Fetal Cardiotocography

OVERVIEW

The objective of this analysis is to indicate the conditions under which fetal health is determined to be suspect or pathological. The data is derived from fetal cardiotocographic monitors which measure the vitals of a fetus in utero. Additional analytical features have been added to the dataset.

An overview of the features is as follows:

- **Date** - Date of examination
- **B, E** - Timestamp features indicating the beginning and end of the reading
- **AC, FM, UC, DL, DS, DP, DR** - Features counting the occurrences of certain indicators (ex. Fetal movements, uterine contractions)
- **LBE, LB, ASTV, mSTV, ALTV, MLTV** - Analytical features indicating baseline values of fetal heart rate, as well as averages and percentages of heart rate variability
- **Width, Min, Max, Nmax, Nzeros, Mode, Mean, Median, Variance, Tendency** - Analytical features indicating the size and shape of the histogram of fetal heart rate. “Nmax” and “Nzeros” are respectively a count of the local maxima and of the zero values.
- **NSP** - Classification feature indicating whether the fetal health is:
    - 1 - Normal
    - 2 - Suspect
    - 3 - Pathological

The analysis will attempt to show which features have the greatest impact on our ability to determine the class in the “NSP” feature. As such, the analysis will prioritize interpretability over predictive power. The error metric to be used for evaluating the models will be “recall” as we want to reduce the occurrence of false negatives. In other words it is better to falsely label a “Normal” pattern as “Suspect” or “Pathological” than to falsely label a “Suspect” or “Pathological” pattern as “Normal.”

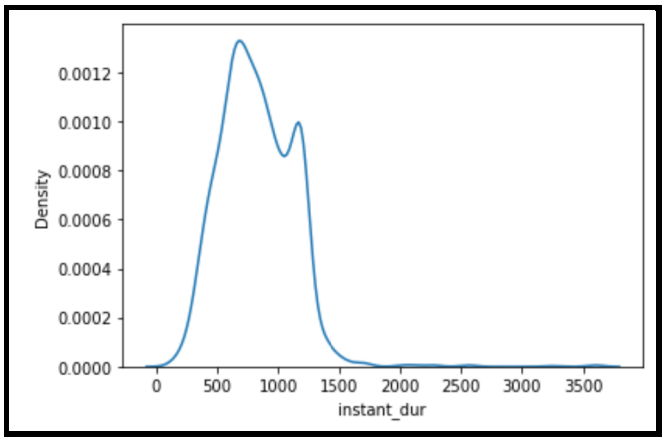
DATA CLEANING AND EXPLORATORY DATA ANALYSIS

The first steps of data cleaning were to remove the “Date” feature and to convert the “B” and “E” features to a single feature indicating the *duration* of the reading (labeled “instant_dur”).

Below are the descriptions of the features. Of particular note are features in which the the values are consistently zero through the 75th quartile, and only have a small number of non-zero values. In addition, it appears the features “LB” and “LBE” have the same data points. This was confirmed upon closer inspection, and therefore the redundant feature “LBE” was removed.

	count	mean	std	min	25%	50%	75%	max
LBE	2126.0	133.303857	9.840844	106.0	126.000000	133.000000	140.000000	160.000000
LB	2126.0	133.303857	9.840844	106.0	126.000000	133.000000	140.000000	160.000000
AC	2126.0	2.722484	3.560850	0.0	0.000000	1.000000	4.000000	26.000000
FM	2126.0	7.241298	37.125309	0.0	0.000000	0.000000	2.000000	564.000000
UC	2126.0	3.659925	2.847094	0.0	1.000000	3.000000	5.000000	23.000000
ASTV	2126.0	46.990122	17.192814	12.0	32.000000	49.000000	61.000000	87.000000
MSTV	2126.0	1.332785	0.883241	0.2	0.700000	1.200000	1.700000	7.000000
ALTV	2126.0	9.846660	18.396880	0.0	0.000000	0.000000	11.000000	91.000000
MLTV	2126.0	8.187629	5.628247	0.0	4.600000	7.400000	10.800000	50.700000
DL	2126.0	1.570085	2.499229	0.0	0.000000	0.000000	3.000000	16.000000
DS	2126.0	0.003293	0.057300	0.0	0.000000	0.000000	0.000000	1.000000
DP	2126.0	0.126058	0.464361	0.0	0.000000	0.000000	0.000000	4.000000
Width	2126.0	70.445908	38.955693	3.0	37.000000	67.500000	100.000000	180.000000
Min	2126.0	93.579492	29.560212	50.0	67.000000	93.000000	120.000000	159.000000
Max	2126.0	164.025400	17.944183	122.0	152.000000	162.000000	174.000000	238.000000
Nmax	2126.0	4.068203	2.949386	0.0	2.000000	3.000000	6.000000	18.000000
Nzeros	2126.0	0.323612	0.706059	0.0	0.000000	0.000000	0.000000	10.000000
Mode	2126.0	137.452023	16.381289	60.0	129.000000	139.000000	148.000000	187.000000
Mean	2126.0	134.610536	15.593596	73.0	125.000000	136.000000	145.000000	182.000000
Median	2126.0	138.090310	14.466589	77.0	129.000000	139.000000	148.000000	186.000000
Variance	2126.0	18.808090	28.977636	0.0	2.000000	7.000000	24.000000	269.000000
Tendency	2126.0	0.320320	0.610829	-1.0	0.000000	0.000000	1.000000	1.000000
NSP	2126.0	1.304327	0.614377	1.0	1.000000	1.000000	1.000000	3.000000
instant_dur	2126.0	824.437441	305.054024	117.0	613.000000	799.000000	1041.000000	3599.000000

