Documentation

IVF Prediction System

By Group B:

- Firdan Rahman W.
- Galih Mahardika
- Ridha Muldina Negara

TABLE OF CONTENTS

1. Background

As humans grow in population, it is crucial for parents to have a clear view on both father's and mother's condition to be able to have a pregnancy. If the cell from both parents is not strong enough to conceive naturally, one of the things that parents could do is by IVF or what we called In Vitro Fertilization. This is by combining the sperm and egg cells that would be taken from both parents in a laboratory, which then be planted again in the mother's womb. Even then, there are many factors that would influence and lead to the successfulness of the IVF process. As such, we aim to make a prediction system using previous results of IVF on other patients based on their profile and their history, while also including the condition of the IVF process, including the storage process, time reserves, etc.

This prediction system would conclude the successfulness from doing the IVF until birth. But this does not include the patient lifestyle during the pregnancy. From this prediction result, we would inform the patients of the outcome, whether it would be successful or not. Based on this information, the patients will now not blindly try the program without any prior knowledge and they would consider if they would risk the IVF process even if the program is expensive.

2. Objectives

We are going to create the prediction system using a machine learning algorithm that uses data from previous patients, including their condition and the successfulness of the program. The prediction system will input the current patient's profile and condition of the process. The output would result in whether the parents would have a successful birth or not.

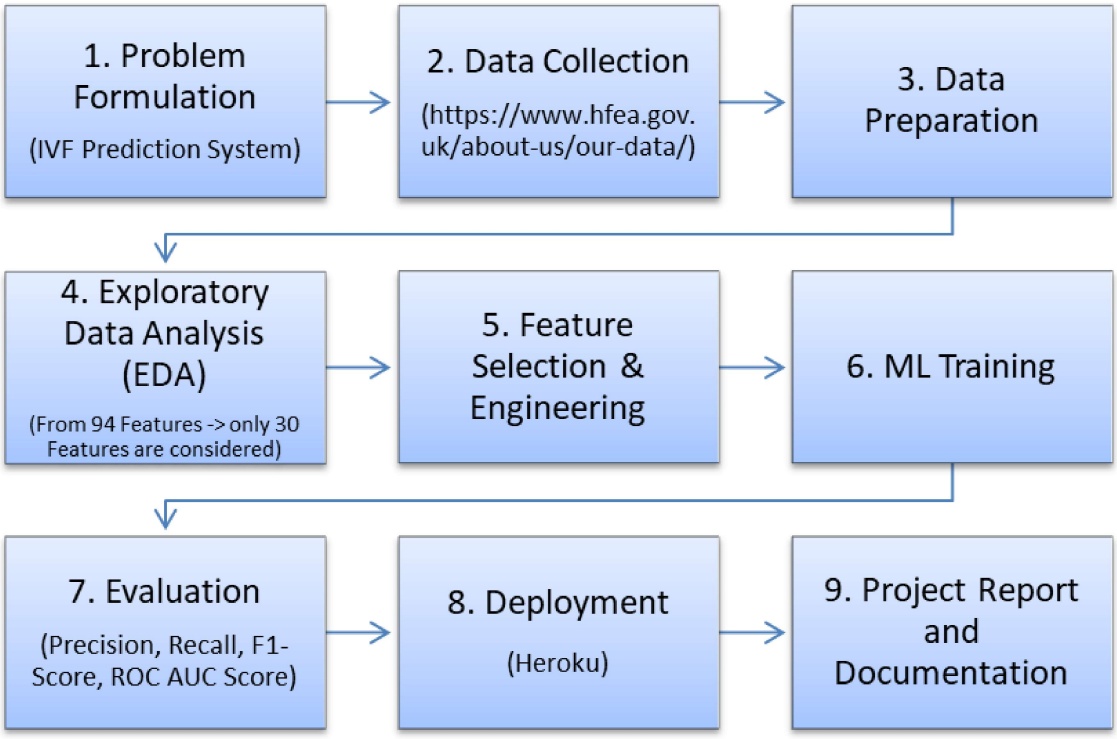
The success would be if the trained model prediction reaches an ROC-AUC score of 75%, and is scored highest compared to other models. The use of ROC-AUC Score is to rate the performance of the model based on the ability to distinguish between the predicted positive and negative compared to the actual value. It would take the patient's profile and condition of the process to ML metrics. The only metrics that we could control would be the condition of taking and storing both of parent's cells in a lab, before it would be taken into the womb. Based on the metrics/features that can be controlled, the hospital or laboratory which would carry out this program could understand and maximize the facility or treatment that would heavily impact the successfulness of this program. As such, it would be beneficial for both the parents' financial and the hospital/lab credibility.

