Detecting COVID19 Underlying Conditions:

CASE STUDY: DIABETES

30-04-2020

Presenter:
Krukrubo Alaso Lawrence

Project Goal

The WHO has stated that up to 40% of COVID19 carriers may be asymptomatic. This is one major reason why the spread of the virus has been unprecedented and unrivalled in history. See link
Some infected people may not display compelling symptoms, yet such people are as infectious as those severely sick.

Therefore the objective of this project is to assist medical practitioners to quickly diagnose underlying conditions in patients who may be Diabetic. This will help to quickly inform those with emerging to acute levels of Diabetes to first be aware, and secondly take immediate remedies to tackle their Diabetes on time, just incase they eventually contract the Corona virus. Early detection of underlying conditions of COVID19 will help reduce fatality rates as more people get treated for pre-conditions before exposure to COVID19.

Data Dictionary:

Kaggle-Link

Context:

This dataset is originally from the National Institute of Diabetes and Digestive and Kidney Diseases. The objective of the dataset is to diagnostically predict whether or not a patient has diabetes, based on certain diagnostic measurements included in the dataset. Several constraints were placed on the selection of these instances from a larger database. In particular, all patients here are females at least 21 years old of Pima Indian heritage.

Content:

The datasets consists of several medical predictor variables and one target variable, Outcome. Predictor variables includes the number of pregnancies the patient has had, their BMI, insulin level, age, and so on.

Acknowledgements:

Smith, J.W., Everhart, J.E., Dickson, W.C., Knowler, W.C., & Johannes, R.S. (1988). Using the ADAP learning algorithm to forecast the onset of diabetes mellitus. In Proceedings of the Symposium on Computer Applications and Medical Care (pp. 261--265). IEEE Computer Society Press.

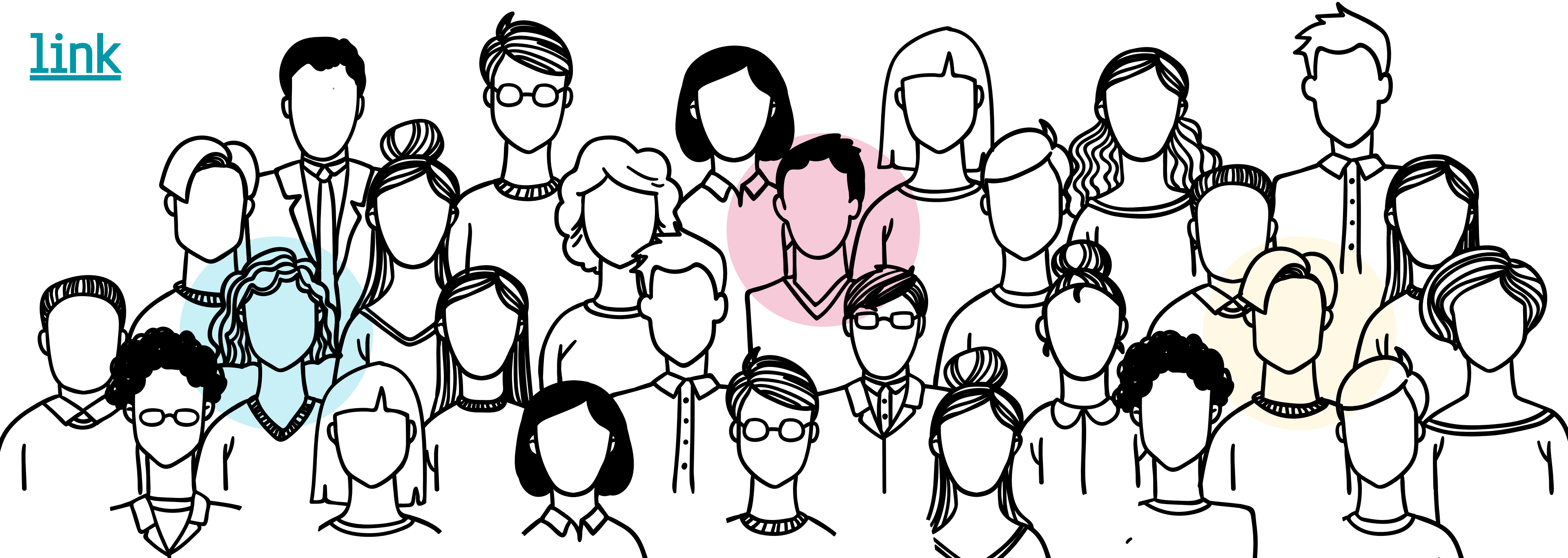
CONTENTS

Preface:

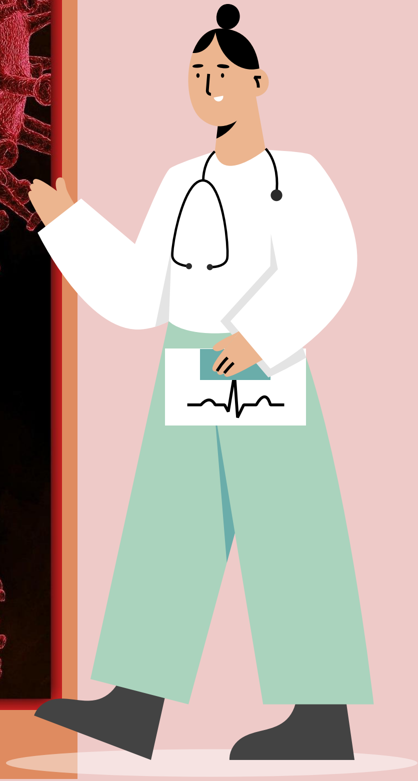
According to the World Health Organisation

Diabetes is one of the leading causes of death in the world.
About 422 million people worldwide have Diabetes.
The majority of Diabetes occur in low and middle income countries
Early detection and intervention is the key to surviving Diabetes.

