Dustin Snow
7/16/2023
Predictive Analytics
DSC630-T301

Detecting Fraud in Healthcare

Dustin Snow

Milestone 2 – Project Proposal

Data Selection

Healthcare fraud is one of the current leading issues in the United States. Every year, healthcare fraud costs the government and taxpayers billions of dollars. The Centers for Medicare and Medicaid Services estimated that Medicare fraud cost more than $40 billion dollars in 2020 alone (Ruge, 2021). There are many different methods of falling into the trap of healthcare fraud. Some common examples are double billing, upcoding, forgery, and identity theft. The perpetrators of these frauds range from healthcare providers to hospitals to the patients themselves (FBI, 2016). This project will focus on the healthcare providers. The reason being, not only does fraud committed by healthcare providers contribute to the billions of dollars in costs to taxpayers, but it can also put patients in significantly harmful scenarios. A healthcare provider can either suggest unnecessary treatments or fail to prescribe the necessary ones. This can result in significant health risks for the patient, not to mention that unnecessary treatments can force patients to be dropped from their insurance, further increasing health risks for the patient (Wise, 2022).

The data for this project was procured from Kaggle.com. The focus of the dataset is on inpatient data, outpatient data, and beneficiary details. For reference, the inpatient data consists of patients admitted to a hospital, the outpatient data consists of patients seeking care but not admitted to a hospital, and the beneficiary details are extra information gathered about the patients (Sharma, 2022). The goal of this project is to use this data to develop a model that can find and predict healthcare providers that commit

healthcare fraud. A secondary goal of this project is to determine the features that most accurately are predictors of healthcare fraud risk.

Model Selection

Due to the focus variable of this project being fraud designated as “yes” or “no”, the models considered will be classification models. According to Destin Gong, classification models are used to put an object into a category (Gong, 2022). The best model that meets the primary goal of this project will need to be determined. Therefore, multiple models will need to be tested and assessed.

Models to be tested include:

1. Logistic Regression
2. Decision Tree Classifier
3. Random Forest Classifier
4. K-Nearest Neighbor
5. Naïve Bayes

Model Evaluation

Model evaluation will be conducted using a variety of statistical methods. The first will be accuracy. While accuracy is not always the best determinate for a model, exceedingly low accuracy can safely be used to consider a model to be a poor one. F1 scores will also need to be assessed for each model. After some preliminary observations of the dataset, it has been determined that the data is exceedingly imbalanced in the category of “yes fraud” versus “no fraud”. SMOTE will be used to oversample the under-represented class as a way to counteract the imbalance. F1 score will be a better determination of model performance than accuracy because of the imbalance (Zach, 2022). AUC (Area Under the Curve) and ROC (Receiver Operating Characteristics) will also be used to assess model performance. The AUC-ROC curve is a very simple and intuitive method of assessing model performance that plots the True Positive Rate against the False Positive Rate for the model. The higher the AUC, the

better the model is at predicting each class as its true class (Narkhede, 2021). More statistical values will be assessed for relevance at the time of testing.

Learning Objectives

I look forward to working with this dataset. One goal will be to learn how to combine multiple datasets efficiently and effectively into fewer, more cohesive datasets that would be easier to work with. Another goal will be to develop a method to handle severely imbalanced data. I have found that one method to handle this is to use SMOTE. SMOTE stands for Synthetic Minority Oversampling Technique. This method creates new (synthetic) examples from the existing minority classification (Brownlee, 2021). SMOTE is the best option for correcting imbalance because the technique avoids the overfitting that simple up-sampling would introduce by creating the synthetic points close to existing minority points. I plan to determine if there are other methods with the same or better efficacy to handle imbalanced classes. Finally, I hope to become more efficient at assessing multiple models and determining the best fitting model for the goal of the project. For example, accuracy and F1 score may not determine the best model. A confusion matrix comparison between models could determine the best model. Recall, Precision, Type I and Type II error are other statistical model performance metrics to consider when assessing the model for this project. Deciding the best model based on a comparison of multiple metrics will be a requirement of this project.

