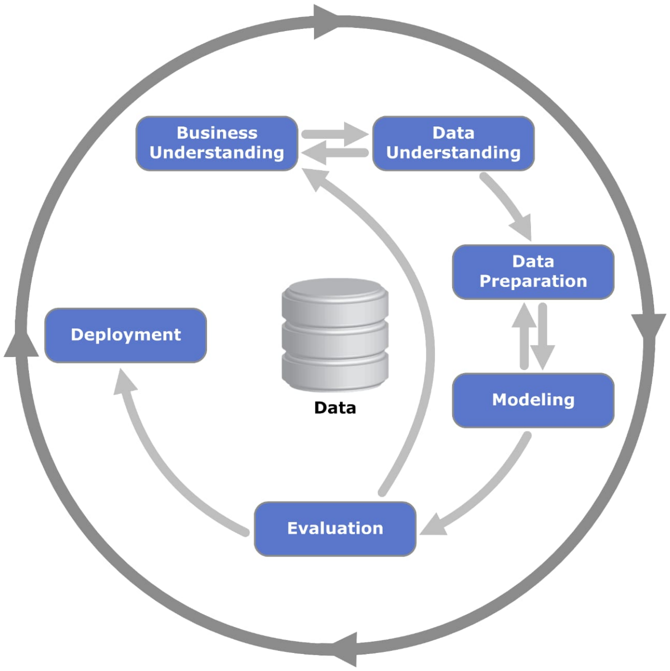


Figure 3.1: CRISP-DM Process

plans, and executes data mining (machine learning) operations (Njeru (2022)). It is a process model that outlines the normal project phases, tasks connected to each phase, and relationships between these tasks while also providing an overview of the data mining life cycle. The technique is used to conceive a data mining project and consists of six successive steps. Depending on the requirements of the developers, iterations might be introduced. The phases are as follows and as depicted by the figure above:

1. Business Understanding – What does the business need?

2. Data Understanding – What data do we have / need? Is it clean?

3. Data Preparation – How do we organise the data for modelling?

4. Modelling – What modelling techniques should we apply?

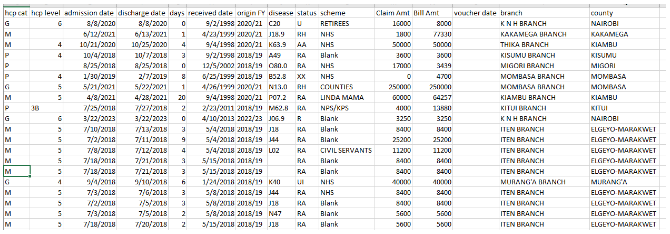
5. Evaluation – Which model best meets the business objectives?

6. Deployment – How are the results accessed?

3.2.1 Business Understanding

We employed primary and secondary sources in this phase to comprehend the issue of fraudulent healthcare insurance claims. For secondary sources, we used journals and research papers on machine learning with a focus on detecting insurance claims fraud. We concentrated on the study's goals and objectives that are related to fraudulent activity committed in healthcare insurance claims. This made it easier to choose the best methods for conducting the research after gathering data. We obtained a healthcare insurance claims data request form from the Statistics and Actuarial Science department and followed through the process at NHIF headquarters to acquire data for our research. In the area of healthcare insurance, there has been an upsurge in fraudulent insurance claims which has resulted in huge financial losses. Therefore, a system that can identify fraudulent insurance claims in real time is needed for the healthcare insurance sector.