[link]

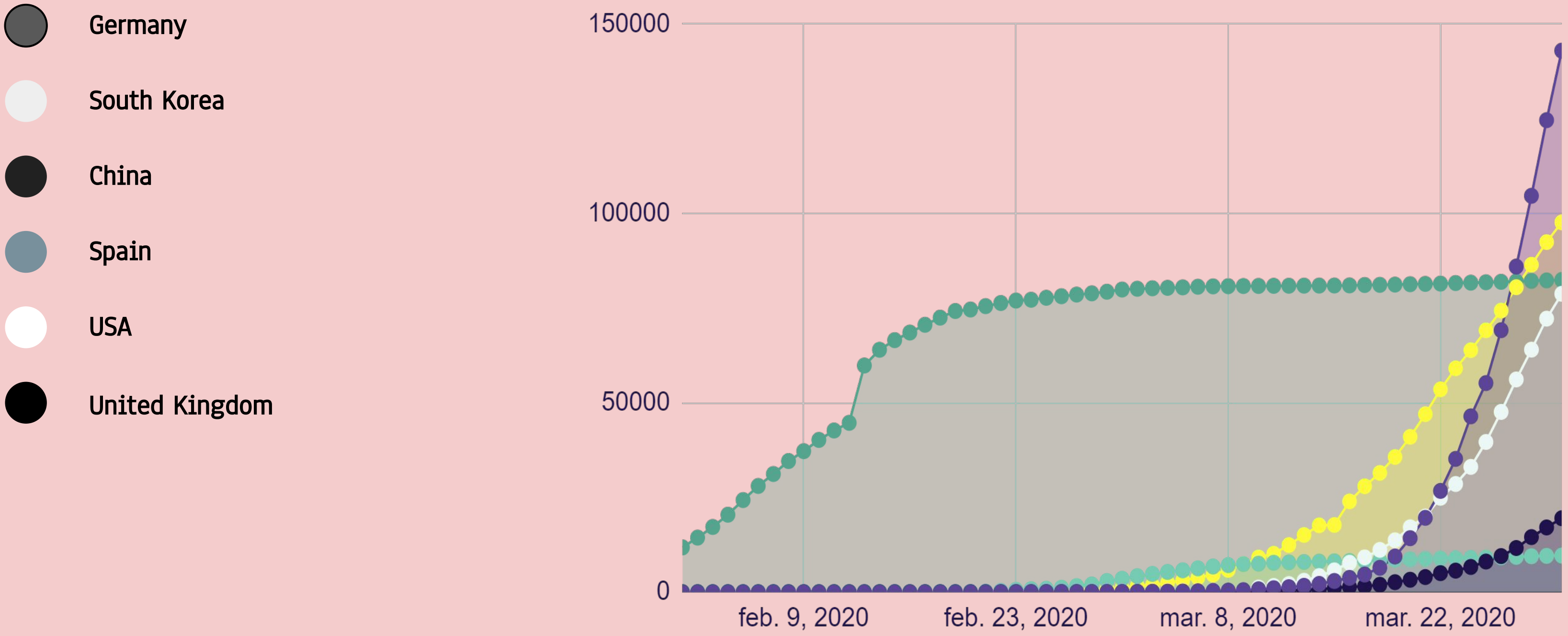


THE PROBLEM...



With the out-break of the Corona-Virus pandemic, people with Diabetes are much more likely to develop severe conditions and death. see link from the CDC.

EXPONENTIAL GROWTH of COVID19 SPREAD



THE SOLUTION...

Using data from previous patients history, I have built an XGBClassifier model that can detect on new data, the onset of Diabetes with over 96% Accuracy, F1_score and AUC_score

PATIENTID	AGE	...	BMI	DIABETIC
001	36	...	-0.68	YES
002	42	...	-0.69	YES
003	34	...	-0.02	NO

THE SOLUTION STAGES...

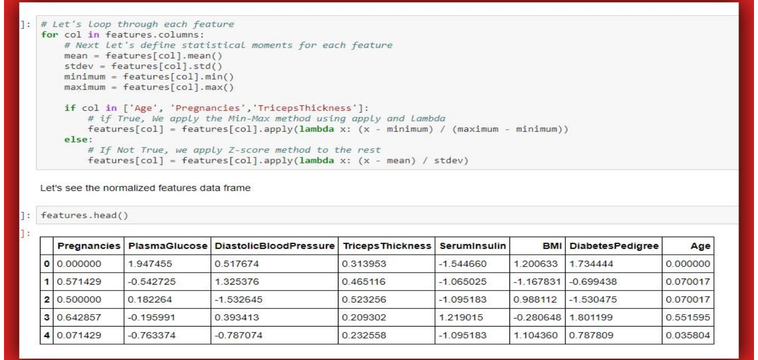
My Solution involved the following steps:

1. Problem Definition and Scope
2. Data Acquisition
3. Exploratory Data Analysis
4. Feature Engineering
5. Feature Selection
6. Balancing the Dataset
7. Normalising the Data
8. Splitting to Train, Val and Test sets
9. Applying LALE pipeline
10. Applying GridSearchCV
11. Selecting the best Model - XGB Classifier.
12. Making Predictions
13. Evaluating Predictions
14. Interpreting Predictions
15. Applying Explainability
16. Using Permutaion Importance
17. Using aix360
18. Retraining the XGB Model
19. Saving the model for production

Link to Solution

Some notable pre-processing highlights include...

Applying Min-Max & Z-score Normalisation



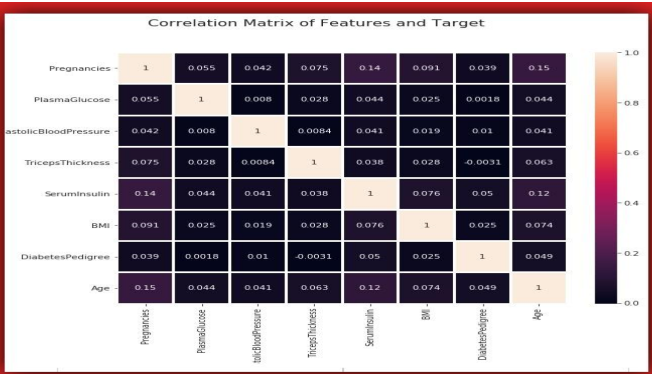
```
# Let's loop through each feature
for col in features.columns:
    # Next let's define statistical moments for each feature
    mean = features[col].mean()
    stdev = features[col].std()
    minimum = features[col].min()
    maximum = features[col].max()

    if col in ['Age', 'Pregnancies','TricepsThickness']:
        # if True, We apply the Min-Max method using apply and lambda
        features[col] = features[col].apply(lambda x: (x - minimum) / (maximum - minimum))
    else:
        # If Not True, we apply Z-score method to the rest
        features[col] = features[col].apply(lambda x: (x - mean) / stdev)
```