3. Overview

3.1 Workflow

In this project we will use previous patients' dataset, obtained from Human Fertilization and Embryology Authority (HFEA) which is not confidential, and consist of 94 features, we will build the model (based on classification) using various Machine Learning algorithm such as Logistic Regression, K-nearest neighbors, random forest, decision tree, and support vector machine, with an additional XGBoost Classifier. We would compare those trained models and choose the best model with the best ROC-AUC Score.

The flow of this project is as follows:



Pic 1. Workflow of the project

Based on the workflow above, the data is first collected from the HFEA, as its data is publicly available. The data obtained were data from 2010-2014 and 2015-2016. After the data is gathered and collected, it is then considered that data from 2010-2014 would be the training data, as it have the most data with longer time span, while the 2015-2016 is for the validation data, as it will be the data to show which model would be best

Quick overlook at the data

```
df.head(10)
```

	Patient Age at Treatment	Date patient started trying to become pregnant OR date of last pregnancy	Total Number of Previous cycles, Both IVF and DI	Total Number of Previous treatments, Both IVF and DI at clinic	Total Number of Previous IVF cycles	Total Number of Previous DI cycles	Total number of previous pregnancies, Both IVF and DI	Total number of IVF pregnancies	Total number of DI pregnancies	Total number of live births - conceived through IVF or DI	...	Heart Three Birth Weight	Heart Three Sex	Heart Three Delivery Date	Heart Three Birth Congenital Abnormalities	Heart Four Weeks Gestation	Heart Four Birth Outcome	Heart Four Birth Weight	Heart Four Sex	Heart Four Delivery Date	Heart Four Birth Congenital Abnormalities
0	18 - 34	NaN	1	1	0	1	0	0	0	0	...	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
1	35-37	NaN	0	0	0	0	0	0	0	0	...	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
2	18 - 34	NaN	0	0	0	0	0	0	0	0	...	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
3	38-39	NaN	1	1	0	1	0	0	0	0	...	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
4	35-37	NaN	0	0	0	0	0	0	0	0	...	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
5	18 - 34	NaN	1	1	0	1	0	0	0	0	...	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
6	38-39	NaN	0	0	0	0	0	0	0	0	...	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
7	38-39	NaN	4	4	0	4	0	0	0	0	...	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
8	35-37	NaN	4	4	0	4	1	0	1	1	...	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
9	18 - 34	NaN	2	1	1	1	1	1	0	1	...	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN

10 rows × 95 columns

```
[ ] df.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 337109 entries, 0 to 337108
Data columns (total 95 columns):
```

Pic 2. Quick Overlook of Data

From the quick overlook of data, there are 95 features available on the training dataset. There are too many features to predict and it may not be beneficial to include all features into our model. Considering we already see there is an abundance of NaN values or missing values on the dataset, we then look at how many rows of data are missing on each feature, based on percentage.

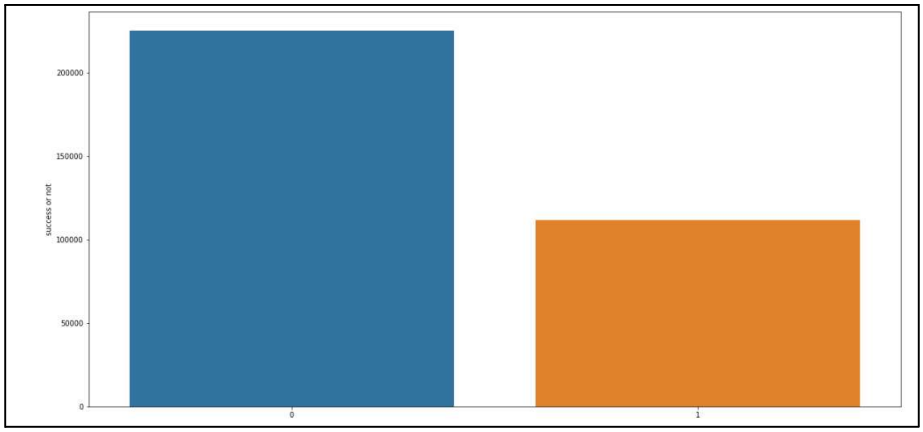
	Missing Ratio
Heart Four Birth Congenital Abnormalities	100.000000
Heart Three Birth Congenital Abnormalities	100.000000
Heart Two Birth Congenital Abnormalities	100.000000
Heart One Birth Congenital Abnormalities	100.000000
Heart Four Birth Weight	99.999407
Heart Four Weeks Gestation	99.999407
Heart Four Sex	99.999407
Heart Four Delivery Date	99.999407
Heart Four Birth Outcome	99.997924
Heart Three Birth Weight	99.890836

Pic 3. Missing Ratio

Based on the missing ratio, there is a lot of missing data on almost half the features, we then limit the missing data with a missing ratio maximum of 25%, decided by what we thought would be conservatives. From the limitation of missing ratio, we gathered about 59 features that would proceed into the feature engineering and selection.