As indicated in the overview, there are certain features which are a raw count of an activity. In the table above we notice there is a large difference between the min and max value of the “instant_dur” feature. In other words, some readings were much longer in duration than others. Because of this, the features indicating a count of an activity may be positively correlated with the duration of the reading, and therefore should be scaled to reflect frequency instead of absolute count. Below shows the density plot of the “instant_dur” feature, its kurtosis and skew, and the correlations of the features with the “instant_dur” feature.




```
kurtosis: 8.7317
skew: 1.3543
```

AC	0.330838
FM	0.028391
UC	0.519352
DL	0.229195
DS	0.020675
DP	0.072802
Nmax	0.120304
Nzeros	-0.015493

The density plot shows the “instant_dur” feature has a positive kurtosis and positive skew. With the exception of “Nzeros,” all features are positively correlated with the “instant_dur” feature. The negative correlation associated with the “Nzeros” feature is less of a concern as almost all values of that feature are zero. Therefore, we would expect all of the features indicated as being correlated with “instant_dur” to also have a positive kurtosis and skew. These features will need to be transformed into frequency features by dividing their values by the “instant_dur” values. If this scaling is effective, we would expect to see the skew and kurtosis of the features to decrease as the correlation with the duration of the reading is eliminated. The skew and kurtosis metrics are shown below.

	kurtosis				skew			
	Original	Scaled	Difference	Pct_Diff	Original	Scaled	Difference	Pct_Diff
AC	3.112334	0.779710	-2.332624	-0.749477	1.657659	1.209231	-0.448429	-0.270519
FM	104.385626	64.113096	-40.272530	-0.385805	9.420843	7.806644	-1.614199	-0.171343
UC	1.283403	-0.650755	-1.934158	-1.507055	0.834757	0.156266	-0.678491	-0.812800
DL	3.136630	2.497658	-0.638972	-0.203713	1.817836	1.720355	-0.097480	-0.053624
DS	298.717589	320.327281	21.609692	0.072342	17.341211	17.799132	0.457922	0.026407
DP	19.172450	20.027846	0.855396	0.044616	4.232899	4.275781	0.042882	0.010131
Nmax	0.500205	4.645230	4.145026	8.286660	0.892256	1.739340	0.847085	0.949374
Nzeros	30.290897	16.814344	-13.476553	-0.444904	3.917521	3.431430	-0.486090	-0.124081

Looking at the “Pct_Diff”, we see five of the features (AC, FM, UC, DL, Nzeros) had a large decrease in both kurtosis and skew, two (DS, DP) had a very small increase in both, and “Nmax” had a large increase in both, especially kurtosis. Though not perfect, on the whole I believe the transformation of these features into frequencies makes them more informative for the analysis going forward.