Let's see the normalized features data frame

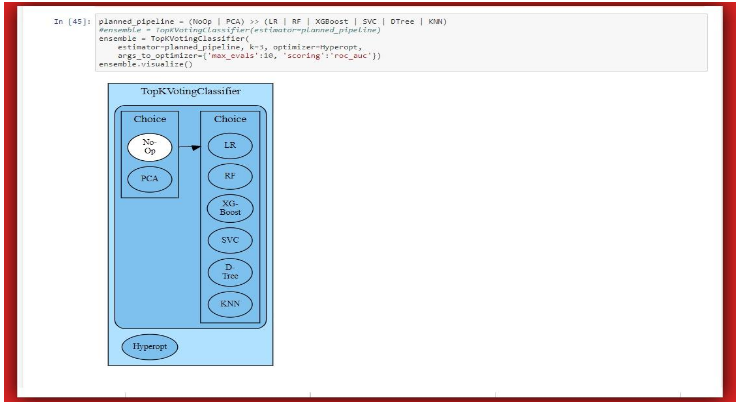
```
features.head()
```

	Pregnancies	PlasmaGlucose	DiastolicBloodPressure	TricepsThickness	SerumInsulin	BMI	DiabetesPedigree	Age
0	0.000000	1.947455	0.517674	0.313953	-1.544660	1.200633	1.734444	0.000000
1	0.571429	-0.542725	1.325376	0.465116	-1.065025	-1.167831	-0.699438	0.070017
2	0.500000	0.182264	-1.532645	0.523256	-1.095183	0.988112	-1.530475	0.070017
3	0.642857	-0.195991	0.393413	0.209302	1.219015	-0.280648	1.801199	0.551595
4	0.071429	-0.763374	-0.787074	0.232558	-1.095183	1.104360	0.787809	0.035804

Evaluating Multi-Collinearity in Features

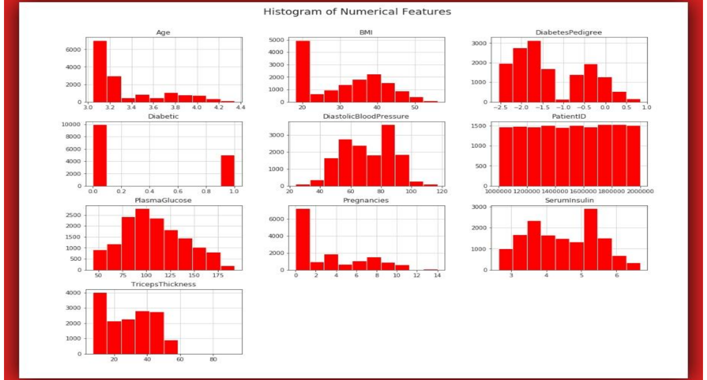


Applying LALE Pipeline semi-automated ML



```
In [45]: planned_pipeline = (NoOp | PCA) >> (LR | RF | XGBoost | SVC | DTree | KNN)
         #ensemble = TopKVotingClassifier(estimator=planned_pipeline)
         ensemble = TopKVotingClassifier(
             estimator=planned_pipeline, k=3, optimizer=Hyperopt,
             args_to_optimizer={'max_evals':10, 'scoring':'roc_auc'})
         ensemble.visualize()
```

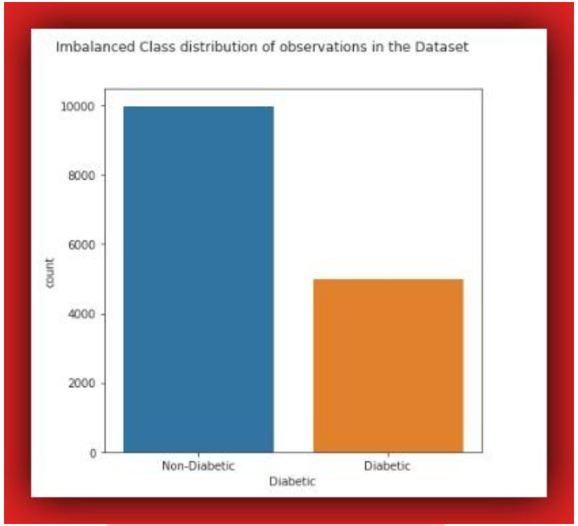

Applying Feature-Engineering

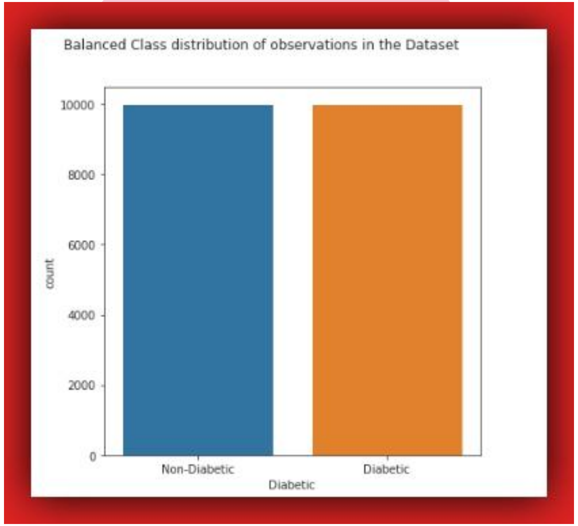


Building an Unbiased Model.