3.2.2 Data Understanding



hcp cat	hcp level	admission date	discharge date	days	received date	origin FY	disease	status	scheme	Claim Amt	Bill Amt	voucher date	branch	county
G	6	8/8/2020	8/8/2020	0	9/2/1998	2020/21	C20	U	RETIREES	16000	8000		K N H BRANCH	NAIROBI
M		6/12/2021	6/13/2021	1	4/23/1999	2020/21	J18.9	RH	NHS	1800	77330		KAKAMEGA BRANCH	KAKAMEGA
M	4	10/21/2020	10/25/2020	4	9/4/1998	2020/21	K63.9	AA	NHS	50000	50000		THIKA BRANCH	KIAMBU
P	4	10/4/2018	10/7/2018	3	9/2/1998	2018/19	A49	RA	Blank	3600	3600		KISUMU BRANCH	KISUMU
P		8/25/2018	8/25/2018	0	12/5/2002	2018/19	O80.0	RA	NHS	17000	3439		MIGORI BRANCH	MIGORI
P	4	1/30/2019	2/7/2019	8	6/25/1999	2018/19	B52.8	XX	NHS	0	4700		MOMBASA BRANCH	MOMBASA
G	5	5/21/2021	5/22/2021	1	4/26/1999	2020/21	N13.0	RH	COUNTIES	250000	250000		MOMBASA BRANCH	MOMBASA
M	5	4/8/2021	4/28/2021	20	9/4/1998	2020/21	P07.2	RA	LINDA MAMA	60000	64257		KIAMBU BRANCH	KIAMBU
P	3B	7/25/2018	7/27/2018	2	2/23/2011	2018/19	M62.8	RA	NPS/KPS	4000	13880		KITUI BRANCH	KITUI
G	6	3/22/2023	3/22/2023	0	4/10/2013	2022/23	J06.9	R	Blank	3250	3250		K N H BRANCH	NAIROBI
M	5	7/10/2018	7/13/2018	3	5/4/2018	2018/19	J18	RA	Blank	8400	8400		ITEN BRANCH	ELGEYO-MARAKWET
M	5	7/2/2018	7/11/2018	9	5/4/2018	2018/19	J44	RA	Blank	25200	25200		ITEN BRANCH	ELGEYO-MARAKWET
M	5	7/8/2018	7/12/2018	4	5/4/2018	2018/19	L02	RA	CIVIL SERVANTS	11200	11200		ITEN BRANCH	ELGEYO-MARAKWET
M	5	7/18/2018	7/21/2018	3	5/15/2018	2018/19		RA	Blank	8400	8400		ITEN BRANCH	ELGEYO-MARAKWET
M	5	7/18/2018	7/21/2018	3	5/15/2018	2018/19		RA	Blank	8400	8400		ITEN BRANCH	ELGEYO-MARAKWET
G	4	9/4/2018	9/10/2018	6	1/24/2018	2018/19	K40	UI	NHS	40000	40000		MURANG'A BRANCH	MURANG'A
M	5	7/3/2018	7/6/2018	3	5/8/2018	2018/19	J44	RA	NHS	8400	8400		ITEN BRANCH	ELGEYO-MARAKWET
M	5	7/2/2018	7/5/2018	3	5/8/2018	2018/19	J18	RA	Blank	8400	8400		ITEN BRANCH	ELGEYO-MARAKWET
M	5	7/3/2018	7/5/2018	2	5/8/2018	2018/19	N47	RA	Blank	5600	5600		ITEN BRANCH	ELGEYO-MARAKWET
M	5	7/18/2018	7/20/2018	2	5/15/2018	2018/19	J18	RA	Blank	5600	5600		ITEN BRANCH	ELGEYO-MARAKWET

Figure 3.2: Healthcare Insurance claims data

In this step, we started by obtaining relevant data for the study,then familiarised ourselves with the data, assessed its quality, gained a fundamental understanding of the data, and extracted relevant features from the dataset to aid in the model construction.Figure(3.2) above contains a snippet of the data that was utilised in the study.

3.2.3 Data Preparation

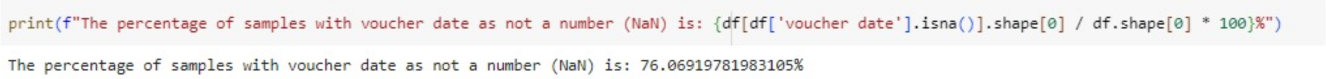
The dataset acquired was in raw format thus, pre-processing was required to generate high-quality features that would be used to train and test the ML classifiers. Data pre-processing is a crucial step for machine learning to produce accurate and insightful results. The reliability of the outcomes is correlated with data quality. Real-world datasets are imperfect, inconsistent, and noisy in nature. Data pre-processing improves the data quality by addressing the gaps in the data, reducing noise, and addressing inconsistencies. Data preparation, according to(Varshney et al. (2022)), entails cleaning, integrating, transforming, and reducing data to eliminate any duplicate or irrelevant data, leaving just the bits that provide valuable information to aid in establishing an efficient and effective classification. The stages in the procedure are as follows:

i. Data cleaning which aims to identify and impute missing values in the dataset.

ii. Application of data transformation techniques like normalization. For instance, normalization may increase the precision and effectiveness of distance-based mining algorithms.

