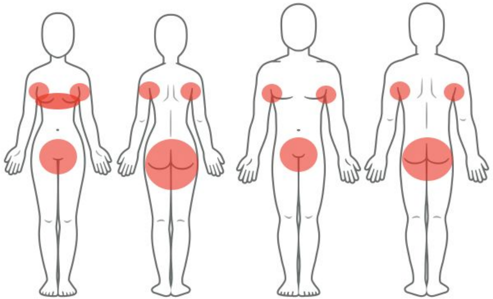
The development of a phenotyping algorithm that incorporates features in structured EHR data that are predictive of an HS diagnosis could help improve the identification of HS research participants and thus increase the size of HS cohorts and the power of HS studies

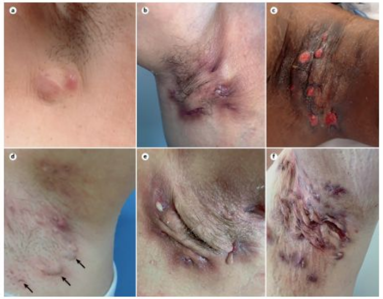
[1] Population-based Clinical Practice Research Datalink study using algorithm modelling to identify the true burden of hidradenitis suppurativa. The British journal of dermatology 2018;178:917-24.

[2] Kjaersgaard Andersen R, Jorgensen IF, Reguant R, Jemec GBE, Brunak S. Disease Trajectories for Hidradenitis Suppurativa in the Danish Population. JAMA dermatology 2020.

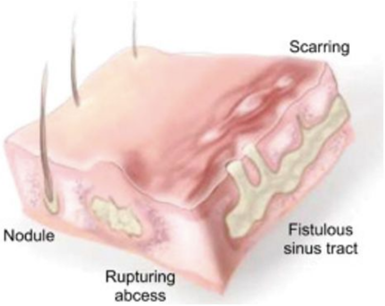
The Clinical Problem - **Hidradenitis Suppurativa**

Hidradenitis Suppurativa (**HS**) is a chronic inflammatory skin disease characterized by **recurrent** outbreaks of **painful** nodules and **boils** that eventually form fistulas and fibrosis in **distinct anatomical regions**.





Sabat et al., Nature Reviews, 2020

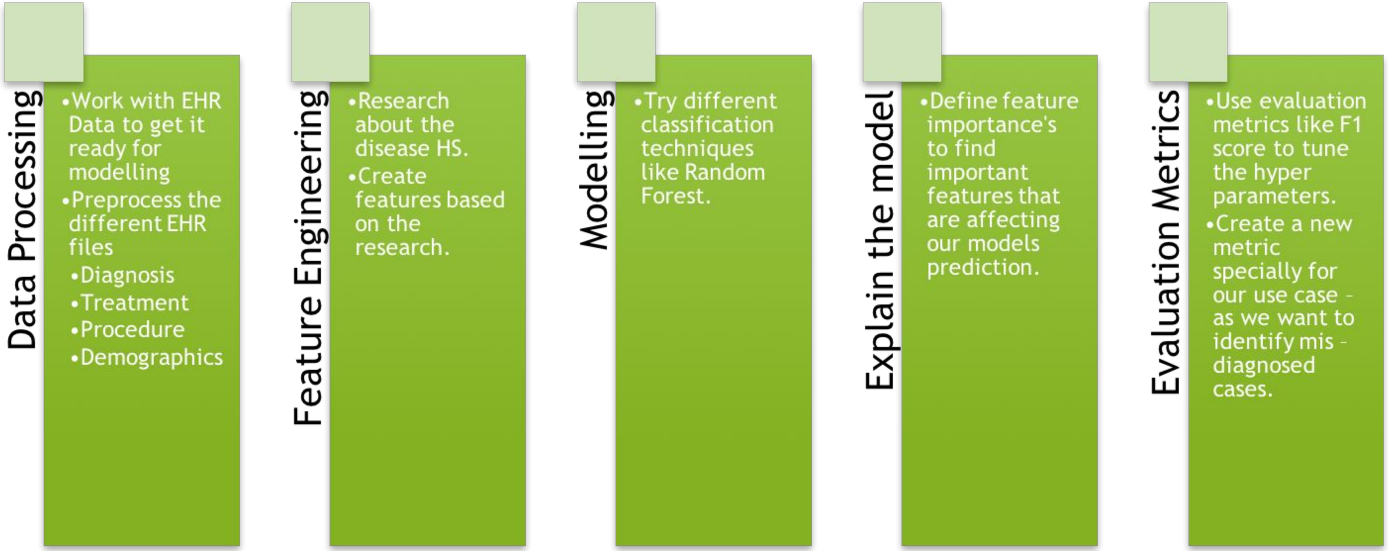


Elkin et al., Skin Res Technol. 2020

Imperative Need for Clinical and Research Studies of **HS**

- Painful, debilitating, stigmatizing
- Prevalent (1%)
- High unmet needs
    - few treatment options
    - limited evidence for most treatments
- Expensive to manage symptoms for patients and healthcare systems