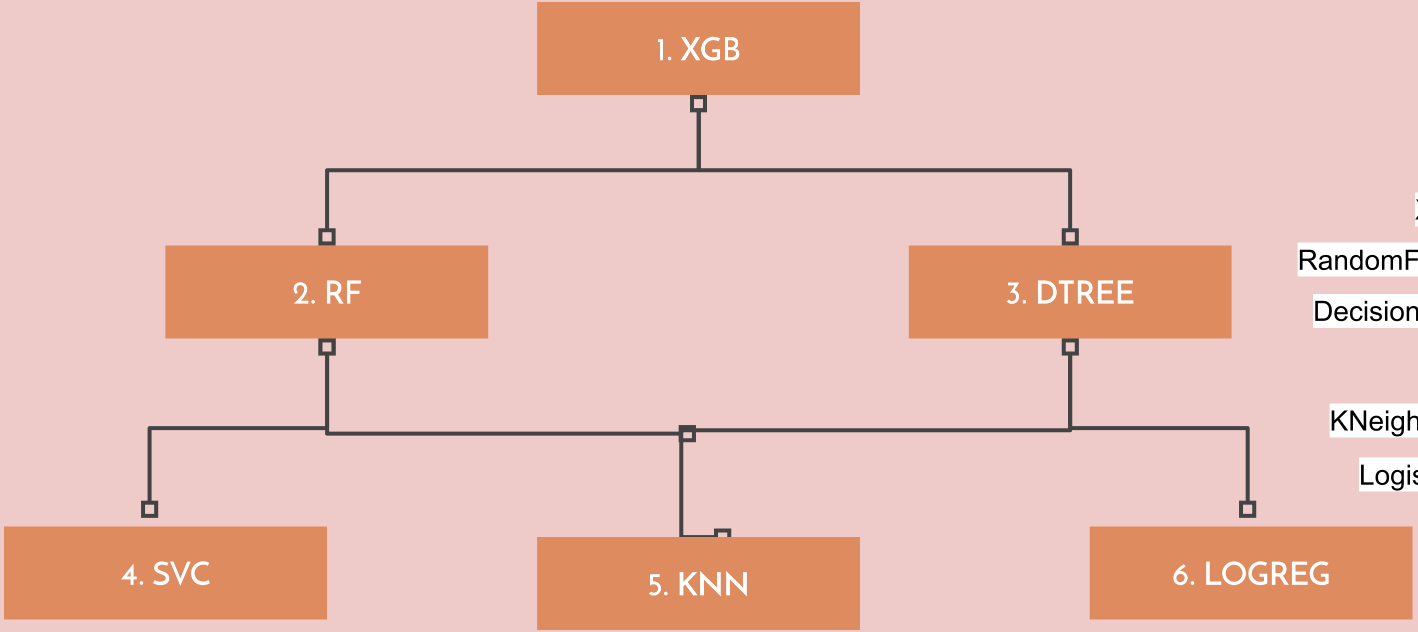
If the data is biased, the model is likely biased...

- I had 10000 Non-Diabetic and 5000 Diabetic observations.
- So I applied SMOTE Over-Sampling technique to equate the classes to 20000 total.
- I also set stratify to the target variable to ensure data split conforms to original data.
- I applied Z-score normalisation to features with a close-to normal distribution and min-max normalisation to others with asymmetrical or Unimodal distribution





Model Selection...



RESULTS

XGBClassifier 0.96 +/- 0.01

RandomForestClassifier 0.94 +/- 0.01

DecisionTreeClassifier 0.91 +/- 0.01

SVC 0.87 +/- 0.01

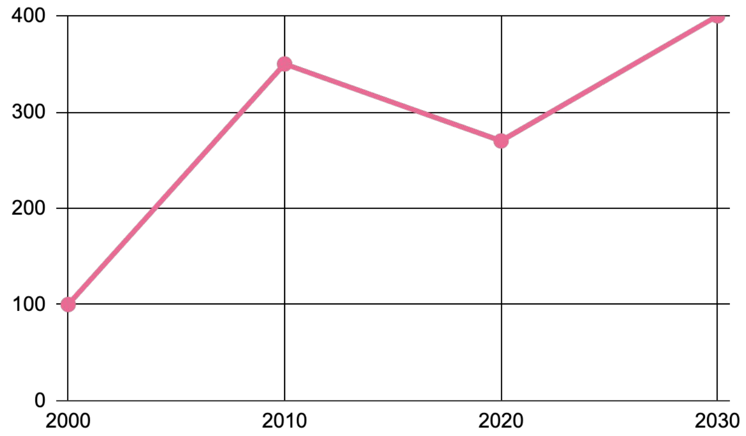
KNeighborsClassifier 0.86 +/- 0.01

LogisticRegression 0.78 +/- 0.01

From the six models depicted above, it turns out that XGBoost Model performed best with a Cross_Val_Score of 0.96. See results above for each model.

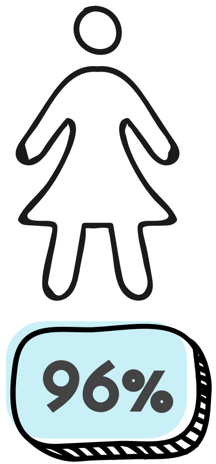
Model Performance

ACCU: 0.96 | F1: 0.96 | AUC: 0.96

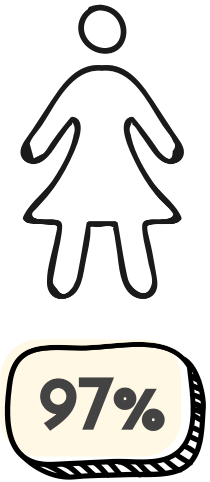


Turns out that on all three parameters above, the XGB Model performed at 96% in the validating set. Then I stacked the validating set (np.vstack) to the training set and retrained the model. It improved slightly on all parameters on Testing set.

