Data Glacier Internship Final Project Report

Group: Single Member Group

Name: Hassan Faheem

Batch: LISUM06

Email: Hassan_hsn9@hotmail.com

Country: UAE

University: Heriot-Watt University

Specialization: Data Science

Topic: Healthcare: Persistency of a Drug

Table of Contents

Problem Description

The problem given here is that the Pharmaceutical Company, ABC is in need to understand the persistency of drug as per the physician prescription. The company ABC has thus approached a company that specializes in Analytics, to get this process of identification to be automated. The company has assigned the case to the relevant member to figure out the solution for the automation of persistency of drug for the company ABC.

Business Understanding

The objective of Pharmaceutical Company, ABC is to understand the persistency of a drug for patients. The data obtained shows a large amount of NTM or Non-Tuberculous Mycobacterial infection. The Company hence wants to verify the persistency of the drug that is being prescribed and so the Company would in turn manufacture more those drugs in demand for a more successful business.

Project Lifecycle

Project Name	Healthcare - Persistency of a drug
Start Date	16th March 2022 (Week 7)
Final Submission Date	20th April 2022
Project Duration	5 weeks
Deliverables Submission Dates	1. March 16th 2. March 23rd 3. March 30th 4. April 6th 5. April 13th 6. April 20th

Data Understanding

The Healthcare Dataset provided has 69 columns and 3424 number of observations. The target variable is Persistency_Flag. This variable is of Boolean data type with values that are either True or False. After understanding and analyzing the data, it’s been found that there are few columns that are of numerical data type. Most of the columns are of either Boolean data type or String data type. The column of “Ptid” which refers to Patient ID has no value in terms of model training and thus will be removed the dataset.

Data Intake Report

Name: Healthcare – Persistency of a Drug
Report date: 18th March 2022
Internship Batch: LISUM06
Version:1.0
Data intake by: Hassan Faheem
Data intake reviewer:
Data storage location: https://github.com/hf904/Data-Glacier-Internship/tree/main/Week%207

Tabular data details:

Total number of observations	3424
Total number of files	1
Total number of features	26
Base format of the file	.xlsx
Size of the data	899 KB

Exploratory Data Analysis (EDA)

After performing Exploratory Data analysis on the dataset, the results show that most of the columns are of the Boolean data type and have the values of "Y" and "N". These values will contribute to the model training in their current type and hence were mapped to the values of 1 and 0. Further analysis shows that no Null values were found in the dataset and so did not require any sort of data handling. The analysis show that a certain feature has some outliers and needed to be handled. To fix this, log transformation was performed on this feature to handle the outliers.

Data Types

After analyzing the data, it can be seen that the data has a dimension of (3424, 69). Most of the features here are of type "Object" and very few are of type "int64". The Object type means that the data is of categorical in nature.

Problems in the Dataset

After analysis, it was found that the dataset has no null values. The function "isnull" was used along with sum function to check and verify the null values in the dataset. There were some outliers present in the numerical columns of the dataset. The **Figure 1** below shows the outliers present in Count_Of_Risks. The **Figure 2** shows outliers in Dexa_Freq_During_Rx.