3.2 Target Value

First of all, we select the column of features that would contribute to our **target value** on our model. There are a total of 4 features on the training, such as Total number of live births - conceived through IVF or DI, Total number of live births - conceived through IVF, Total number of live births - conceived through DI, and Number of Live Births. We would combine 4 of those features into target value by combining them and set the data to be binary, 0 for unsuccessful and 1 for successful birth. First we engineer the data of numerical features such as total number of live births, as to 0 for no live birth, and 1 for any value of live birth above 1, that indicates a history of successful birth. After we engineered 4 of those features, we would then create a new column called “Successful or Not”, that is filled with 0 if there are no recorded live births on 4 of those features, and 1 if there is at least 1 successful live birth from each of the previously stated 4 columns.



Pic 4. Compared Data Target Value

This data is then saved to target value. Based on the picture above, the data is a bit imbalanced, as successful birth is only a half of the unsuccessful, but still could be acceptable.

3.3 Dataset Features

3.3.1 Feature Engineering

Based on the previous 59 columns we gathered from the missing ratio selection, we still try to eliminate the features and also try to change the data it contains, the value or even the data types, to be able to fit on the model we proposed.

In the feature of "Patient Age at Treatment," there are patient records containing 999 that are assumed to be >50 years. This data is then converted into categorical data, such as

18–34 is converted to 0

35–37 is converted to 1

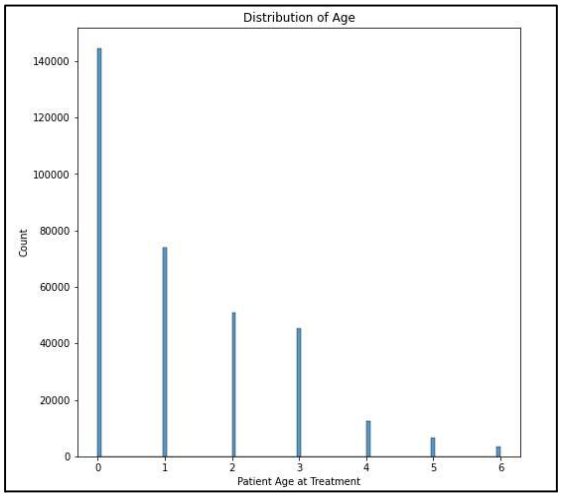
38–39 is converted to 2

40–42 is converted to 3

43–44 is converted to 4

45–50 is converted to 5

999 is converted to 6



Pic 5. Age distribution

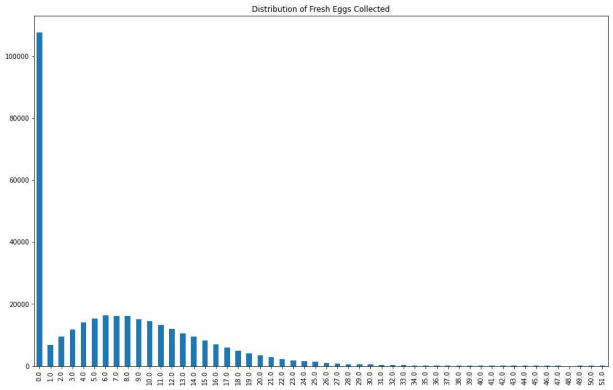
As seen above, the age distribution is skewed to the categorical data of 0, with the age of 18-34, for previous IVF treatment. One of the reasons is that during that age, it is the most productive years in which people would marry and have children, while also the gap between the years is also higher than others.

For other features, such as Total Number of Previous treatments, Both IVF and DI at clinic, Treat Total Number of Previous IVF cycles, Treat Total Number of Previous DI cycles, etc. We would convert the text data into categorical data by replacing the text into numbers to be included as categorical type, such as converting text “>=5” into number 6.

For the Treat Total Main Reason for Producing Embroys Storing Eggs feature, due to too many variables inside the feature, we would cut almost all variables that contain some of the main information, and would be left with 4 data variables.