DATA SETS



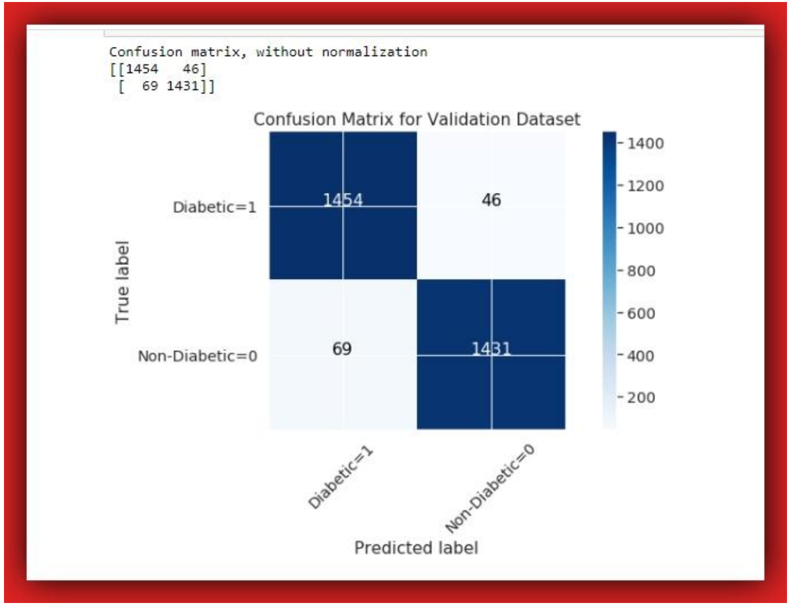
validating Set



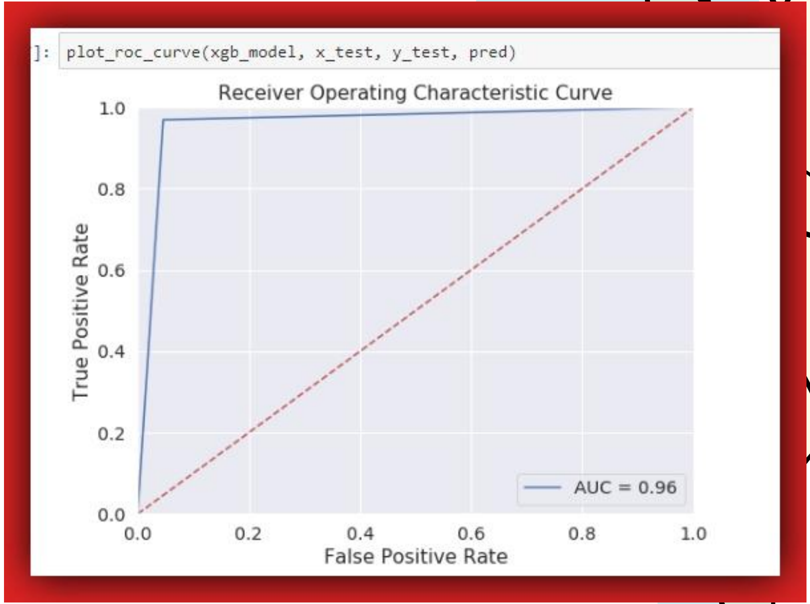
Testing Set

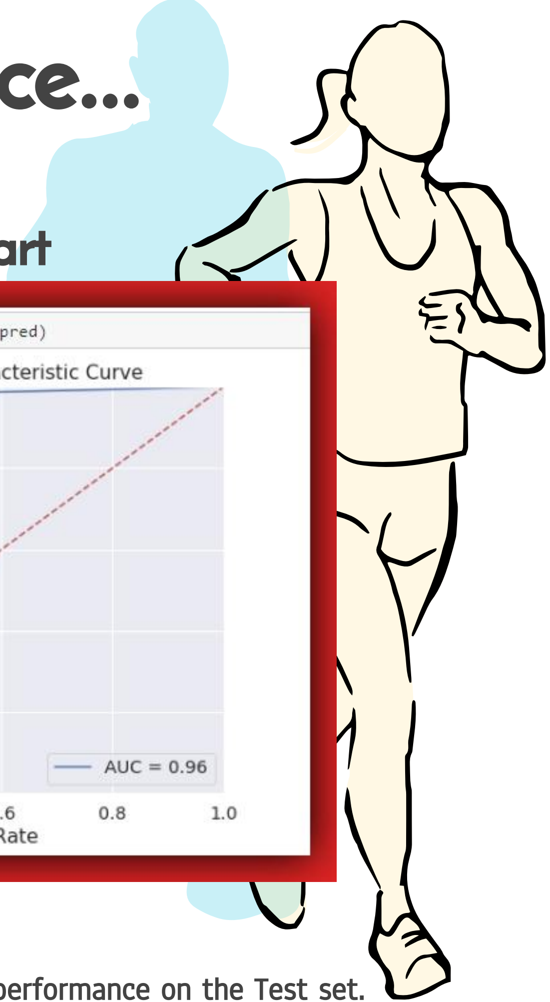
Visualizing Model Performance...

Confusion Matrix



ROC Chart





Displaying the Confusion matrix and the Receiver Operator Characteristics Chart for the Model performance on the Test set.

EXPLAINABILITY by Permutation Importance...

The Top Four Features affecting model predictions are:

Pregnancies

Age

BMI

SerumInsulin

```
# Let's show the weights of features in the data set
eli5.show_weights(perm, feature_names = class_names_)
```

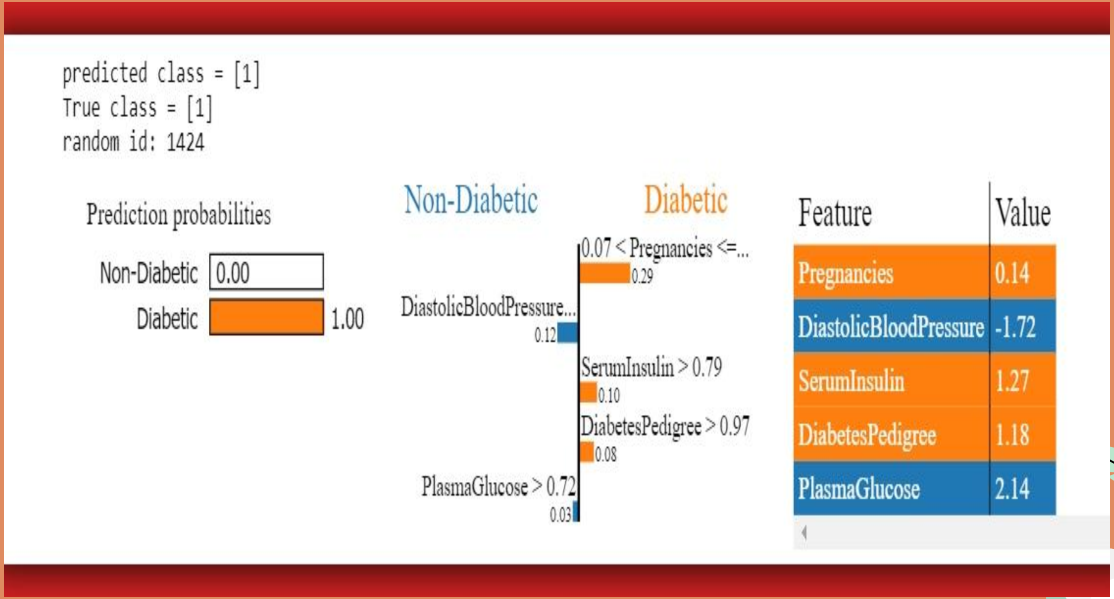
Out[80]:

Weight	Feature
0.1105 ± 0.0123	Pregnancies
0.0721 ± 0.0085	Age
0.0371 ± 0.0069	BMI
0.0257 ± 0.0034	SerumInsulin
0.0211 ± 0.0037	PlasmaGlucose
0.0177 ± 0.0040	TricepsThickness
0.0094 ± 0.0023	DiastolicBloodPressure
0.0039 ± 0.0037	DiabetesPedigree

Interpreting Permutation Results:

EXPLAINABILITY by AIX360…

CASE 1:
Diabetic Classified as Diabetic



Diabetic Indicating Variable Scores:

1. Pregnancies = 0.14
2. SerumInsulin = 1.27
3. DiabetesPedigree = 2.14

Non-Diabetic Indicating Variable Scores:

1. DiastolicBP = -1.72
2. PlasmaGlucose = 2.14

AIX360 EXPLAINABILITY SUMMARY...