3.2.4 Data Cleaning

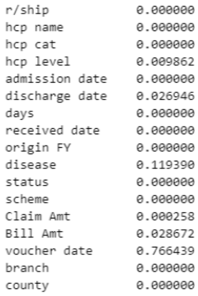
3.2.4.1 Handling missing values



```
print(f"The percentage of samples with voucher date as not a number (NaN) is: {df[df['voucher date'].isna()].shape[0] / df.shape[0] * 100}%")

The percentage of samples with voucher date as not a number (NaN) is: 76.06919781983105%
```

Figure 3.3: Checking the percentage of null values per feature

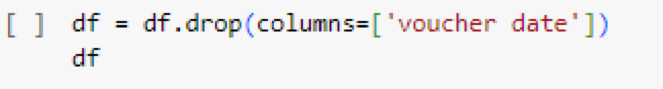


```
r/ship            0.000000
hcp name          0.000000
hcp cat           0.000000
hcp level         0.009862
admission date    0.000000
discharge date    0.026946
days              0.000000
received date     0.000000
origin FY         0.000000
disease           0.119390
status            0.000000
scheme            0.000000
Claim Amt         0.000258
Bill Amt          0.028672
voucher date      0.766439
branch            0.000000
county            0.000000
```

Figure 3.4: Percentage of null values per feature.

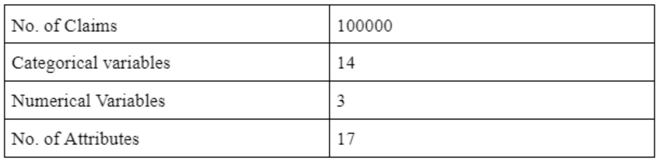
The data preparation process started by checking for missing values as shown in figure (3.3). Several input variables had null values as can be shown in figure(3.4)

We dropped the voucher date column since it had more than 50 percent null values as shown in figure (3.5)



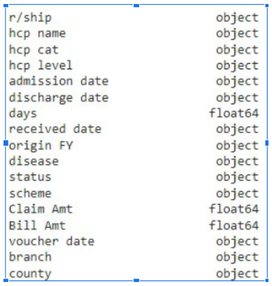
```
[ ]  df = df.drop(columns=['voucher date'])
     df
```

Figure 3.5: Dropping the voucher date column



No. of Claims	100000
Categorical variables	14
Numerical Variables	3
No. of Attributes	17

Table 3.1: Description of claims data

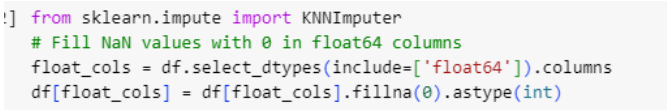


```
r/ship                         object
hcp name                       object
hcp cat                        object
hcp level                      object
admission date                 object
discharge date                 object
days                          float64
received date                  object
origin FY                      object
disease                        object
status                         object
scheme                         object
Claim Amt                     float64
Bill Amt                      float64
voucher date                   object
branch                         object
county                         object
```

Figure 3.6: Variables of the claims data

The original dataset consisted of 904,884 rows and 17 columns.However, due to limited computing power,we used a sample that consisted of 100,000 rows and 17 columns.The variables are as shown in table (3.1) and figure (3.6)

Some of the variables were later used to generate the features that were used to train the models used in this study.

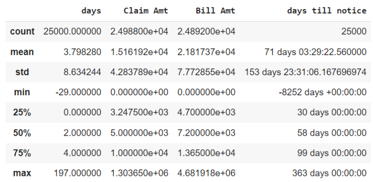


```
]  from sklearn.impute import KNNImputer
   # Fill NaN values with 0 in float64 columns
   float_cols = df.select_dtypes(include=['float64']).columns
   df[float_cols] = df[float_cols].fillna(0).astype(int)
```

Figure 3.7: Imputation of null values per feature.

The missing values were then imputed using the simple imputer and KNN imputer in scikit learn as shown in figure (3.7). We then proceeded to check the object and numerical features of our data which were 14 columns and 3 columns respectively.