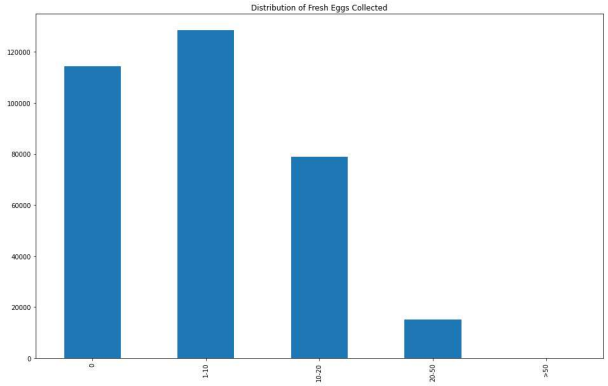
Treat Total Type of treatment - IVF or DI, Treat Specific treatment type, etc. is classified by converting text into categorical data. For Date of Embryo Transfer & Year of Treatment, we would drop both of them as it is not considered to affect the model, as it is just a time information.

While for Treat Fresh Eggs Collected, Treat Fresh Eggs Stored, etc., we were trying to engineer the data from numerical value into categorical by binning the data, based on the distribution of data it gave. Such as follows:



Pic 6. Fresh Eggs Collected Feature

Based on the information above, the best way to convert is by dividing data into 5 categorized data, such as '0', '1-10','10-20','20-50','>50'.

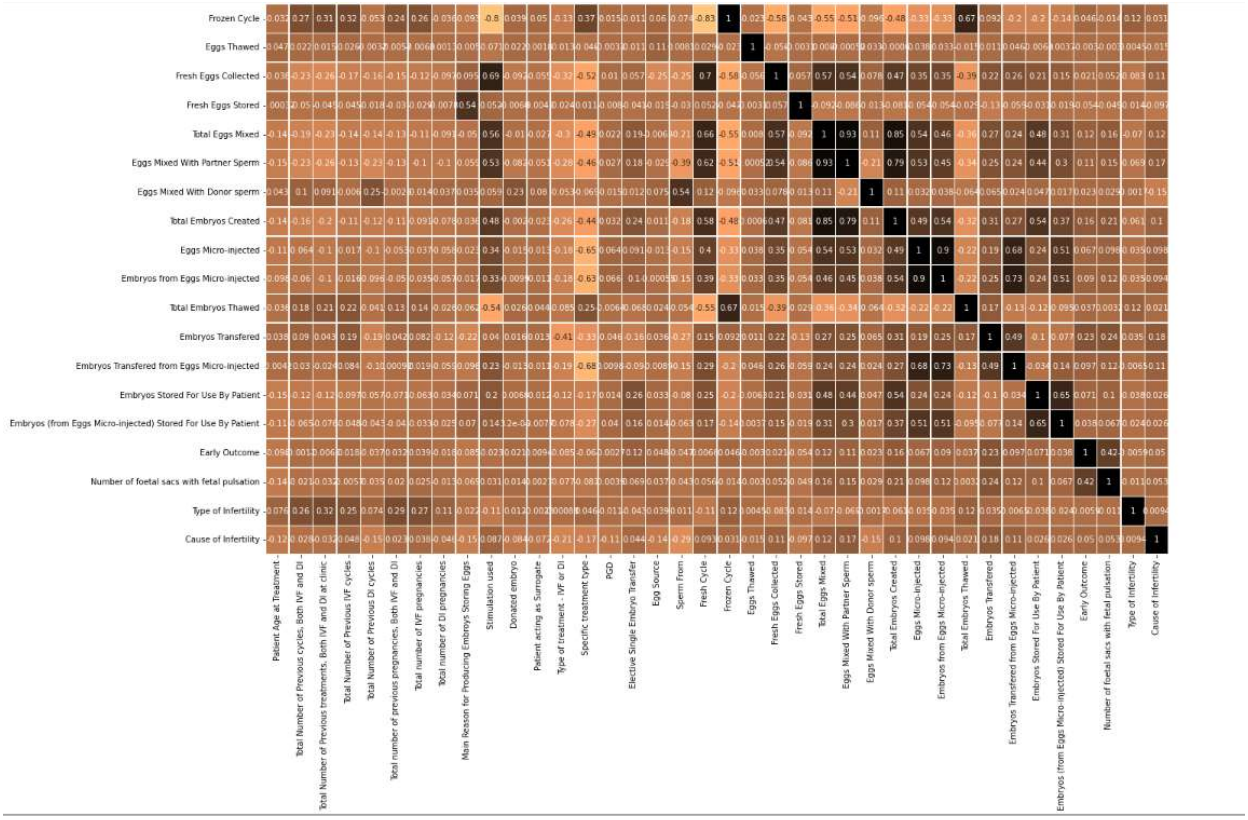


Pic 7. Divided Fresh Eggs Collected Feature

From the transformation, we could see the distribution from the categorized data to be more compact and informative. The divided data then converted into categorical data, e.g. 10-20 into number 2, and 10-20 into number 3.