A data driven approach



Data Processing

- Work with EHR Data to get it ready for modelling
- Preprocess the different EHR files
  - Diagnosis
  - Treatment
  - Procedure
  - Demographics

Feature Engineering

- Research about the disease HS.
- Create features based on the research.

Modelling

- Try different classification techniques like Random Forest.

Explain the model

- Define feature importance's to find important features that are affecting our models prediction.

Evaluation Metrics

- Use evaluation metrics like F1 score to tune the hyper parameters.
- Create a new metric specially for our use case - as we want to identify mis - diagnosed cases.

Goal

To differentiate between patients who have HS symptoms and patients who actually have HS. Develop a model to identify patients who have HS but did not receive a diagnosis.

We first make the assumption that any patient with an HS diagnosis does have HS, but our hypothesis is that patients without an HS diagnosis may actually have HS.

Dataset

To develop the model we have used patient electronic health record (EHR) data from Columbia's datastore.

There are around 3000 patients coming from this data.

A patient's EMR includes information on their demographics (age, gender, ethnicity, etc.), diagnoses (e.g., an ICD code for whether a doctor diagnosed them with a disorder), medications, and procedures.

Labels

HS ICD Code : Patients who actually have HS

NLP symptoms : Patients who have symptoms like HS, determined parsing the clinical notes and other documents (i.e., free text notes within a patient's EMR) using an NLP software

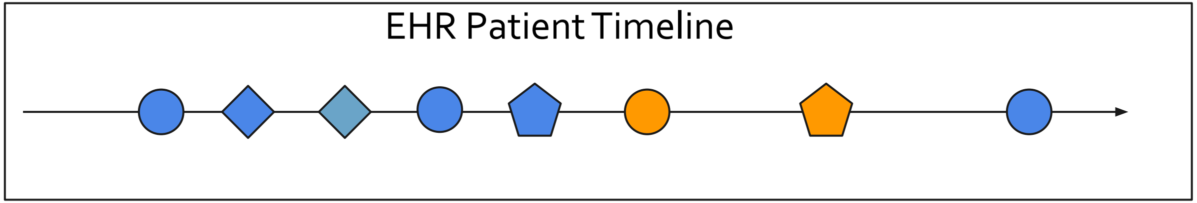
NLP - hidradenitis: patients that have HS but are missing the HS diagnosis code, again tagged using the NLP software

Literature Survey

As mentioned before, HS is a prevalent disease (almost 1% of the population has it). Despite of its high prevalence, physicians are missing this disease.
According to previous research, there is a diagnostic delay of approximately 7 years in identifying the disease.
HS is also associated with a lot of comorbidities like smoking, depression, acne, obesity etc. We researched on the symptoms of HS and engineered features based on our research.
HS is more prevalent in women over men.

There is not much done to create a data driven approach in identifying HS patients. One of the goals of this project is to use the model in tagging new patients so that we can increase the dataset size of HS patients, which in turn can be used to understand the disease better.

Data Preprocessing



EHR Patient Timeline

Data Source: Electronic Health Records (EHR)
Target Source: Labels from clinical notes

Along with EHR data, we also use the **demographic information** related to each patient, such as the gender and age

● ● => Diagnosis

◆ ◆ => Treatments

⬟ ⬟ => Procedures

Feature Engineering

Diagnosis

Diagnosis information is encoded using ICD10* codes
We compute a feature vector that contains the counts of each diagnosis that the patient had
We use various groupings such as pheocodes to compress the dimensionality of this vector
We also engineered specific indicators and measures derived from the literature, such as

- Counts of boils, furuncle and abscess per region
- Percentage of symptoms in flexural vs non-flexural sites
- Time duration of symptoms (from first occurrence to last occurrence)
- Existence of symptoms in early ages (18-29)
- Existence of comorbidities and mood disorders

*10th revision of International Statistical Classification of Diseases and Related Health Problems (ICD)

Procedures

Procedures are encoded using CPT* and ICD10 codes.
Given the incoherency between the two codes we use the description of the procedures to select a specific subset of procedures that are related to the treatment protocol for skin diseases.
We generate counts of these set of procedures for each patient.
Examples of such procedures are:

- Injections and Infusions
- Drainage of abscess
- Nebulizer therapy

*Current Procedural Terminology (CPT) code set is a medical code set maintained by the American Medical Association(AMA)

Treatments

Treatments/Medication information have to be identified by their description.
We extract the medicine name and quantity from the text data.