Checking summary statistics for the numerical data:



	days	Claim Amt	Bill Amt	days till notice
count	25000.000000	2.498800e+04	2.489200e+04	25000
mean	3.798280	1.516192e+04	2.181737e+04	71 days 03:29:22.560000
std	8.634244	4.283789e+04	7.772855e+04	153 days 23:31:06.167696974
min	-29.000000	0.000000e+00	0.000000e+00	-8252 days +00:00:00
25%	0.000000	3.247500e+03	4.700000e+03	30 days 00:00:00
50%	2.000000	5.000000e+03	7.200000e+03	58 days 00:00:00
75%	4.000000	1.000000e+04	1.365000e+04	99 days 00:00:00
max	197.000000	1.303650e+06	4.681918e+06	363 days 00:00:00

Figure 3.8: Checking summary statistics

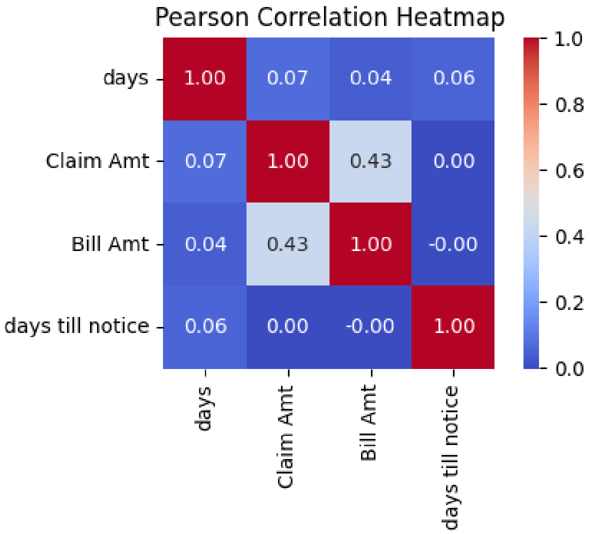


Figure 3.9: Pearson correlation heatmap

The heatmap analysis among the numerical values showed a low correlation and we thus retained all the numerical variables as shown in figure (3.9)

For the models to use the data with converted categorical values(Verma et al. (2017)), defines categorical data encoding as the process of turning categorical data into integer format. He continued by defining categorical data as information that has been obtained and is organised into groups and has a limited number of possible values.After encoding the categorical variables by both one-hot and label encoding,the dimension of our dataset increased to 100,000 rows × 2875 columns.

3.2.4.2 Feature Selection

The following features were maintained for machine learning model classification: days(difference between discharge date and admission date), claim amount(Claim Amt),bill amount(Bill Amt),difference between claim amount and bill amount(amount difference),difference between received date and admission date(days till notice), scheme encoded, disease encoded, status encoded, relationship(r/ship), hospital category(hcp cat),hospital level (hcp level), admission month, discharge month, received month, admission day, discharge day and received day.

3.2.5 Modelling

For this study, three models -Isolation Forest,Local Outlier Factor and One Class SVM were trained and tested to identify the algorithm that performed the best with the unsupervised dataset. The three models were selected based on prior research that produced promising results when the models' performance were evaluated. The claims dataset was split into two: 30 percent for testing the classifier's prediction and 70 percent for training the classifiers. To ascertain which classifier performed the best, the performance metrics for each classifier employed was obtained.After using the unsupervised dataset to train and test the classifiers, the dataset was then classified into normal and anomalous class samples.

3.2.5.1 Hyper Parameter tuning

To improve model performance for optimal results, hyperparameter tuning is essential. For the Isolation forest model, we tweaked the contamination to 0.01,0.05,0.075 and 0.1. Contamination = 0.1 produced the best result for this model.

For the Local outlier Factor, we also tweaked the contamination and n-neighbours hyperparameters.We started with a contamination 0f 0.05 and 10 n-neighbours, then a contamination of 0.15 an 100 n-neighbours, we then settled on a contamination of 0.1 and 1000 n-neighbours because the model trained faster and had better performance

For the OCSVM model we tweaked the nu and gamma hyperparameters.We started with nu=0.01 and gamma=0.01, but this did not yield good model performance.We then changed the hyperparameters to nu=0.5 and gamma=0.5.This yielded the best model performance.