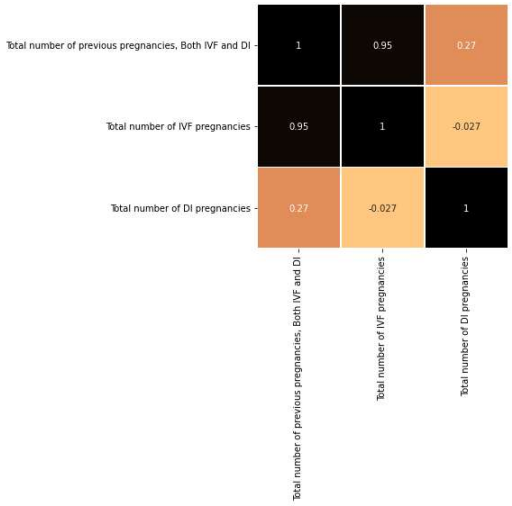
As for the **Cause of Infertility** and **Type of Infertility,** from common knowledge, it may contain valuable information for the success of the program, and it may be better if we included all variables into the training model. So each text variable of **Cause of Infertility** and **Type of Infertility** is converted into categorical data.

3.3.2 Advanced Feature Selection



Pic 8. A Slice of Features Correlation

Then we proceed to do the advanced feature selection, to further eliminate unnecessary features. The unnecessary features are caused by the high correlation and information from other features, and this would affect the fitting of the training model. We would take an example of feature between Total number of previous pregnancies, Both IVF and DI to Total number of IVF pregnancies & Total number of DI pregnancies. It is shown below:



Pic 9. Correlation between Some features

The total number of previous pregnancies, both IVF and DI is heavily affected by the total number of IVF pregnancies, and as it also contains the total number of DI Pregnancies. So we could remove The total number of previous pregnancies, both IVF and DI columns so it is not redundant. This applies to other features that correlate with each other.

Special Case is for some features that may affect the target value. The number of foetal sacs with fetal pulsation is heavily correlated to the success or not target variable, so we will drop the feature. Early outcome feature is also removed as it is also contains the fetal pulsation information. Both of these features would overfit the training data and may cause a bias to our final model.

3.3 Dataset used

The final dataset features to be used on our model, as previously have been through the engineering and converting process, is as follows:

```
Index(['Patient Age at Treatment', 'Total Number of Previous IVF cycles',
       'Total Number of Previous DI cycles', 'Total number of IVF pregnancies',
       'Total number of DI pregnancies',
       'Main Reason for Producing Embroys Storing Eggs', 'Donated embryo',
       'Patient acting as Surrogate', 'Type of treatment - IVF or DI',
       'Specific treatment type', 'PGD', 'Elective Single Embryo Transfer',
       'Egg Source', 'Sperm From', 'Fresh Cycle', 'Frozen Cycle',
       'Eggs Thawed', 'Fresh Eggs Collected', 'Fresh Eggs Stored',
       'Eggs Mixed With Partner Sperm', 'Eggs Mixed With Donor sperm',
       'Total Embryos Created', 'Eggs Micro-injected', 'Total Embryos Thawed',
       'Embryos Transfered', 'Embryos Transfered from Eggs Micro-injected',
       'Embryos Stored For Use By Patient', 'Type of Infertility',
       'Cause of Infertility'],
      dtype='object')
```

Pic 10. Features to be used on Model

It would then be split into training data and testing data, with a proportion of 80% training, and 20% testing data.

3.4 Modeling and ML Training

First and foremost, we would prepare the validation data from 2015-2016. We then Feature engineered and selected the data based on the treatment we had on the Training and Testing dataset.

For Modeling, the list of models used are Logistic Regression, K-nearest neighbor, Decision Tree, Random Forest Tree, Support Vector machine, and XGBoost Classifier. It is also considered using cross validations and a list of hyperparameters to be tested and finding out the best parameters to be used, using the help of GridSearchCV and RandomizedSearchCV, of which the parameters are selected based on the ROC-AUC score it produces from fitting of training data to the testing data.