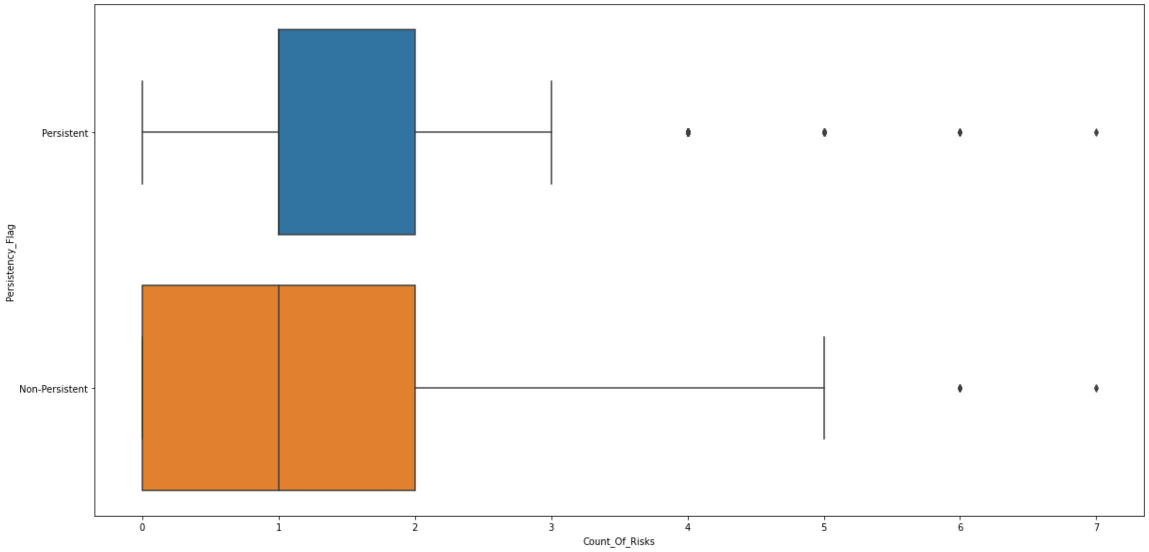


Figure 1: Outliers in Count_Of_Risks

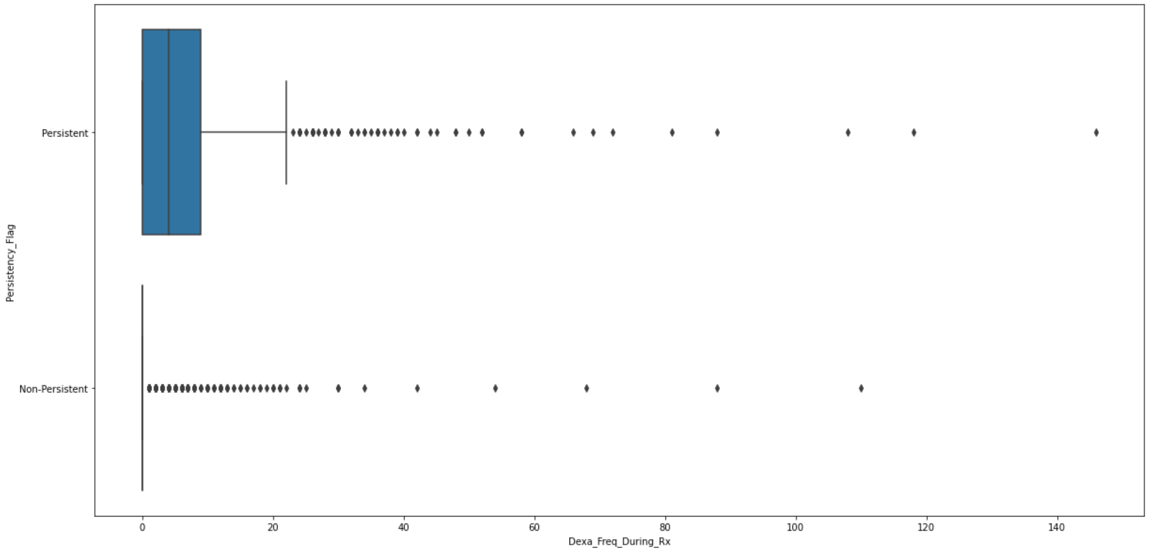
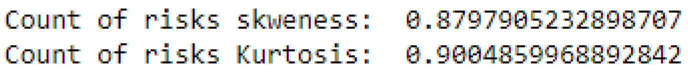


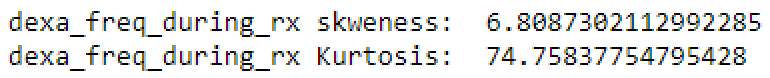
Figure 2: Outliers in Dexa_Freq_During_Rx

Figure 3 and **Figure 4** shows the difference in outliers through skewness and kurtosis in fields of Count_Of_Risks and Dexa_Freq_During_Rx. It can be seen that Dexa_Freq_During_Rx have more skewness and kurtosis which shows that it has more outliers.



```
Count of risks skweness:  0.8797905232898707
Count of risks Kurtosis:  0.9004859968892842
```

Figure 3



```
dexa_freq_during_rx skweness:  6.8087302112992285
dexa_freq_during_rx Kurtosis:  74.75837754795428
```