Ethical Considerations

First and foremost, the goal of this project and the data are very sensitive subjects. Anytime patient data or healthcare data is handled, the utmost care with the data and methods must be taken. Any results or conclusions taken from the project must be rigorously tested because of the risks and wide-ranging consequences should they be wrong. Furthermore, this dataset is already imbalanced. Any model developed from this dataset must take that into account if considering deployment. Developing a model that accuses them of fraud will always have ethical concerns because false accusations of this nature

could ruin careers. Not only would this be bad for the individual providers, but it could also lead to poor outcomes for the United States as a whole. The US already has a significant healthcare provider shortage. Any meddling in that area could prove disastrous. Additionally, because of the nature of the information, bias will have to be checked when developing conclusions by reporting exactly what the results are, good or bad.

Contingency Plan

If there are issues along the path of developing this project, then it will need to be determined if more data is necessary or if the focus of the project as a whole needs to be changed. If more data is necessary, it should be apparent during the exploratory data analysis step of the process. Should this occur, there are more datasets on sites such as Kaggle that offer additional information under this subject. If it is determined that the focus of the project needs to be changed completely, I will go through the same process as I have with the original project and dataset.

Project Milestone 3 – Preliminary Analysis

Data Versus Expectations

Upon preliminary analysis, the data I found has multiple null or missing values. There are many ways to handle missing values. Deleting rows or using mean/median to fill in missing values are some of the ways to handle this issue. Many of the variables are categorical and will need to be converted to a binomial value in order to facilitate model construction. Feature creation can also be implemented for this dataset that may yield more accurate features. For example, age of patient is a feature I intend to create as age could be a significant factor in taking advantage of patients (fraud). Another problem with the complete data available is that the data is already split into a train/test set, and, unfortunately, the test set does not come with the target variable. The train set has enough data to train and test a model with. I will

add an additional step of creating a prediction list for the target variable for the test set after the best model has been selected.

Visualizations

Because of the many categorical variables in the data, countplots will be a good starting point for visualizing the data. I want to understand the count of each variable with respect to the possible fraud variable. This is where the need for fixing the class imbalance in the target variable can be seen. Confusion matrices will be visualizations to assess model performance. In addition to the Confusion matrices, I will be able to create AUC-ROC curve chart that will make it easier to visually compare the models.

Questions versus Modeling

Because of one of my learning objectives, training and assessing multiple models will be the correct path. I do not know which one will be the best fit for the data, but that will be determined at the end of the project using the visualizations and performance metrics listed above. Finding the best model will also complete one of the other goals (developing a model to predict possible healthcare fraud). Extra steps will need to be taken to determine the features that most accurately predict fraud. Some possible methods for this problem are forward feature elimination, backward feature elimination, or recursive feature elimination. Not only will this process help answer what features are most important, but it may also make the models more accurate and generalizable because it reduces the excess from the data.

Project Milestone 4

Import Data and Preparation

- Data Import and Fixing
  - Imported the required data
  - Data was split into 4 different CSV files
  - Needed to develop a process to combine all the files into one file for modeling
  - First, I standardized the column names
  - Then made the outpatient data frame by merging via common variables, checked for duplicates, and created a new variable designating the data as outpatient
  - Next made the inpatient data frame using the same method
  - Merged the inpatient and outpatient data frames to make the final data frame
- Variable Cleaning
  - Convert Categorical variables to binary
  - Convert date and time columns to datetime
  - Feature creation based on the datetime variables
- Looked for Correlations with the Target Variable
  - No features have significant correlation with the target variable
  - The variable with the highest correlation will unfortunately have to be dropped because there are too many missing values with no means of filling them in