We use a two-step approach to extract the medicinal information:

1. We use an external NLP tool called Stanza* to tag the medicine names.
2. We use regular expression to capture the quantities(dosage) and also to tag medications that Stanza missed.

*Stanza is a natural language analysis package built by the Stanford NLP group

Modeling

A classification problem

We assumed that the patients tagged HS ICD code and NLP - hidradenitis are the same. The only difference between patients is that the latter one is missing an HS diagnosis code.

After making this safe assumption, we made our problem a binary classification problem

- Positive class or 1 : patient has HS (HS ICD Code and NLP - hidradenitis)
- Negative class or 0: patient does not have HS diagnosis (NLP - symptoms)

Machine Learning Model

- We used Tree based models because explaining the model as well as the important features was important.

Note: All models were evaluated on a common test set which has 763 patients

Evaluation Metrics

To evaluate the different models we use the F1 score as well as ROC AUC score.

F1 score was used to get an overall idea about how our model is performing, giving equal weightage to precision and recall.

ROC AUC score is the most common metric used in healthcare classification problems.

Further we created a new set of evaluation metrics, let’s call it Evaluation B.

Evaluation B

As mentioned earlier, we have different labels in our test set - HS ICD Code, NLP - HS and NLP - Symptoms. We care more about getting the patients tagged by the Columbia NLP software correct (as the other patients have an HS Diagnosis code present).

To understand this, we remove the patients with the HS ICD label from our test set and only keep the patients tagged by the NLP software. Once we do this, we calculate the F1 scores again. This becomes our main metrics to compare the models on.

Baseline Model

ML Model: Random Forest

Features: Diagnosis and Demographics

Results:

Evaluation	No. of samples in each class	F1 on 0	F1 on 1	ROC-AUC score
A	0: 201, 1: 557	0.36	0.8	0.578
B	0: 201, 1: 88	0.42	0.45	0.517

Dealing with class imbalance

- Unlike the normal class imbalance problem where the positive class samples are usually less, we have a higher number of samples for the positive class and a very less number of samples for the negative class thus introducing the problem of class imbalance.

- Data oversampling is a technique applied to generate data in such a way that it resembles the underlying distribution of real data. The oversampling is done in a way such that more samples are added to the minority class i.e. in our case we oversampled the negative samples in the training set.

- We tried both random oversampling and SMOTE but achieved better results using random oversampling and hence decided to use it for the final model.

Best model

- We achieved best results with XGBoost and random oversampling. After trying a variety of combination of features like important diagnosis, treatments, procedures, demographics, we obtained the best results with the following:

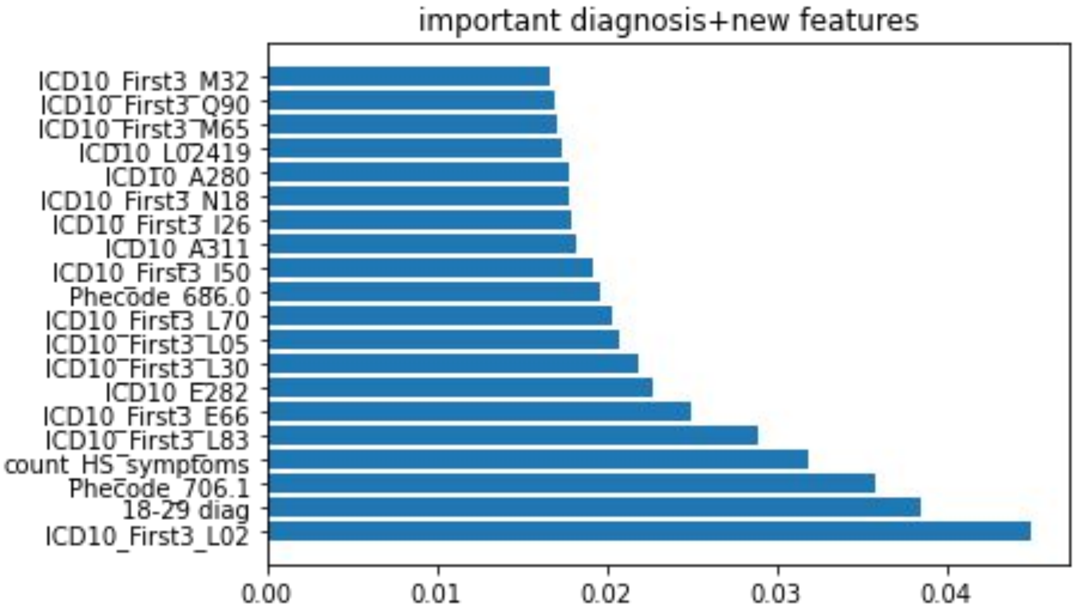
  ML Model: XGBoost with random oversampling