Chapter 4

Data Analysis and Results

4.1 Introduction

The key goal of this chapter is to present the research findings and to investigate how machine learning algorithms can leverage features extracted from NHIF claims datasets to aid in the identification of fraudulent insurance claims. The best fraudulent detection results were determined using comparative analysis of three classification models, namely Isolation Forest, Local Outlier Factor(LOF) and One-Class SVM

4.2 Data Exploratory Analysis

4.2.1 Schemes

From the output of the Isolation forest, we were able to analyse the anomalies and observations were made as shown in figure (4.1).The Kangata care scheme had the highest anomaly count with Linda Mama being the lowest.

4.2.2 Hospital Category

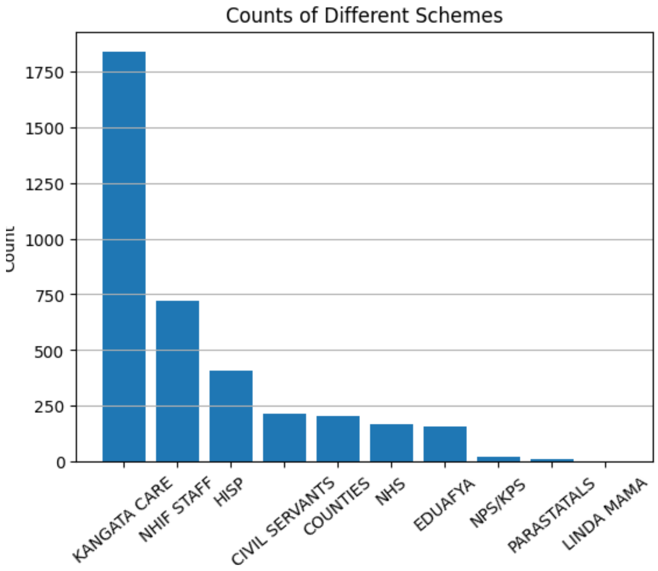


Figure 4.1: Counts of the Schemes

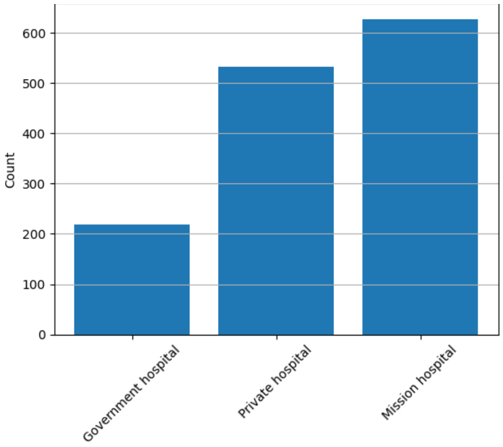


Figure 4.2: Counts of the Hospital Category

The above plot shows most anomalies were in mission hospitals and Government hospitals had the least count of anomalies.

4.2.3 Status

In the figure(4.3),AD(Accounts Deposited) had highest anomalies compared to others while RH(Return to Hospital)has the least anomalies.

4.2.4 Admission Month

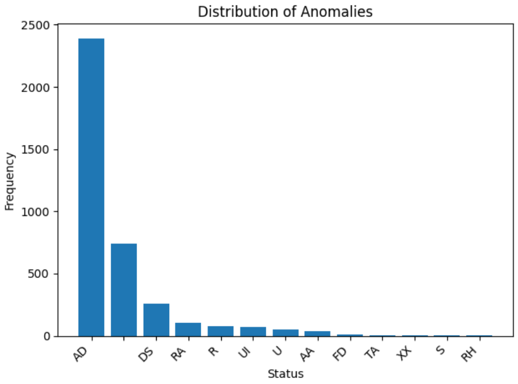


Figure 4.3: Counts of the Status Anomalies

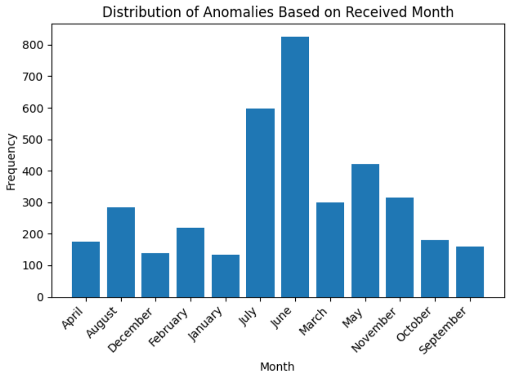


Figure 4.4: Counts of the Status

the figure above shows that most anomalous claims were of the month of June.This might be because of the budget updates that happen around this time and it might pose as an opportunity to upbill/upcode/ claims to get more money.