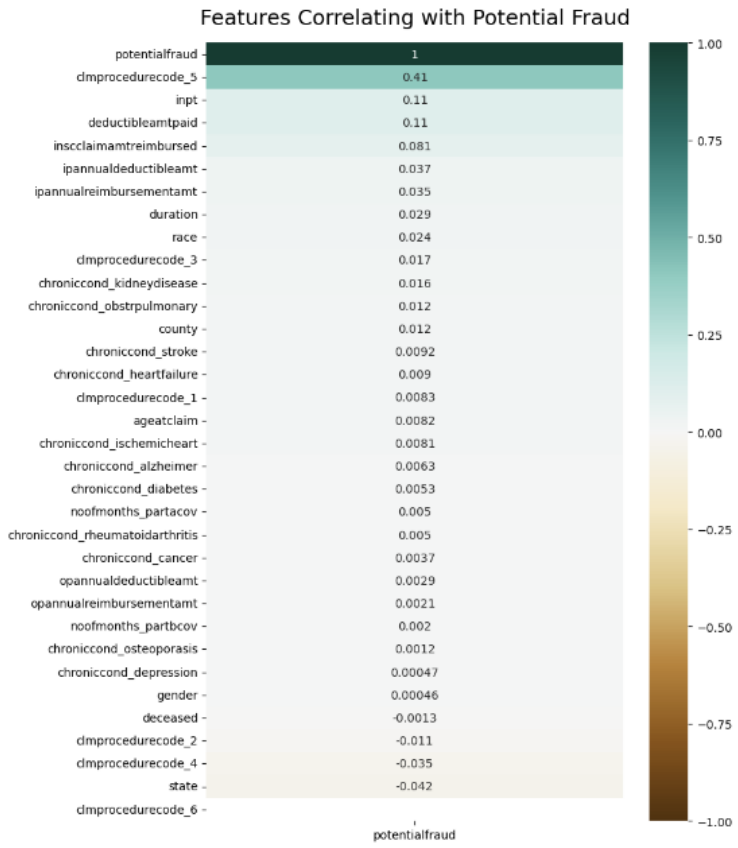
```
#Change 2 to 0 in chronic disease categories & gender
binary=['chroniccond_alzheimer','chroniccond_heartfailure','chroniccond_kidneydisease',
        'chroniccond_cancer','chroniccond_obstrpulmonary','chroniccond_depression',
        'chroniccond_diabetes','chroniccond_ischemicheart','chroniccond_osteoporasis',
        'chroniccond_rheumatoidarthritis','chroniccond_stroke','gender']
for i in binary:
  fraud[i].replace((2),0,inplace=True)
```

```
#encode character variables to 0 or 1
fraud['potentialfraud'].replace(['Yes','No'],[1,0],inplace=True)
fraud['potentialfraud']=fraud['potentialfraud'].astype('int64')

fraud['renaldiseaseindicator'].replace(('Y'),1,inplace=True)
```

```
#convert date and time cols to dt
dates=['claimstartdt','claimenddt','dob','dod','admissiondt','dischargedt']

for i in dates:
  fraud[i]=pd.to_datetime(fraud[i], format='%Y-%m-%d')
```

```
#Calculate age on the first day of claim
fraud['ageatclaim']=np.floor(((fraud['claimstartdt'] - fraud['dob']).dt.days)/365.25)

#Calculate duration of claim
fraud['duration'] = ((fraud['claimenddt'] - fraud['claimstartdt']).dt.days)+1
```

```
#fix the deceased column
fraud.loc[fraud.dod.isnull(),'deceased']=0
fraud.loc[fraud.dod.notnull(), 'deceased']=1
```



- Missing Values
  - Determined how many missing values there were for each variable
  - Dropped the variables with mostly missing data
  - Dropped the variables that would not be important for modeling, such as the ID columns
  - Filled in the ‘deductibleamtpaid’ variable with the mean value for the column

