# Big Data and Intelligent Analytics - Fall 2022

Final Project :  Life Insurance Assessment


TEAM 5


| NAME | NUID |
|---|---|
| SAI VENKATA SAMANTH KODURU | 2983840 |
| SHANTAN DADI | 2927718 |

# Overview:

In a one-click shopping world with on-demand everything, the life insurance application process is antiquated. Customers provide extensive information to identify risk classification and eligibility, including scheduling medical exams, a process that takes an average of 30 days.

The result? People are turned off. That's why only 40% of U.S. households own individual life insurance. Prudential wants to make it quicker and less labor intensive for new and existing customers to get a quote while maintaining privacy boundaries.

# Goals:

1.Developing a predictive model that accurately classifies risk using a more automated approach.
2.To assess the impact of certain features and their importance towards making a judgment.
3.To use different ML algorithms/build different models to predict risk response.
4.To explain their models and understand the value and accuracy of their findings.
5. To use SHAP Values and Lime for model interpretability to fathom:
    I. What drives the model prediction?
    Ii. Why did the model make a certain decision?
    iii.How can we trust model predictions?
6.To provide a simple web interface to interact with the model and observe model metrics.

# Use Cases:

Providing a faster streamline service to curtail lengthy procedures like identifying risk classification and eligibility, including scheduling medical exams, a process that takes an average of 30 days.

**Users**:
Customers classify their data to prudential in the form of age, employment status and medical history.

**Prudential**:
Employees of prudential check the risk category and analyze a particular model to check risk category and determine which factor/features have precedence on the models decision.

# Data:

**Dataset** : https://www.kaggle.com/competitions/prudential-life-insurance-assessment/data

**Description**:
In this dataset, we have been given over a hundred variables describing attributes of life insurance applicants. The task is to predict the "Response" variable for each Id in the test set. "Response" is an ordinal measure of risk that has 8 levels.

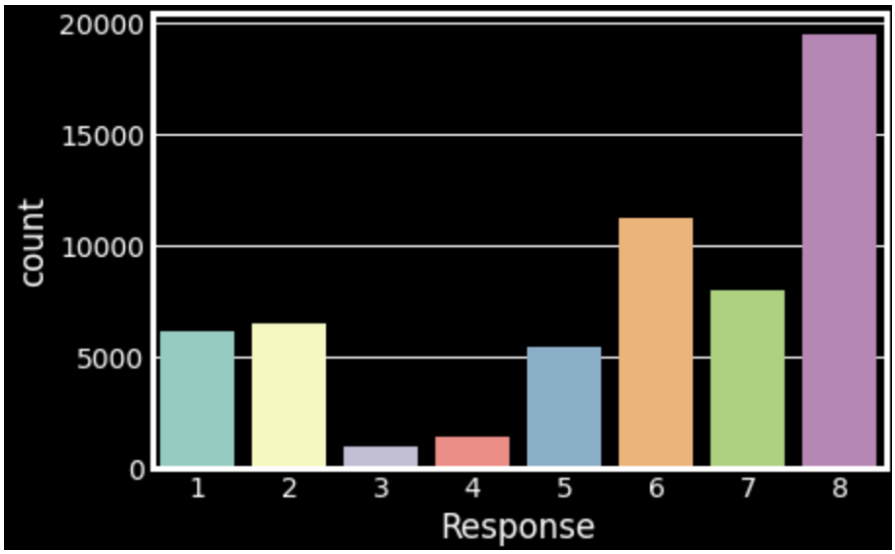
## Data Overview:

| Id | Product_Info_1 | Product_Info_2 | Product_Info_3 | Product_Info_4 | Product_Info_5 | Product_Info_6 | Product_Inf |
|---|---|---|---|---|---|---|---|
| 2 | 1 | D3 | 10 | 0.076923 | 2 | 1 | 1 |
| 5 | 1 | A1 | 26 | 0.076923 | 2 | 3 | 1 |
| 6 | 1 | E1 | 26 | 0.076923 | 2 | 3 | 1 |
| 7 | 1 | D4 | 10 | 0.487179 | 2 | 3 | 1 |
| 8 | 1 | D2 | 26 | 0.230769 | 2 | 3 | 1 |

| Product_Info_7 | Ins_Age | Ht | Wt | ... | Medical_Keyword_40 | Medical_Keyword_4 |
|---|---|---|---|---|---|---|
| 1 | 0.641791 | 0.581818 | 0.148536 | ... | 0 | 0 |
| 1 | 0.059701 | 0.600000 | 0.131799 | ... | 0 | 0 |
| 1 | 0.029851 | 0.745455 | 0.288703 | ... | 0 | 0 |
| 1 | 0.164179 | 0.672727 | 0.205021 | ... | 0 | 0 |
| 1 | 0.417910 | 0.654545 | 0.234310 | ... | 0 | 0 |

| Medical_Keyword_46 | Medical_Keyword_47 | Medical_Keyword_48 | Response |
|---|---|---|---|
| 0 | 0 | 0 | 8 |
| 0 | 0 | 0 | 4 |
| 0 | 0 | 0 | 8 |
| 0 | 0 | 0 | 8 |
| 0 | 0 | 0 | 8 |

## Response:




Response is what will be predicting, response is categorical data; primarily, we could call it the risk category.