CASE 1

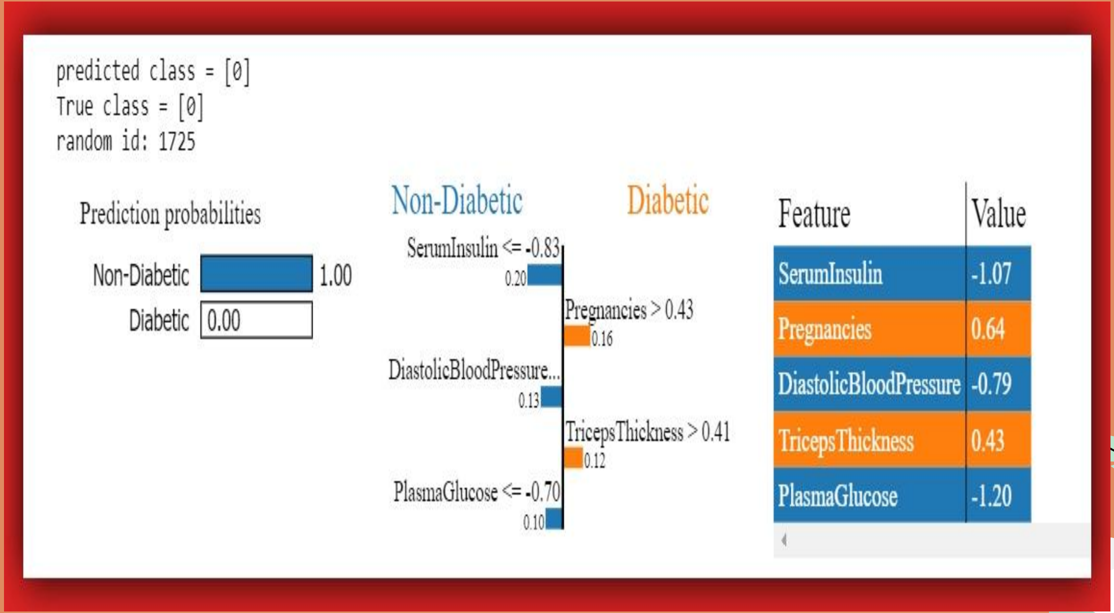
- True Positive Classified as True Positive **(TP)**
- With Explainability, The Doctor is better informed and more confident in the Model classification. Thus, he can better provide fine-grain recommendations to the Patient.

EFFECT

- For the top 5 features in the chart, the doctor has ample info to manage this Patient based on the Patients' probability scores per feature.
- Although the Patient is classified as Diabetic, The Doctor can clearly see that he is within healthy BloodPressure and PlasmaGlucose levels.

EXPLAINABILITY by AIX360...

CASE 2:
Non-Diabetic Classified as Non-Diabetic



Diabetic Indicating Variable Scores:

1. Pregnancies = 0.64
2. TricepsThickness = 0.43

Non-Diabetic Indicating Variable Scores:

1. SerumInsulin = -1.07
2. DiastolicBloodPressure = -0.79
3. PlasmaGlucose = -1.20

AIX360 EXPLAINABILITY SUMMARY...

CASE 2

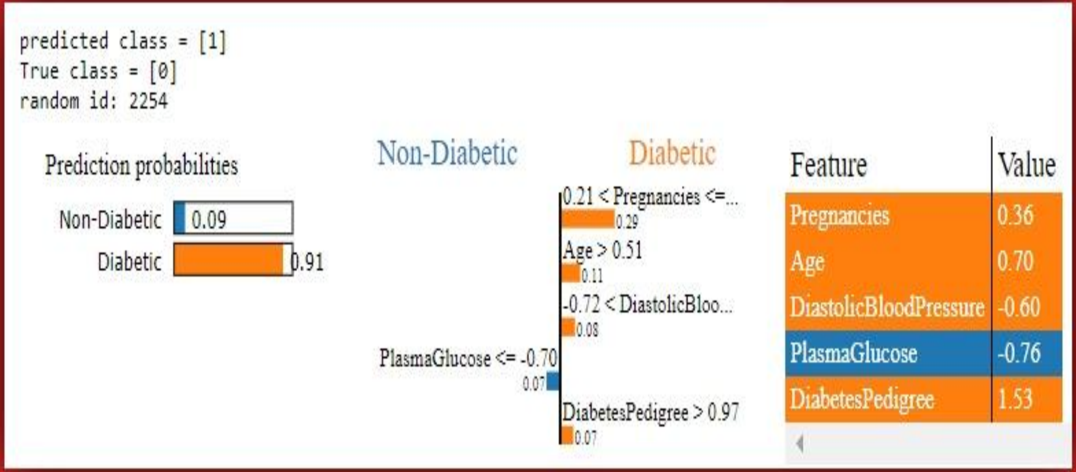
- True Negative Classified as True Negative **(TN)**
- With Explainability, The Doctor is better informed and more confident in the Model classification. Thus, he can better provide fine-grain recommendations to the Patient.

EFFECT

- For the top 5 features in the chart, the doctor has ample info to manage this Patient based on the Patients' probability scores per feature.
- Although the Patient is classified as Non-Diabetic, The Doctor can clearly see that the he is within unhealthy limits in Pregnancies and TricepsThickness features.

EXPLAINABILITY by AIX360...