One of the examples is from XGBoost Classifier, the setting of parameters can be shown below:

Setting parameter

```
[ ]  fit_params={"early_stopping_rounds":100,
                 "eval_metric":'auc',
                 "eval_set":[(X_test,y_test)],
                 "verbose":True
                }

     param_dist = {'n_estimators': [k for k in range(300,1000,100)],
                   'learning_rate': np.linspace(0.01,0.2,5),
                   'subsample': np.linspace(0.1,1,5),
                   'max_depth': np.arange(3,10,2),
                   'colsample_bytree': np.linspace(0.1,1,10)
                  }
```

Pic 11. Hyperparameters on XGBoost to be tested

The cross validation method is using RepeatedStratifiedKFold with 5 splits and 2 repeats. When it was fitted, it shows the best hyperparameters as follows:

```
Tuned hyperparameters : {'subsample': 1.0, 'n_estimators': 500, 'max_depth': 3, 'learning_rate': 0.15250000000000002, 'colsample_bytree': 0.8}
Best ROC_AUC score: 0.8158541174920891
Best Estimator: XGBClassifier(base_score=0.5, booster='gbtree', callbacks=None,
              colsample_bylevel=1, colsample_bynode=1, colsample_bytree=0.8,
              early_stopping_rounds=None, enable_categorical=False,
              eval_metric=None, gamma=0, gpu_id=-1, grow_policy='depthwise',
              importance_type=None, interaction_constraints='',
              learning_rate=0.15250000000000002, max_bin=256,
              max_cat_to_onehot=4, max_delta_step=0, max_depth=3, max_leaves=0,
              min_child_weight=1, missing=nan, monotone_constraints='()',
              n_estimators=500, n_jobs=0, num_parallel_tree=1, predictor='auto',
              random_state=0, reg_alpha=0, reg_lambda=1, ...)
```

Pic 12. Best Hyperparameters on XGBoost

3.5 Model Result

The final ROC AUC Scores all of the models which has been fitted into training data, while the hyperparameter is fitted from the testing data, when it is applied on the validation data is as follows:

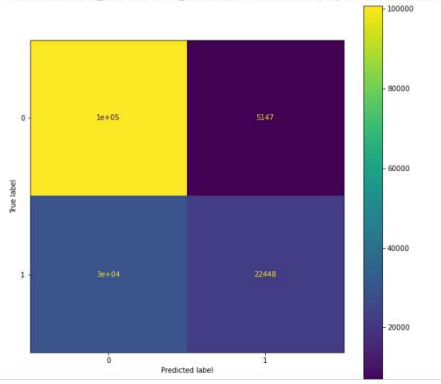
	Logistic Regression	K Nearest Neighbor	Decision Tree	Random Forest	Support Vector Machine	XGBoost Classifier
ROC AUC SCORE	0.800903	0.691247	0.754222	0.789997	0.773984	0.814418

Pic 13. Score of Models

Based on the data above, it is then selected that the best model for predicting is from XGBoost Classifier. From then, we show the classification test and f1 score test of the model on validation data.

```
#Testing
report = classification_report(y_valid, clf_xgb.predict(X_valid_float), target_names=['Success', 'Not Success'])

print(report)

plot_confusion_matrix(clf_xgb, X_valid_float, y_valid)
```

```
              precision    recall  f1-score   support

     Success       0.77      0.95      0.85    105841
 Not Success       0.81      0.43      0.56     52678

    accuracy                           0.78    158519
   macro avg       0.79      0.69      0.70    158519
weighted avg       0.78      0.78      0.75    158519

<sklearn.metrics._plot.confusion_matrix.ConfusionMatrixDisplay at 0x2238eaa8be0>
```



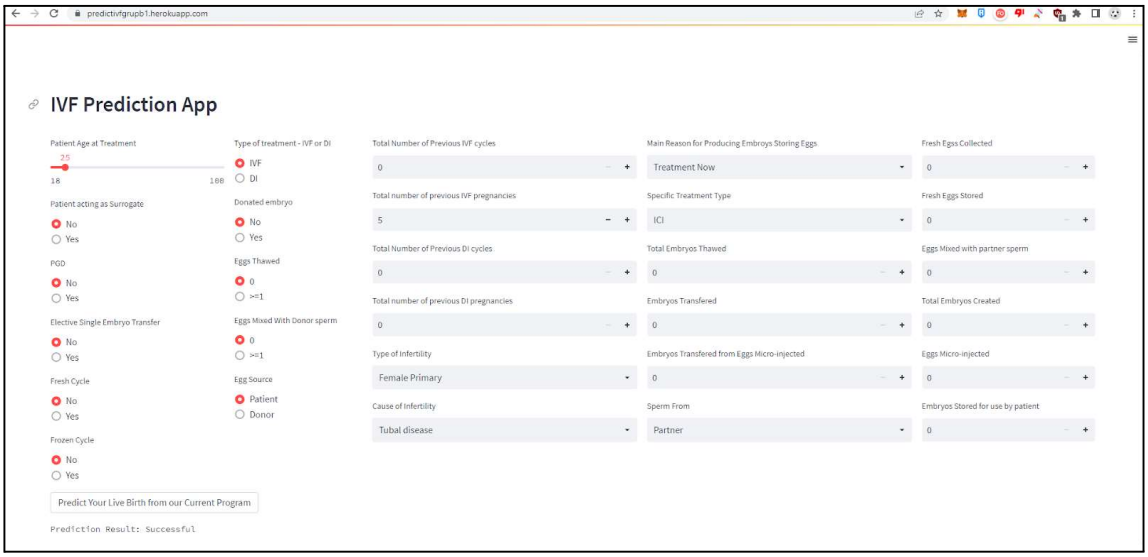
Pic 14. Classification report data of XGBoost to Validation Data