  Features: Important diagnosis and new features

  Results:

Evaluation	No. of samples in each class	F1 on 0	F1 on 1	ROC-AUC score
A	0: 201, 1: 562	0.44	0.78	0.617
B	0: 201,1: 88	0.6	0.52	0.623

Important features



Conclusion

From the results, we can clearly see that we need to solve the class imbalance issue to get a good score for our problem at hand. Thus, when we perform random oversampling on the train set and then train a XGBoost Classifier.

Using evaluation B, we get a F1 score on label 0 as 0.62 and an F1 score on label 1 as 0.52. This tells us that we are doing well in predicting patients that have only HS symptoms and not the disease HS.

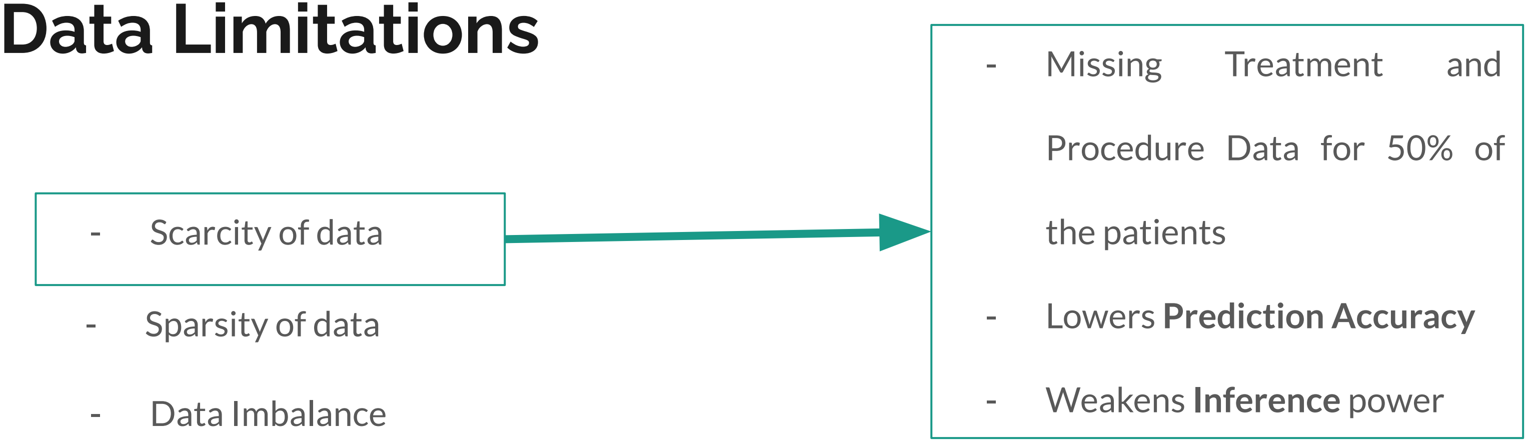
We feel that this model performs the best because firstly we have solved the class imbalance issue.

Secondly, the features that we give to the model are tailored according to the disease we are studying. This helps the model understand the patients better. Feature engineering played a major role in getting this particular score for our models.