CASE 3:
Non-Diabetic Mis-Classified as Diabetic



Diabetic Indicating Variable Scores:

1. Pregnancies = 0.36
2. Age = 0.70
3. DiastolicBP = -0.60
4. DiabetesPedigree = 1.53

Non-Diabetic Indicating Variable Scores:

1. PlasmaGlucose = -0.76

AIX360 EXPLAINABILITY SUMMARY...

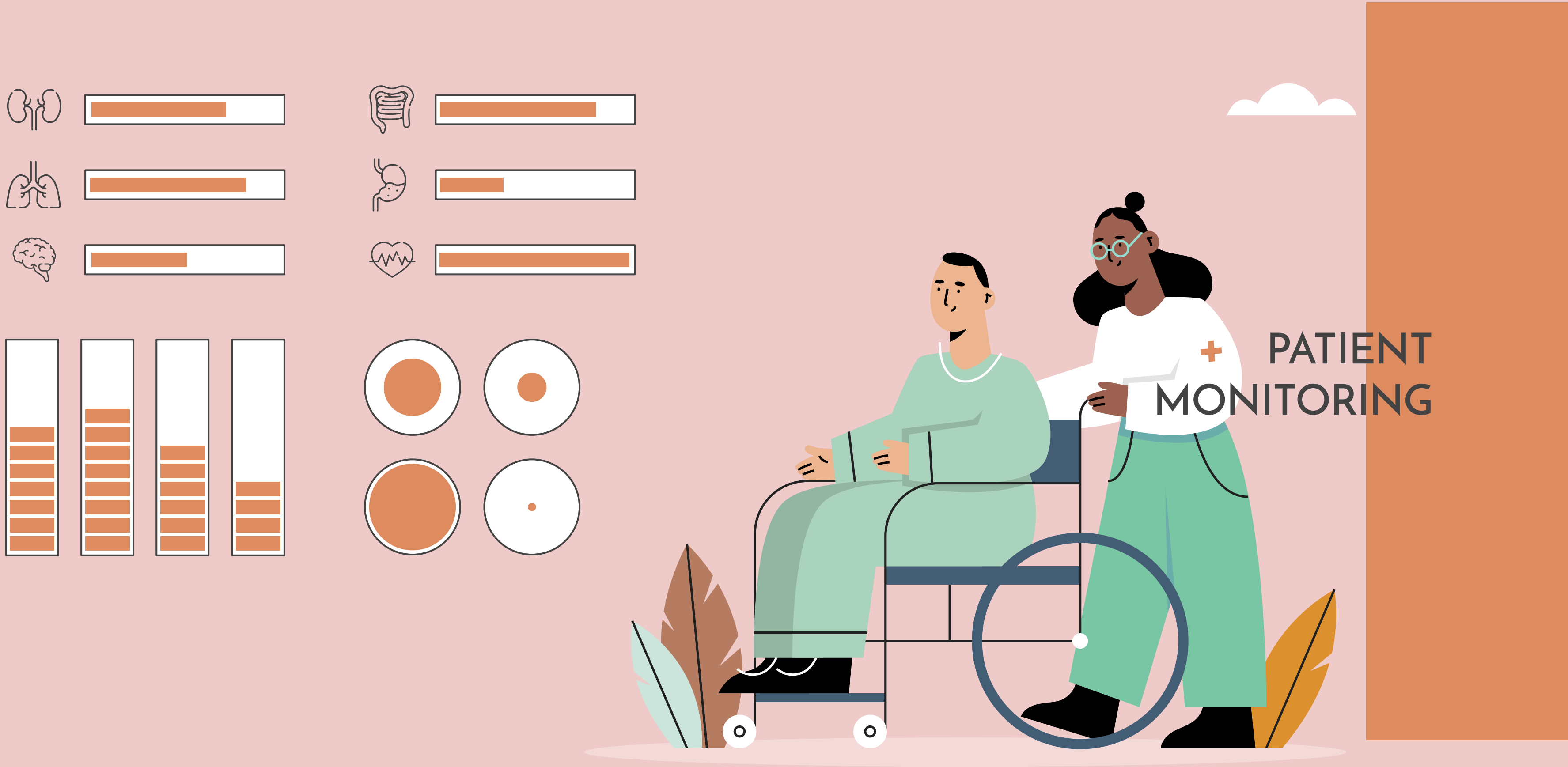
CASE 3

- True Negative Mis-classified as False Positive **(FP)**
- With Explainability, The Doctor has better insights to the wrong classification by the model in this case. He can apply domain expertise and further tests for this Patient.

EFFECT

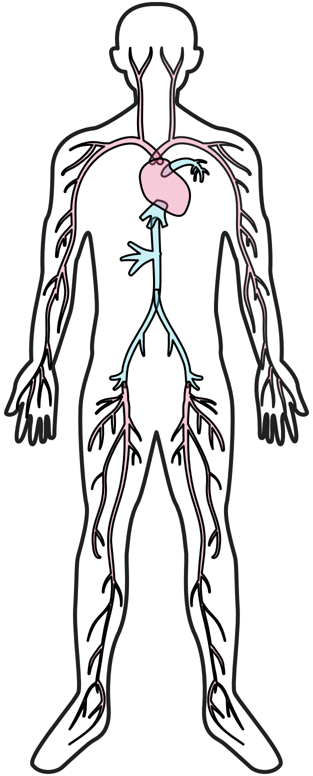
- Although no Model is perfect. In this case, it's better for the model to misclassify a Non-Diabetic as Diabetic **(FP)** rather than to misclassify a Diabetic as Non-Diabetic **(FN)**.
- The aim of an experienced Data Scientist is to reduce the reducible error as much as possible but in a medical situation like Diabetes, **FN** should never be higher than **FP**

EXPLAINABILITY can improve Diabetes detection



Improved Diagnosis | Personalised Recommendations | Confident Doctors | Happy Patients

A Few Emergent questions from EXPLAINABILITY...



False Positives

What is the cost associated with a False Positive?