Figure 4

Data Transformation

- **Null Values**: The dataset did not have any Null values present after the analysis and thus no step was taken in this transformation step.
- **Outliers**: In the numerical features of the dataset, there were outliers present which were shown by the skewness and kurtosis. The function RobustScaler was used to scale the values and the next step is to remove the outliers present and this is done by calculation the inter-quartile range and removing the values with lie outside the whiskers. This step changes and decreases the shape of the data from (3424, 69) to (2964, 69).
- **Changing data type**: The dataset had a lot of columns with the Boolean values of “Y” and “N”. For the purpose of model training, all the values of “Y”, “N” and of the target feature “Persistent”, “Non-Persistent” were changed to [1,0].
- **Unbalanced Dataset**: Unbalanced dataset was the next issue faced and this unbalanced dataset will in turn affect the prediction results. Hence to counter this unbalancing issue, Up Sampling method was used. This method will bump up the records of the class with minority and thus will make all the records equal in count. **Figure 5** below shows the before and after the use of Up Sampling method.
- **One-Hot Encoding**: The final step in the data transformation was the implementation of the function get_dummies which was used for the purpose of One-Hot Encoding. The numerical values are needed for the classifiers to work on and so by this method, the values are transformed into numerical values which can be used by the classifiers.

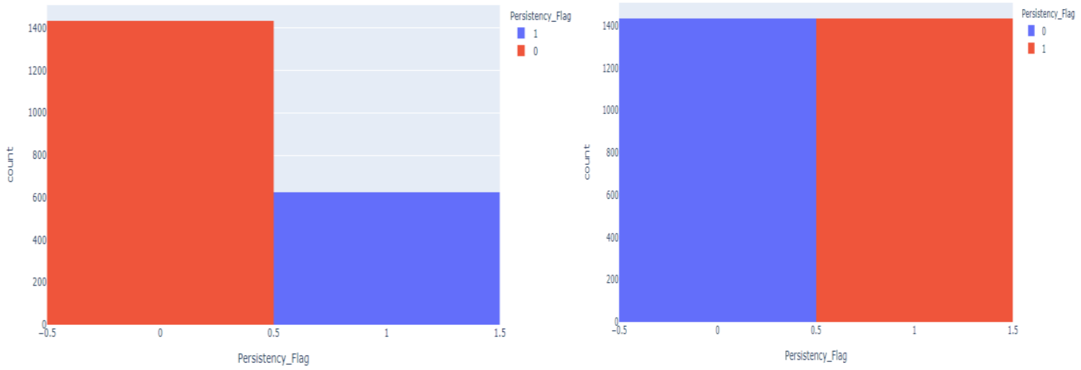


Figure 5: Before & after Up Sampling

Dependency of Data Features

The **Figure 6** below shows the correlation between all the features. It can be seen from the figure that the features that are less correlated are in darker color while the features that are highly correlated are in lighter color.