4.2.5 Days Admitted

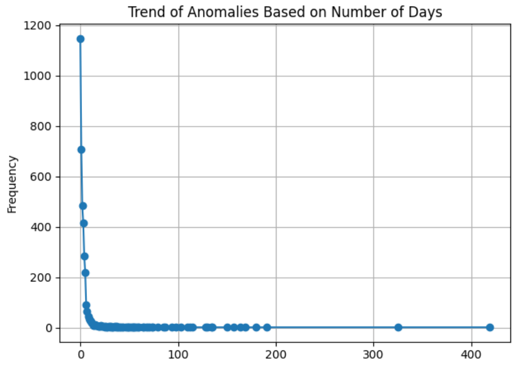


Figure 4.5: Counts of the days anomalies

Most of the anomalies in the days a patient is admitted in an hospital lies around the zero which means that a patient is admitted in the Hospital as an inpatient but end up leaving the same day.

4.3 Evaluation of Machine Learning Algorithms

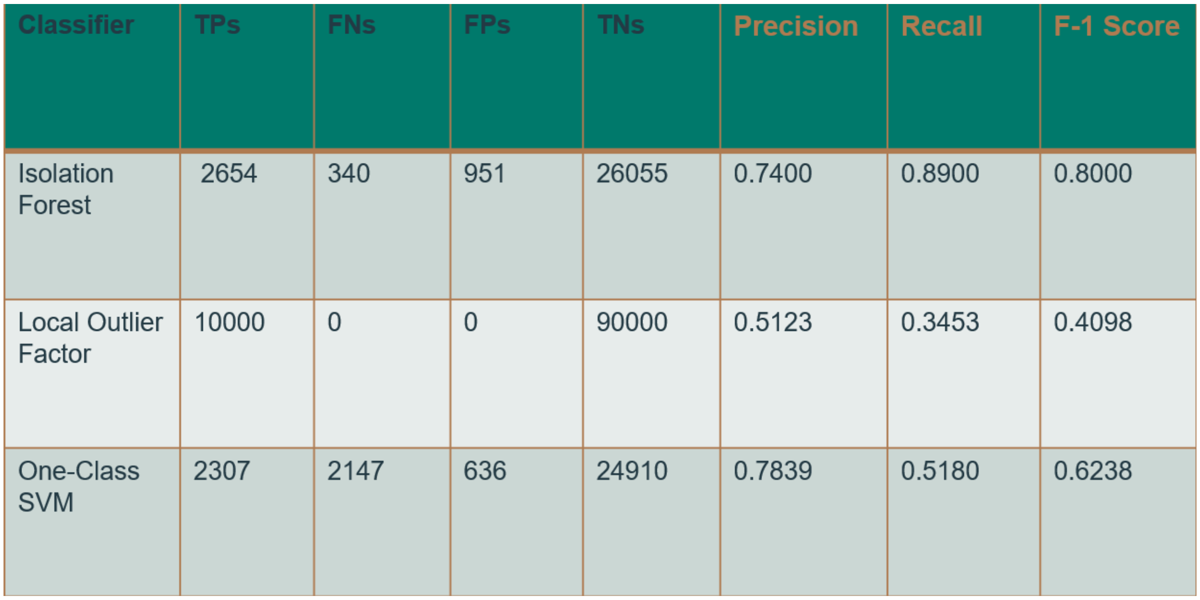
An analysis of the following machine learning classifiers: Isolation Forest, Local Outlier Factor(LOF) and One-Class SVM was performed to identify anomalous claims. Using unlabelled dataset, the classifiers were trained and evaluated to determine which model performed the best. Isolation Forest was the fastest to execute compared to LOF and one class SVM.LOF was seen to execute considerably slow during training compared to the other models which executed quickly.

4.4 Performance Evaluation and Results

In our study, three classification models were trained using the selected features and a dataset divided into 70 percent for training and 30 percent for testing the classifiers. Following the training of all classifiers, the models' performance were evaluated and the classification report of each classifier was calculated based on recall, precision and F-1 score to determine which classifier performed the best.