# Process Outline:

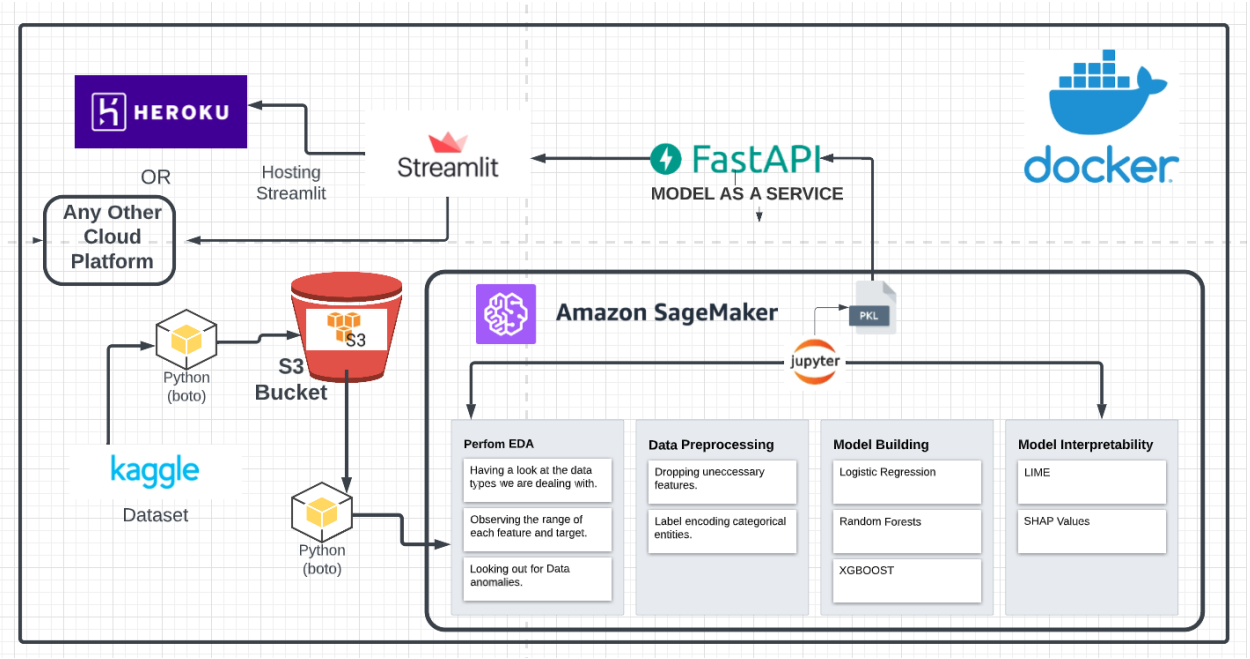
1. LOAD:
   We use BOTO to load the csv to an AWS S3 Bucket.
2. EXTRACT:
   Recurrently, we use BOTO to extract the data into a jupyter notebook.
3. Preprocessing:
   Having a peak at the data to look at data types, transforming some categorical using label encoders, filling null or n/a values, filtering out unwanted features.
4. EDA:
   Scouring useful information like the range of data.
   Normalizing data if required.
   Finding out correlations between features.
5. Modeling:
   Building multiple models like:
   i) Linear Classifiers,
   ii) Random Forests,
   iii) XGBOOST.
6. Model Interpretation:
   Using SHAP Values and Lime, also metrics such as roc, auc, logloss, f1_score; we implement:
   i) Feature Importance.
   ii) Feature Selection.
   iii) Monitor Model Metrics.
7. FAST-API:
   We instrument fast-api to fabricate a serviceable Model, by developing a Model As A Service platform and host it on a platform like Heroku or Digital Ocean.
8. Heroku/ Digital Ocean:
   Utilizing either of these cloud platforms we host the fast-api model as a service.
9. Streamlit:
   Now all we need is an interface to interact with the model, thereupon we benefit from streamlit where we could interact with the models we built, have a look at their performances by analyzing model metrics and interoperability.

# Milestones:

| | |
|---|---|
| Day 1 | LOAD and EXTRACT |
| Day 2 and 3 | Preprocessing and EDA |
| Day 4,5,6 | Modeling and Model Interpretation |
| Day 7 | Fast-api |
| Day 8 | Hosting on Heroku/Digital Ocean |
| Day 9 | Streamlit |
| Day 10 | Dockerization |

# Deployment:

1) Laguage: Python
2) Cloud: AWS S3, Heroku/Digital Ocean
3) Service: fast-api
4) Website: Streamlit
5) Container: Docker


HEROKU
OR
Any Other Cloud Platform
Hosting Streamlit
Streamlit
FastAPI
MODEL AS A SERVICE
docker
Python (boto)
S3
S3 Bucket
kaggle
Dataset
Python (boto)
Amazon SageMaker
PKL
jupyter
Perfom EDA
Having a look at the data types we are dealing with.
Observing the range of each feature and target.
Looking out for Data anomalies.
Data Preprocessing
Dropping uneccessary features.
Label encoding categorical entities.
Model Building
Logistic Regression
Random Forests
XGBOOST
Model Interpretability
LIME
SHAP Values

# References and Sources:


1.https://www.kaggle.com/competitions/prudential-life-insurance-assessment/data
2.https://shap.readthedocs.io/en/latest/
3.https://towardsdatascience.com/understanding-model-predictions-with-lime-a582fdff3a3b
4.https://github.com/digitalocean/doctl