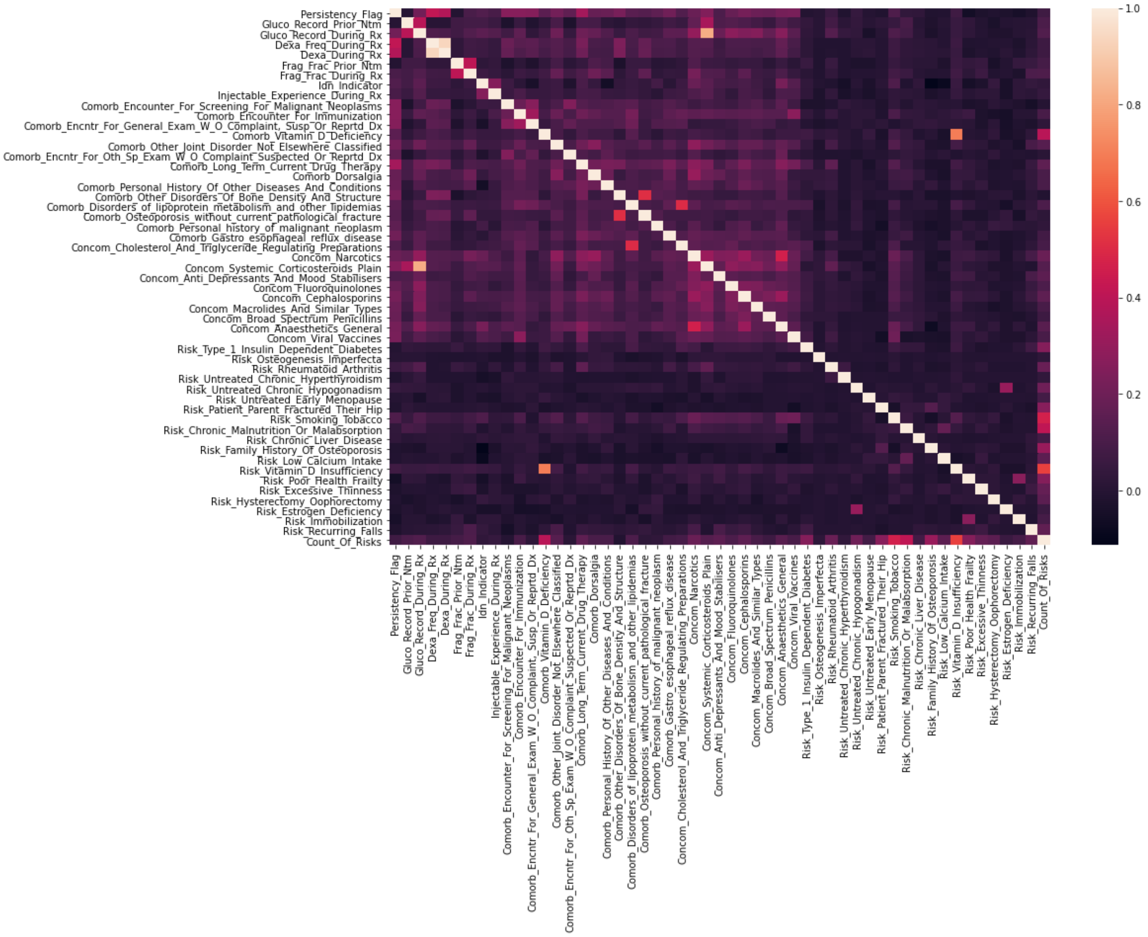
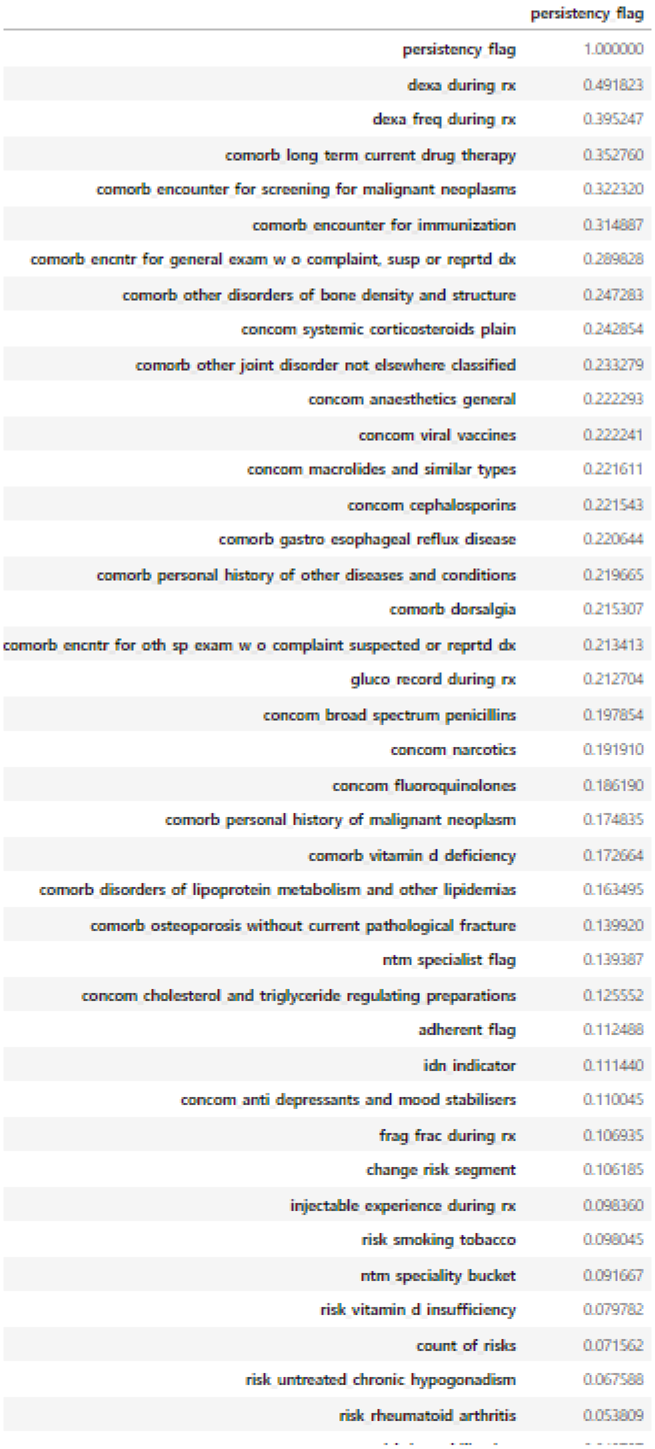