From the result data, the recall value of success is as high as 0.95 while the recall for not success is 0.43, it would show that the model has poor ability to predict the not success of validation data. It is also applied on any other models, so it is still accepted. The model is then saved to be used on our prediction app for the user.

4. Model Demonstration

The deployed app is based on streamlit, an open source API framework, which is useful for displaying the input it needed to process while also showing the result in one page. Streamlit could also run the model saved on the file after it has been trained. It is then deployed on heroku, a cloud platform to host the page of which the streamlit can run, as it supports the programming of the model we deployed.

The prediction app can be used here https://predictivfgrupb1.herokuapp.com/, and the image overview of the page can be seen below.



Pic 15. Deployed App on Heroku

The prediction app needs the information of patient's profile history, and also the method it is going to go through, the full list of data could be shown below:

Table 1. Data Input

Patient Age at Treatment	Age of the patient, by slider
Patient acting as Surrogate	Patient acting as a surrogate, Yes or No
PGD	Pre-implantation Genetic Diagnosis, Yes or No
Elective Single Embryo Transfer	Elective Single Embryo Transfer, Yes or No
Fresh Cycle	Using fresh embryos, Yes or No
Frozen Cycle	Using frozen embryos, Yes or No

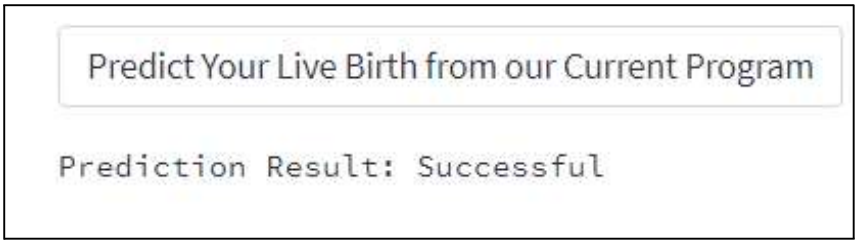
Type of treatment - IVF or DI	IVF if using IVF treatment, DI if using DI Treatment
Donated embryo	Using donated embryo for program, Yes or No
Eggs Thawed	If this cycle frozen eggs, the number of eggs thawed
Eggs Mixed With Donor sperm	Patient's Eggs mixed with donor patient
Egg Source	Egg source, Patient or Donor
Total Number of Previous IVF cycles	Total number patient previously had IVF Treatment
Total number of previous IVF pregnancies	Total number patient previously had IVF Pregnancies
Total Number of Previous DI cycles	Total number patient previously had DI Treatment
Total number of previous DI pregnancies	Total number patient previously had DI Pregnancies
Type of Infertility	Type of infertility patients, list of input
Cause of Infertility	Cause of infertility, Primary cause of the infertility, list of input
Main Reason for Producing Embroys Storing Eggs	Reason for producing embryos storing eggs, list of input
Specific Treatment Type	Specific treatment patient will go through, list of input
Total Embryos Thawed	Total embryos thawed
Embryos Transferred	Number of embryos transferred into the patient receiver
Embryos Transferred from Eggs Micro-injected	Reason for producing embryos injected, List of input
Sperm From	Patient's Sperm origin for this cycle, Patient or Donor
Fresh Eggs Collected	Number of fresh eggs collected

Fresh Eggs Stored	Number of fresh eggs stored
Eggs Mixed with partner sperm	Total numbers of Eggs mixed with partner sperm
Total Embryos Created	Total embryos created
Eggs Micro-injected	Total Eggs micro-injected
Embryos Stored for use by patient	Total embryos stored to use by patient