Data Limitations

- Scarcity of data
- Sparsity of data
- Data Imbalance

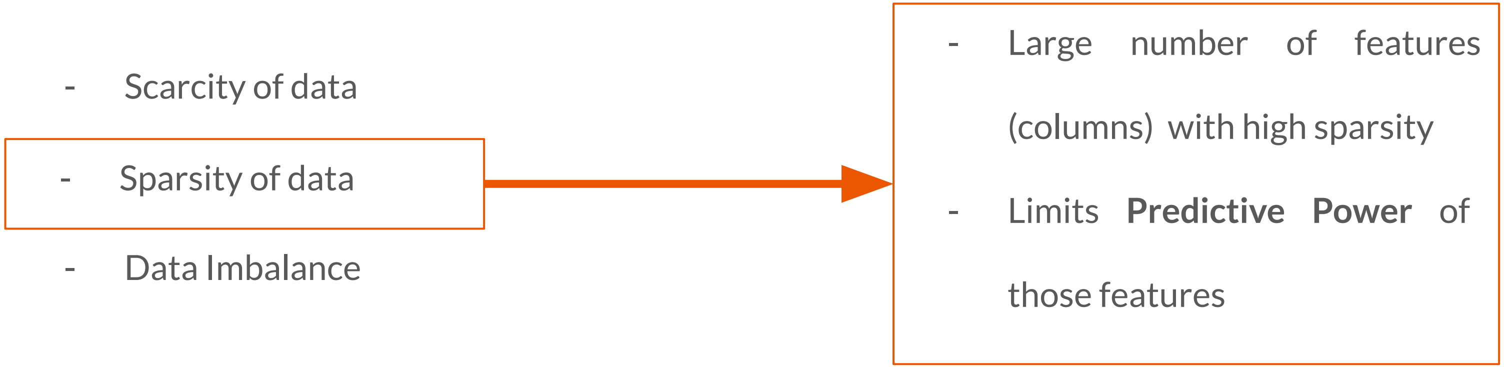
Data Limitations



- Scarcity of data
- Sparsity of data
- Data Imbalance

- Missing Treatment and Procedure Data for 50% of the patients
- Lowers **Prediction Accuracy**
- Weakens **Inference** power

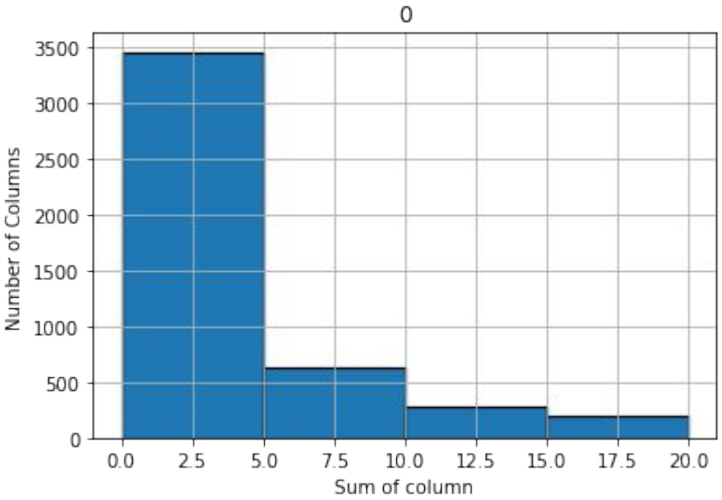
Data Limitations



- Scarcity of data
- Sparsity of data
- Data Imbalance

- Large number of features (columns) with high sparsity
- Limits **Predictive Power** of those features