Figure 6: Correlation between Features

Figure 7 shows correlation between the target value and the features. It can be seen that there are not many features that are highly correlated with the target value.



	persistency flag
persistency flag	1.000000
dexa during rx	0.491823
dexa freq during rx	0.395247
comorb long term current drug therapy	0.352760
comorb encounter for screening for malignant neoplasms	0.322320
comorb encounter for immunization	0.314887
comorb encntr for general exam w o complaint, susp or reprtd dx	0.289828
comorb other disorders of bone density and structure	0.247283
concom systemic corticosteroids plain	0.242854
comorb other joint disorder not elsewhere classified	0.233279
concom anaesthetics general	0.222293
concom viral vaccines	0.222241
concom macrolides and similar types	0.221611
concom cephalosporins	0.221543
comorb gastro esophageal reflux disease	0.220644
comorb personal history of other diseases and conditions	0.219665
comorb dorsalgia	0.215307
comorb encntr for oth sp exam w o complaint suspected or reprtd dx	0.213413
gluco record during rx	0.212704
concom broad spectrum penicillins	0.197854
concom narcotics	0.191910
concom fluoroquinolones	0.186190
comorb personal history of malignant neoplasm	0.174835
comorb vitamin d deficiency	0.172664
comorb disorders of lipoprotein metabolism and other lipidemias	0.163495
comorb osteoporosis without current pathological fracture	0.139920
ntm specialist flag	0.139387
concom cholesterol and triglyceride regulating preparations	0.125552
adherent flag	0.112488
idn indicator	0.111440
concom anti depressants and mood stabilisers	0.110045
frag frac during rx	0.106935
change risk segment	0.106185
injectable experience during rx	0.098360
risk smoking tobacco	0.098045
ntm speciality bucket	0.091667
risk vitamin d insufficiency	0.079782
count of risks	0.071562
risk untreated chronic hypogonadism	0.067588
risk rheumatoid arthritis	0.053809

Figure 7: Correlation Between Target & Features

Next Step: Final Recommendation

As seen from the figures of the previous section, its clear that not many features are highly correlated with the target value. Therefore, to avoid any overfitting, it would be in the best interest to ignore the less correlated features during the model training section of the project which comes after this. In the model training section, the dataset in divided into two section with 70% data for training the model and 30% data is given for testing the model.

Model Training & Testing

Classifiers Used

There are Classifiers used from each of the family of Models which include Linear Models, Ensemble & Boosting Models and Neural Network. The following are the classifiers that were trained and tested on:

1. Ensemble & Boosting Models
    - 1.1. Bagging Classifier
    - 1.2. Gradient Boosting Classifier
    - 1.3. Random Forest Classifier
    - 1.4. ExtraTrees Classifier
    - 1.5. AdaBoost Classifier
    - 1.6. XGBoost Classifier
    - 1.7. Stacking Classifier

2. Linear Models
    - 2.1. Ridge Classifier
    - 2.2. SGD Classifier
    - 2.3. Logistic Regression Classifier

3. Neural Network
    - 3.1. Multi-Layer Neural Network
    - 3.2. Multi-Layer Perceptron

Classifiers Train & Test Results

The following table shows the accuracy results of each of the classifiers used for training & testing.