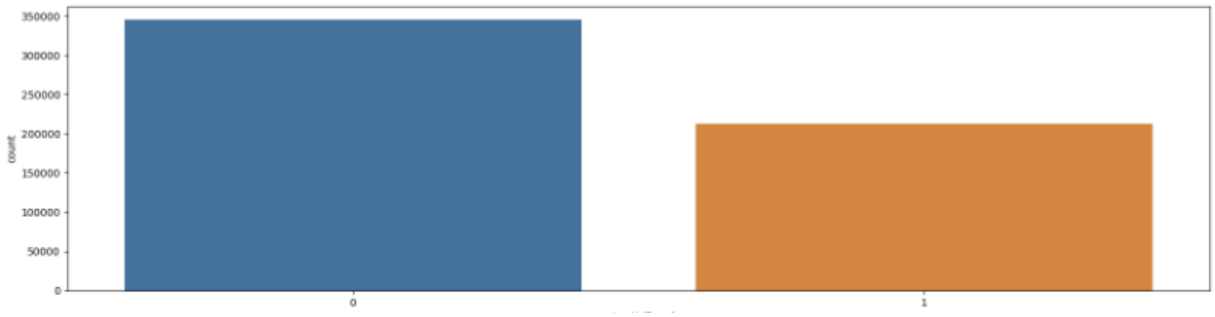
```
#drop columns that are either unnecessary for modeling or have excess NaN values
fraud=fraud.drop(['beneid','claimid','dod','noofmonths_partacov','noofmonths_partbcov',
                  'operatingphysician','otherphysician','attendingphysician',
                  'admissiondt','dischargedt','claimstartdt','claimenddt',
                  'clmadmitdiagnosiscode','diagnosisgroupcode','clmdiagnosiscode_1',
                  'clmdiagnosiscode_2', 'clmdiagnosiscode_3','clmdiagnosiscode_4',
                  'clmdiagnosiscode_5', 'clmdiagnosiscode_6','clmdiagnosiscode_7',
                  'clmdiagnosiscode_8', 'clmdiagnosiscode_9','clmdiagnosiscode_10',
                  'clmprocedurecode_1', 'clmprocedurecode_2','clmprocedurecode_3',
                  'clmprocedurecode_4', 'clmprocedurecode_5','clmprocedurecode_6','provider','los'],axis=1)
```

```
mean_value=fraud['deductibleamtpaid'].mean()
```

```
fraud['deductibleamtpaid'].fillna(value=mean_value, inplace=True)
```

- Target variable inspection, Train, Test split, and using SMOTE to correct underrepresentation in the target variable, Implemented a standard scaler to increase fit for the models and reduce bias.

```
#get a count of the potetial fraud
plt.figure(figsize = (20,5))
sns.countplot(x = 'potentialfraud', data = fraud)
plt.show()
#non fraud cases outweigh fraud cases
```




```
#use SMOTE to correct for underrepresentation of target variable
sm = SMOTE(random_state=42)
X_train_SMOTE, y_train_SMOTE = sm.fit_resample(X_train, y_train)

print("Shape before SMOTE: ", X_train.shape, y_train.shape, "\n")
print("Shape after SMOTE: ", X_train_SMOTE.shape, y_train_SMOTE.shape, "\n")
```

```
Shape before SMOTE:  (418658, 24) (418658,)

Shape after SMOTE:  (518122, 24) (518122,)
```

```
stdsc = StandardScaler()
X_train_SMOTE_std = stdsc.fit_transform(X_train_SMOTE)
X_val_std = stdsc.transform(X_val)
```

- Used Recursive Feature Elimination to determine the 10 best features to use for modeling to increase accuracy and lower bloat

```
#use RFE to find the 10 best features for modeling
from sklearn.ensemble import RandomForestClassifier
rfe = RFE(estimator=RandomForestClassifier(), n_features_to_select=10)
_ = rfe.fit(X_train_SMOTE_std,y_train_SMOTE)
print('Important Features\n',X.columns[rfe.support_])
```