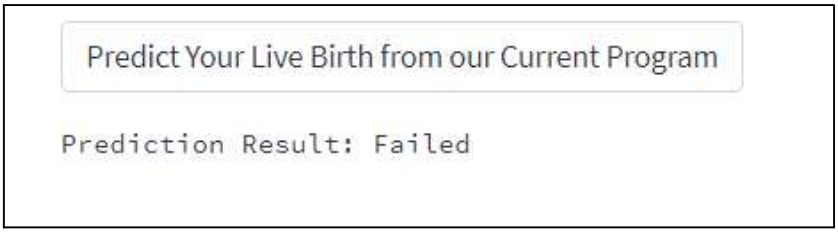
Input from the total list above is already bounded by the ability of the streamlit to limit what kind of data input it can receive, e.g. Patient's Age of treatment is limited by slider from 0 to 100, while for the number input, e.g. Total number of Previous IVF cycles, the only way a user can input the data is by typing an integer number, typing the fraction/float number would be removed and only the leading digits remain, while also trying to input any string would not be possible, as it would not register anything on the app. All of these features are due to streamlit programming.

The outcome of this model would show whether the result would be successful or failed, the display of this would be as follows:

1. If the prediction is successful, it will prompt:



2. While if it failed, it will prompt:



5. Conclusion

5.1 Result

From the previous history data gathered from Human Fertilization and Embryology Authority (HFEA), we have trained a list of models using machine learning algorithms to predict the outcome of IVF program, categorized as binary classifier problem, based on the patient's history and the preparations of lab/hospital took before inducing the program. This program would help the patient's decision to further continue the program or not and the lab/hospital to maximize the method it is going to take for increasing the chance of success.

List of models used are Logistic Regression, K-nearest neighbor, Decision Tree, Random Forest Tree, Support Vector machine, and XGBoost Classifier. It is also considered using cross validations and a list of parameters to be tested and finding out the best parameters to be used, using the help of GridSearchCV and RandomizedSearchCV, based on the ROC-AUC score it produces.

Based on this scoring, the best model of this is XGBoost Classifier, with an ROC-AUC Score above 0.8, highest compared to other models as it optimizes using the gradient descent and boosting algorithm. As such, the prediction app is made based on the saved XGBoost Classifier trained model.

The predicted app can be used on https://predictivfgrupb1.herokuapp.com/.Input the parameters based on the patients and the program decision, and the result could be viewed directly on the page.

5.2 Notes

Further improvement can be made for the prediction system regarding IVF program is by collecting the patient's lifestyle habits and history of medical data other than infertility, as it also affects the possibility of success using the IVF program, as it is one of the factors that should be heavily impact the possibility of pregnancies. Detailed importance of parameters could be further studied by the medical expertise to really explain how it affects the outcome. The balance dataset could also be better to improve the low recall of unsuccessful outcomes.

6. Reference

1. Scientific Report: “Machine learning predicts live-birth occurrence before in-vitro fertilization treatment”, “AshishGoyal, Maheshwar Kuchana & Kameswari Prasada Rao Ayyagari”, https://www.nature.com/scientificreports

2. “Multifactor Prediction of Embryo Transfer Outcomes Based on a Machine Learning Algorithm”, “Ran Liu, Shun Bai, Xiaohua Jiang, Lihua Luo, Xianhong Tong, Shengxia Zheng, Ying Wang and Bo Xu”.https://www.frontiersin.org/articles/10.3389/fendo.2021.745039/full

3. Human Fertilisation & Embryology Authority, Anonymised register data for 2010 - 2014 and Anonymised register data for 2015 - 2016. https://www.hfea.gov.uk/about-us/our-data/

APPENDIX 1

Artifacts	**Where to check?**
Dataset Original	https://www.hfea.gov.uk/about-us/our-data/ There are 94 features in the dataset
Dataset Final	Training Data: https://drive.google.com/file/d/1NIC6InbFTbIBPdWI8OwhUHjxA9rJAERN/view?usp=sharing Testing Data: https://drive.google.com/file/d/1t3Zt6DUY3x6aSdzBGX2vqTRehM41ekpV/view?usp=sharing Consist of 29 features and 1 target value feature
Project Milestone	https://github.com/FearDawn/PredictionIVFGrupB
App	https://predictivfgrupb1.herokuapp.com/
Reference	1. Scientific Report: "Machine learning predicts live-birth occurrence before in-vitro fertilization treatment", "AshishGoyal, Maheshwar Kuchana & Kameswari Prasada RaoAyyagari", https://www.nature.com/scientificreports 2. "Multifactor Prediction of Embryo Transfer Outcomes Based on a Machine Learning Algorithm", "Ran Liu, Shun Bai, Xiaohua Jiang, Lihua Luo, Xianhong Tong, Shengxia Zheng, Ying Wang and Bo Xu".https://www.frontiersin.org/articles/745039