Ensemble & Boosting Models	
Classifier	**Accuracy**
Bagging Classifier	0.80
Gradient Boosting Classifier	0.73
Random Forest Classifier	0.79
ExtraTrees Classifier	0.79
AdaBoost Classifier	0.78
XGBoost Classifier	0.77
Stacking Classifier	0.80
Linear Models	
Classifier	**Accuracy**
Ridge Classifier	0.79
SGD Classifier	0.78
Logistic Regression Classifier	0.78
Neural Network	
Classifier	**Accuracy**
Multi-Layer Neural Network	0.79
Multi-Layer Perceptron	0.76

Table 1: Classifiers Accuracy Comparison

From the **Table 1**, it can be seen that, in the Ensemble & Boosting models' category, Bagging Classifier and Stacking Classifier gave out the best accuracy followed by Random Forest Classifier and ExtraTrees Classifer. In Linear Models category, Ridge Classifier gave out the best accuracy. Finally, in the category of Neural Network, Multi-Layer Neural Network had the better accuracy compared to Multi-Layer Perceptron.

Final Recommendation & Conclusion

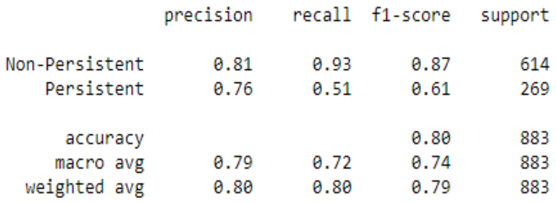
From the table 1 in the above section it is clear that the following classifiers gave out the best results:

- Bagging Classifier
- Stacking Classifier
- Ridge Classifier

To evaluate the classifier models and select the best classifier, there are various methods that can be used. For this project, the following matrices are used to evaluate the classifiers:

- Accuracy
- Recall
- Precision
- F1-Score
- Confusion Matrix

By taking into consideration all the above mentioned metrices, it can be seen that Stacking Classifier gives slightly better result than Ridge Classifier and Bagging Classifier. **Figure 8** shows the evaluation metrices of the Classifier and **Figure 9** shows the Confusion Matrix. Therefore, it can be concluded that Stacking Classifier gives out the best result and hence will be the primary chosen Classifier. The secondary chosen Classifier will be Ridge Classifier. **Figure 10** & **Figure 11** show the evaluation metrices and Confusion Matrix respectively.



```
               precision    recall  f1-score   support

Non-Persistent       0.81      0.93      0.87       614
    Persistent       0.76      0.51      0.61       269

      accuracy                           0.80       883
     macro avg       0.79      0.72      0.74       883
  weighted avg       0.80      0.80      0.79       883
```

Figure 8: Classifier Evaluation

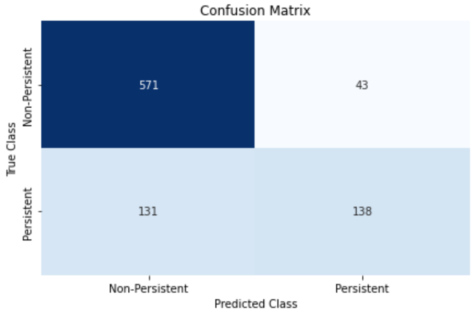
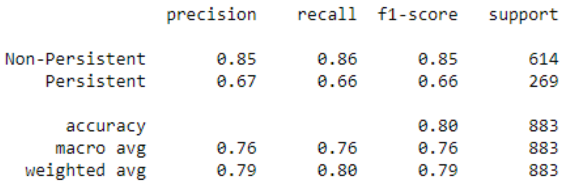


Figure 9: Confusion Matrix



```
               precision    recall  f1-score   support

Non-Persistent       0.85      0.86      0.85       614
    Persistent       0.67      0.66      0.66       269

      accuracy                           0.80       883
     macro avg       0.76      0.76      0.76       883
  weighted avg       0.79      0.80      0.79       883
```

Figure 10: Classifier Evaluation

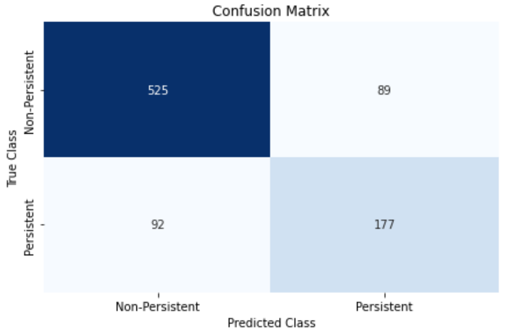


Figure 11: Confusion Matrix