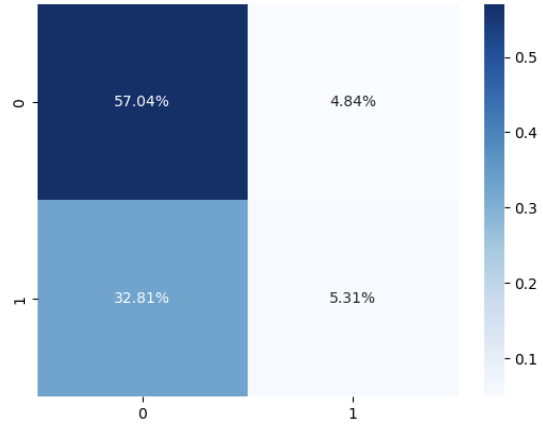
```
Important Features
 Index(['inscclaimamtreimbursed', 'deductibleamtpaid', 'race',
       'renaldiseaseindicator', 'ipannualreimbursementamt',
       'ipannualdeductibleamt', 'opannualreimbursementamt',
       'opannualdeductibleamt', 'ageatclaim', 'duration'],
      dtype='object')
```

```
X = X[['inscclaimamtreimbursed', 'deductibleamtpaid', 'race',
       'renaldiseaseindicator', 'ipannualreimbursementamt',
       'ipannualdeductibleamt', 'opannualreimbursementamt',
       'opannualdeductibleamt', 'ageatclaim', 'duration']]
```

```
X_train, X_val, y_train, y_val = train_test_split(X, y, test_size=0.25, random_state=54321, stratify=y)
X_train_SMOTE, y_train_SMOTE = sm.fit_resample(X_train, y_train)
X_train_SMOTE_std = pd.DataFrame(stdsc.fit_transform(X_train_SMOTE), columns=X_train_SMOTE.columns)
X_val_std = pd.DataFrame(stdsc.fit_transform(X_val), columns=X_val.columns)
```

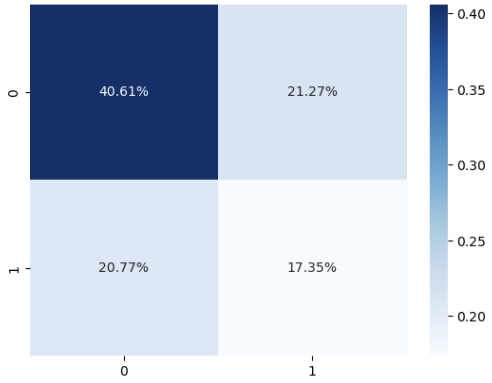
Logistic Regression Model

```
Classification Report for Logistic Regression Model
              precision    recall  f1-score   support

           0       0.63      0.92      0.75     86354
           1       0.52      0.14      0.22     53199

    accuracy                           0.62    139553
   macro avg       0.58      0.53      0.49    139553
weighted avg       0.59      0.62      0.55    139553
```



Decision Tree Classifier

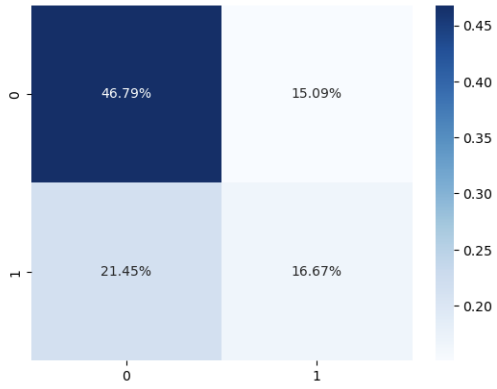
```
Classification Report for Decision Tree Classifier Model
              precision    recall  f1-score   support

           0       0.66      0.66      0.66     86354
           1       0.45      0.46      0.45     53199

    accuracy                           0.58    139553
   macro avg       0.56      0.56      0.56    139553
weighted avg       0.58      0.58      0.58    139553
```