Data Sparsity

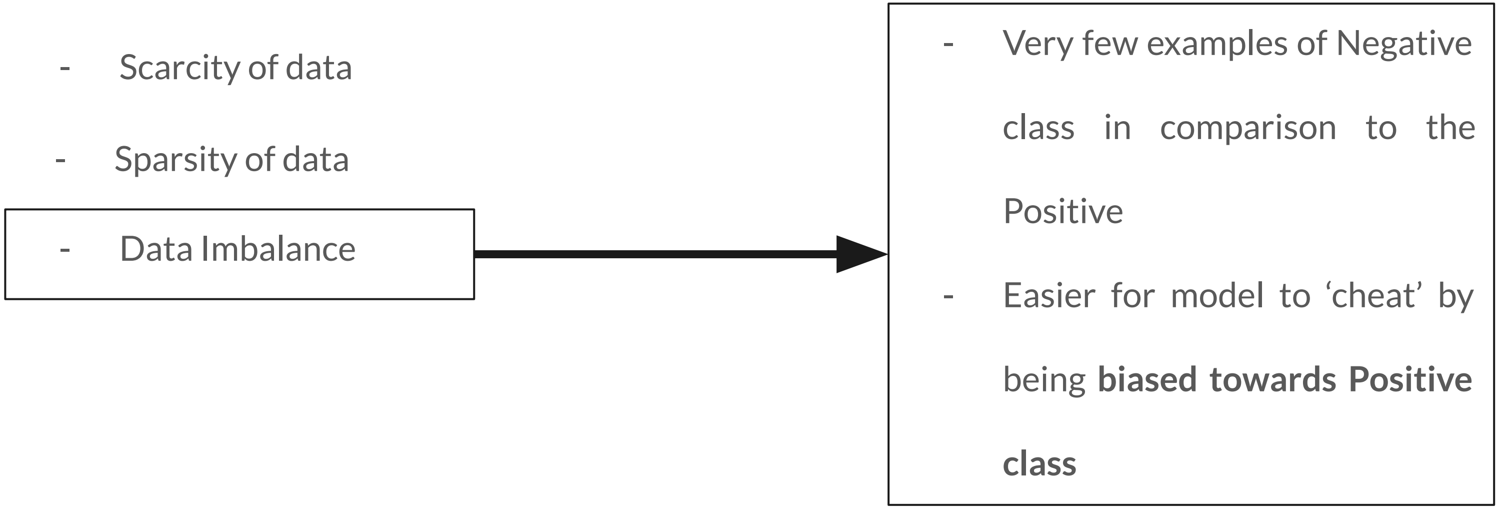


As expected, sparse features do not appear in ‘Top 40 important features’.

Is this sparsity Natural or a Data Entry issue?

Data Limitations

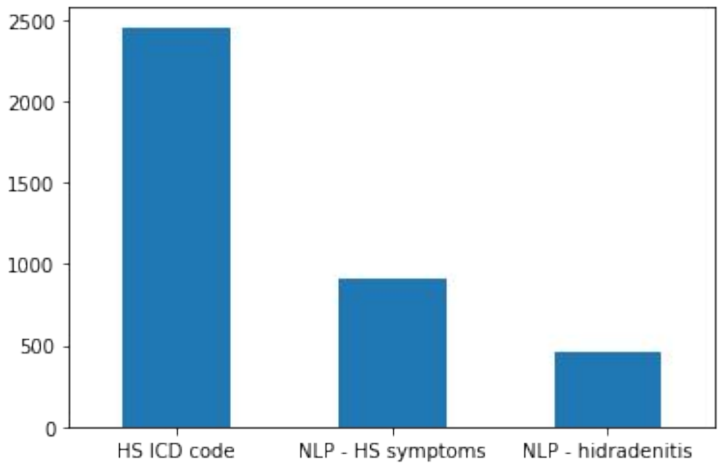
- Scarcity of data
- Sparsity of data



- Data Imbalance

- Very few examples of Negative class in comparison to the Positive
- Easier for model to ‘cheat’ by being **biased towards Positive class**

Data Imbalance



Limited exposure to Negative data.

Only 915 out of 3836 patients, or 23% of the patients, have NLP - HS symptoms (Negative class)

Next steps - Model Improvement

- Incorporating Time Element into Model
- Feature Engineering after deeper Literature Review
- Manual Hyper Parameter Tuning
- Visualization of Random Forest for validation with Domain Experts

Next steps - Model Improvement

- Incorporating Time Element into Model
- Feature Engineering after deeper Literature Review
- Manual Hyper Parameter Tuning
- Visualization of Random Forest for validation with Domain Experts

❖ RNNs

❖ Time Series Forecasting - ARIMA

Next steps - Model Improvement

- Incorporating Time Element into Model
- Feature Engineering after deeper Literature Review
- Manual Hyper Parameter Tuning
- Visualization of Random Forest for validation with Domain Experts

Composite Features like **Skin-Health specific metrics** and **Medical indicators**

Next steps - Model Improvement

- Incorporating Time Element into Model
- Feature Engineering after deeper Literature Review
- Manual Hyper Parameter Tuning / Model Selection
- Visualization of Random Forest for validation with Domain Experts

Additionally Optimizing for:

❖ Model Parsimony

❖ Generalizability

❖ Interpretability

Next steps - Model Improvement

- Incorporating Time Element into Model
- Feature Engineering after deeper Literature Review
- Manual Hyper Parameter Tuning / Model Selection
- Visualization of Random Forest for validation with Domain Experts

Reporting and validating Decision Tree **branching logic**

(Random Forest = ensemble of Decision Trees)

Thank You