A confusion matrix is a classification performance indicator that is used to assess the effectiveness of a machine learning algorithm based on target classes(Heydarian et al. (2022)). It is formed by generating TP, which are correctly classified positive claims, TN, which are correctly classified as negative claims, FP, which are classified as negative claims but are positive claims, and FN, which are classified as negative claims but are positive claims.

We also investigated other variables including as precision, recall, F1 score, and accuracy to acquire a deeper understanding of the effectiveness of our models. Precision is the percentage of class members among all those who were expected to be class members who were accurately identified.

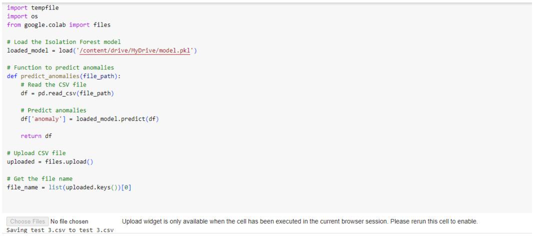


Classifier	TPs	FNs	FPs	TNs	Precision	Recall	F-1 Score
Isolation Forest	2654	340	951	26055	0.7400	0.8900	0.8000
Local Outlier Factor	10000	0	0	90000	0.5123	0.3453	0.4098
One-Class SVM	2307	2147	636	24910	0.7839	0.5180	0.6238

Table 4.1: Evaluation report

Recall is the percentage of all members of a class who were correctly predicted to be a member of that class. F1 score is the harmonic mean of the recall and precision. If precision and recall are both high, the F1 score will be high(Chicco and Jurman (2020)).The evaluation report showed that Isolation Forest performed the best compared to the other two models based on recall,precision and F-1 Score making it superior to other models.

4.5 Overview of the Web-based Tool



```
import tempfile
import os
from google.colab import files

# Load the Isolation Forest model
loaded_model = load('/content/drive/MyDrive/model.pkl')

# Function to predict anomalies
def predict_anomalies(file_path):
    # Read the CSV file
    df = pd.read_csv(file_path)

    # Predict anomalies
    df['anomaly'] = loaded_model.predict(df)

    return df

# Upload CSV file
uploaded = files.upload()

# Get the file name
file_name = list(uploaded.keys())[0]
```

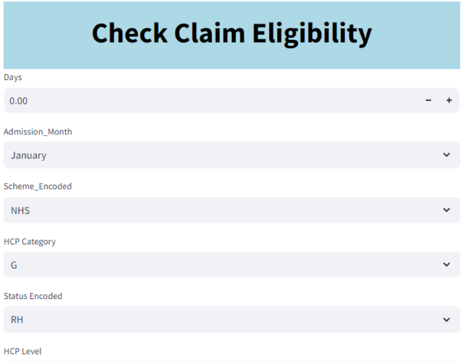
Upload widget is only available when the cell has been executed in the current browser session. Please rerun this cell to enable.

```
Saving test 3.csv to test 3.csv
```

Figure 4.6: The tool deployed in Colab

We deployed two tools that can be used for anomaly detection using streamlit.

This tool takes a standardised dataset and then proceeds to predict each input using the Isolation Forest model.It can be improved by adding a pipeline that can take raw data and automatically standardise it and encode the features that need to be encoded.



Check Claim Eligibility

Days: 0.00

Admission_Month: January

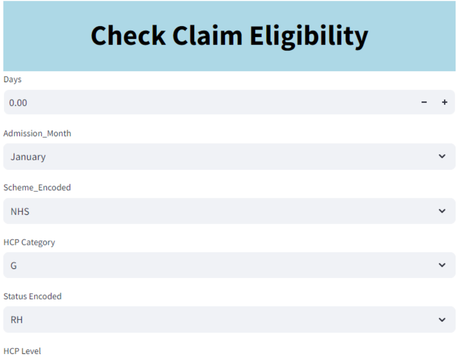
Scheme_Encoded: NHS

HCP Category: G

Status Encoded: RH

HCP Level

Figure 4.7: Tool deployed as a web-based tool



Check Claim Eligibility

Days: 0.00

Admission_Month: January

Scheme_Encoded: NHS

HCP Category: G

Status Encoded: RH

HCP Level

Figure 4.8: web-based Tool

The next model is a web based tool that takes inputs of a single claim and the gives the output, that is, is the claim accepted or should it be investigated as shown in figures (4.8) and (4.9).