Random Forest Classifier

Classification Report for the Random Forest Classifier Model

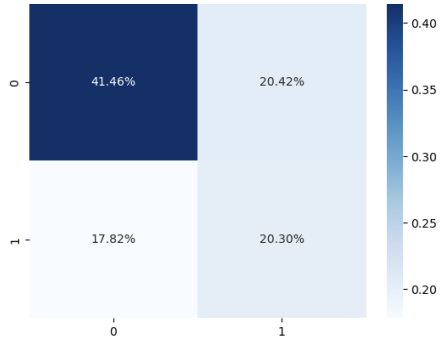
	precision	recall	f1-score	support
0	0.69	0.76	0.72	86354
1	0.52	0.44	0.48	53199
accuracy			0.63	139553
macro avg	0.61	0.60	0.60	139553
weighted avg	0.62	0.63	0.63	139553



K-Nearest Neighbors Classifier

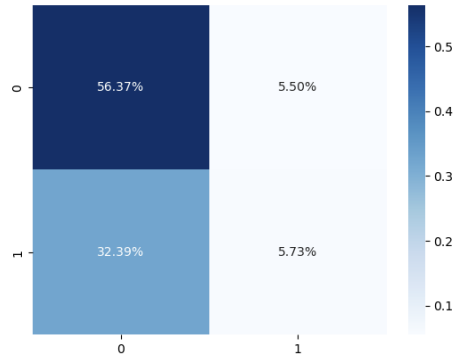
Classification Report for the K-Nearest Neighbors Model

	precision	recall	f1-score	support
0	0.70	0.67	0.68	86354
1	0.50	0.53	0.51	53199
accuracy			0.62	139553
macro avg	0.60	0.60	0.60	139553
weighted avg	0.62	0.62	0.62	139553



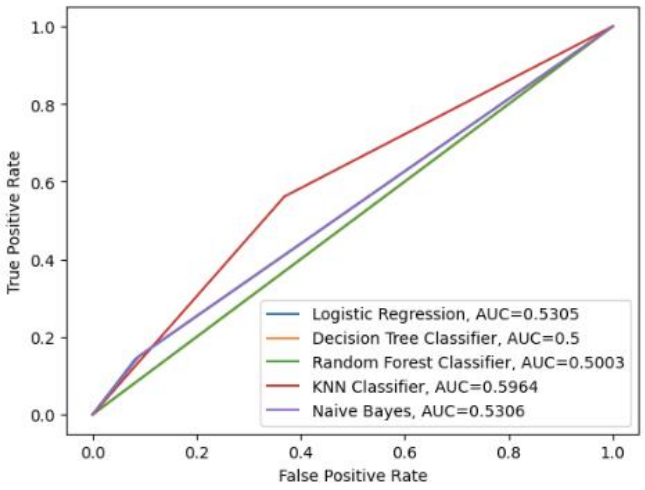
Naïve Bayes Classifier

```
Classification Report for the Naive Bayes Model
              precision    recall  f1-score   support

           0       0.64      0.91      0.75     86354
           1       0.51      0.15      0.23     53199

    accuracy                           0.62    139553
   macro avg       0.57      0.53      0.49    139553
weighted avg       0.59      0.62      0.55    139553
```



Plotting the AUC Curve for each of the Models

```
#Plot AUC for each of the Models
from sklearn import metrics
fpr, tpr, _ = metrics.roc_curve(y_val, y_pred_lr)
auc = round(metrics.roc_auc_score(y_val, y_pred_lr), 4)
plt.plot(fpr,tpr,label="Logistic Regression, AUC="+str(auc))

fpr, tpr, _ = metrics.roc_curve(y_val, y_pred_dt)
auc = round(metrics.roc_auc_score(y_val, y_pred_dt), 4)
plt.plot(fpr,tpr,label="Decision Tree Classifier, AUC="+str(auc))

fpr, tpr, _ = metrics.roc_curve(y_val, y_pred_rf)
auc = round(metrics.roc_auc_score(y_val, y_pred_rf), 4)
plt.plot(fpr,tpr,label="Random Forest Classifier, AUC="+str(auc))

fpr, tpr, _ = metrics.roc_curve(y_val, y_pred_knn)
auc = round(metrics.roc_auc_score(y_val, y_pred_knn), 4)
plt.plot(fpr,tpr,label="KNN Classifier, AUC="+str(auc))

fpr, tpr, _ = metrics.roc_curve(y_val, y_pred_nb)
auc = round(metrics.roc_auc_score(y_val, y_pred_nb), 4)
plt.plot(fpr,tpr,label="Naive Bayes, AUC="+str(auc))

plt.ylabel('True Positive Rate')
plt.xlabel('False Positive Rate')
plt.legend()
```