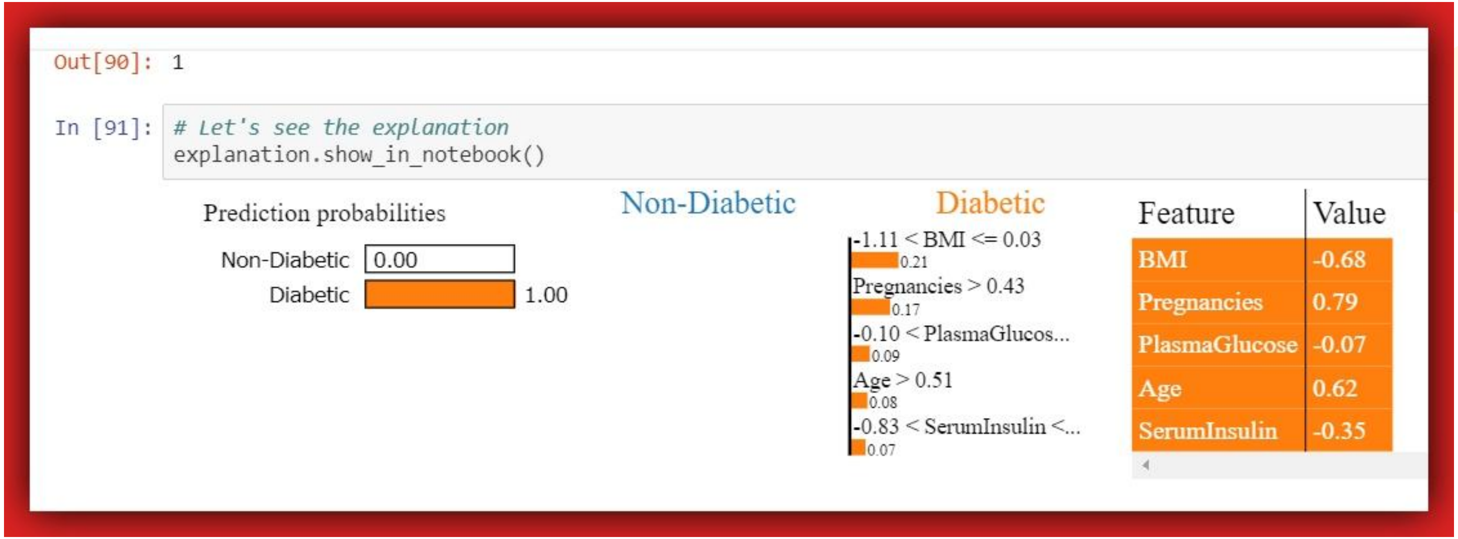
Information Gain

Where is information gained the most?

False Negatives

What is the cost associated with a False Negative?

Evaluating EXPLAINABILITY…



With the above Patient Data, we made a Prediction as seen above and applied Monotonicity and Faithfulness metrics to the explanation above.
It turns out we got 0.45 for faithfulness and False for Monotonicity.
This means the explainer Model is strongly correlated to my XGB Model, although adding more features does not necessarily translate to performance improvement.

SUMMARY

Explainability can help Doctors to proffer fine-grain therapies to Diabetic Patients based on Patient data applied to Statistical Models.

I have built a Performant Statistical Model from XGBoost Classifier module of Sklearn library and I have saved the model ready-for-production as seen in my Project document.

My model has performed significantly well by all Metrics. But as we all know no Model is perfect and Explainability is as iterative as Machine Learning. So as I gather more data, training will be done to improve performance even more.

[PROJECT LINK]

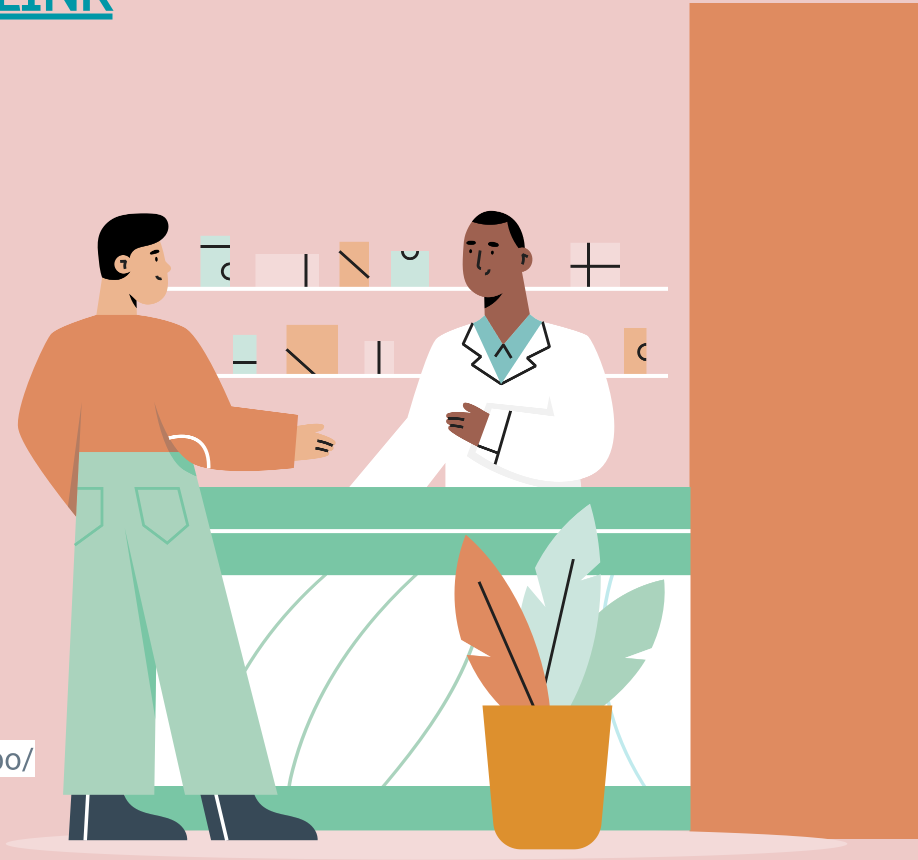
THANKS

Does anyone have any questions?

sisokels@gmail.com

@LKrukrubo

https://www.linkedin.com/in/lawrencekrukrubo/



CREDITS

Acknowledgements:

Smith, J.W., Everhart, J.E., Dickson, W.C., Knowler, W.C., & Johannes, R.S. (1988). Using the ADAP learning algorithm to forecast the onset of diabetes mellitus. In Proceedings of the Symposium on Computer Applications and Medical Care (pp. 261--265). IEEE Computer Society Press.

- Presentation template by Slidesgo
- Icons by Flaticon
- Infographics by Freepik
- Images created by Freepik
- Text & Image slide photo created by Freepik.com