Results Interpretation

None of the models produced great results. Accuracy, F1 score, and AUC curve were used to evaluate the models. Only the KNN classifier model and the Naïve Bayes classifier model had an accuracy at or above 60%. The F1 score for the KNN classifier was 0.66 for a no fraud classification and 0.52 for a fraud classification. The F1 score for the Naïve Bayes classifier model was 0.75 for a no fraud classification and 0.23 for a fraud classification. This suggests the Naïve Bayes classifier actually has a

hard time accurately predicting true fraud. The AUC Curve plot supports this interpretation. The plot shows the KNN classifier has the highest ability to distinguish between fraud and non-fraud cases.

Initial Conclusions

Feature selection was used as a means to increase the accuracy of the models, as well as to determine possible indicators of future fraud. Because the reasoning behind committing fraud is rooted in money, it is no wonder that most of the important features Recursive Feature Elimination found are compensation related. The race and age of the patient at the time of the claim also make sense. It is not unrealistic to assume the providers that are going to commit fraud will take advantage of those that are easy targets or those that they have bias against. The modeling unfortunately did not produce great results. The KNN classifier was determined to be the best model for this data, however it was only able to accurately classify positive and negative cases just slightly over half the time (only slightly better than guessing). This suggests that it is very difficult to predict fraud from providers, which is the commonly held understanding. Companies need to commit to better data collection for this problem for there to be any possibility of predicting fraud in the future. I had to drop too many variables that may have been important because things such as diagnosis codes were not accurately recorded.

References

Brownlee, J. (2021, March 16). *Smote for imbalanced classification with python*. MachineLearningMastery.com. https://machinelearningmastery.com/smote-oversampling-for-imbalanced-classification/

Fact sheet 2020 estimated improper payment rates for Centers for Medicare & Medicaid Services (CMS) programs. CMS. (n.d.). https://www.cms.gov/newsroom/fact-sheets/2020-estimated-improper-payment-rates-centers-medicare-medicaid-services-cms-programs

FBI. (2016, June 1). *Health Care Fraud*. FBI. https://www.fbi.gov/investigate/white-collar-crime/health-care-fraud

Gong, D. (2022, July 12). *Top 6 machine learning algorithms for classification*. Medium. https://towardsdatascience.com/top-machine-learning-algorithms-for-classification-2197870ff501

Narkhede, S. (2021, June 15). *Understanding AUC - ROC Curve*. Medium. https://towardsdatascience.com/understanding-auc-roc-curve-68b2303cc9c5

Ruge, E. (2021, November 11). *Medicare fraud costs $65 billion a year. you can help.* ClearMatch Medicare. https://clearmatchmedicare.com/blog/medicare/medicare-fraud-costs-65-billion-a-year-you-can-help

Sharma, B. (2022, April 20). *Fraudlent claim in Healthcare*. Kaggle. https://www.kaggle.com/datasets/beenusharma42/fraudlent-claim-in-healthcare

Wise, C. (2022, March 14). *Healthcare fraud and its consequences*. HRG. https://www.hrgpros.com/blog/healthcare-fraud-and-its-consequences#:~:text=Unnecessary%20procedures%20and%20prescriptions%20can,a%20patient's%20medical%20insurance%20coverage.

Zach. (2022, May 19). *What is a "good" accuracy for Machine Learning Models?* Statology. https://www.statology.org/good-accuracy-machine-learning/