MODEL #1 - LOGISTIC REGRESSION

Because we are placing a premium on models that provide interpretability, I wanted to start with Logistic Regression using L1 regularization which will provide the greatest amount of interpretability. After splitting the data to Test and Training datasets using a stratified shuffle split, we see the classes in the “NSP” feature are imbalanced, with over 77% of the records belonging to NSP = 1. This indicates we will need to use undersampling and/or oversampling to correct for the imbalance.

```
y_train.value_counts(normalize=True)
```

```
1    0.778226
2    0.139113
3    0.082661
Name: NSP, dtype: float64
```

```
y_test.value_counts(normalize=True)
```

```
1    0.778997
2    0.137931
3    0.083072
Name: NSP, dtype: float64
```

The sampling methods to be used are:

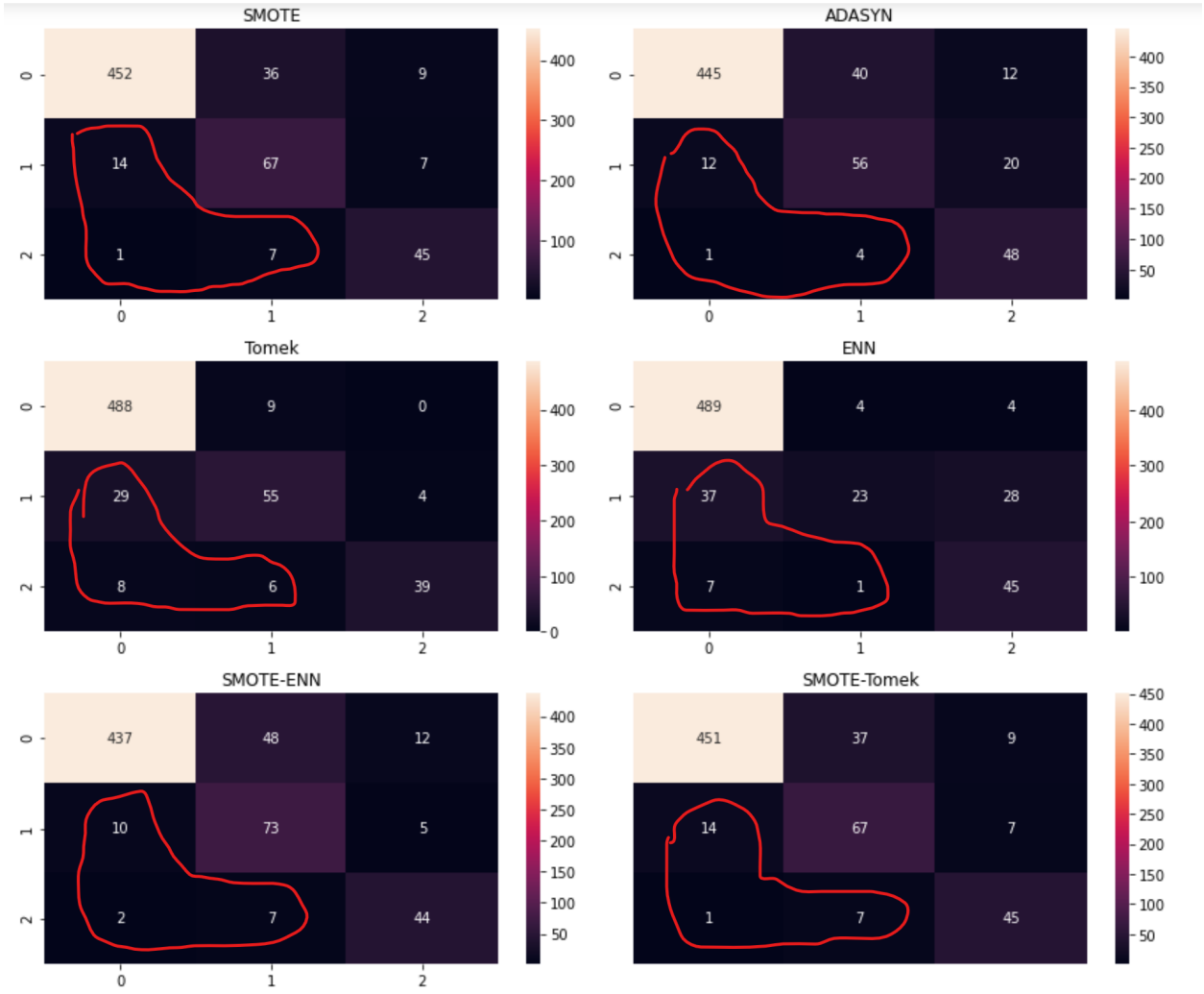
- SMOTE
- ADASYN
- Edited Nearest Neighbors (ENN)
- Tomek Links (Tomek)
- SMOTE + ENN
- SMOTE + Tomek

After fitting the model, we see that the L1 regularization was not able to eliminate many features, and all of the sampling methods use the full 21 coefficients for at least one of the classes. Therefore, none of the sampling methods can be favored over another in terms of simplicity. After printing the error metrics and the confusion matrices, we see the following performance of the different sampling methods.

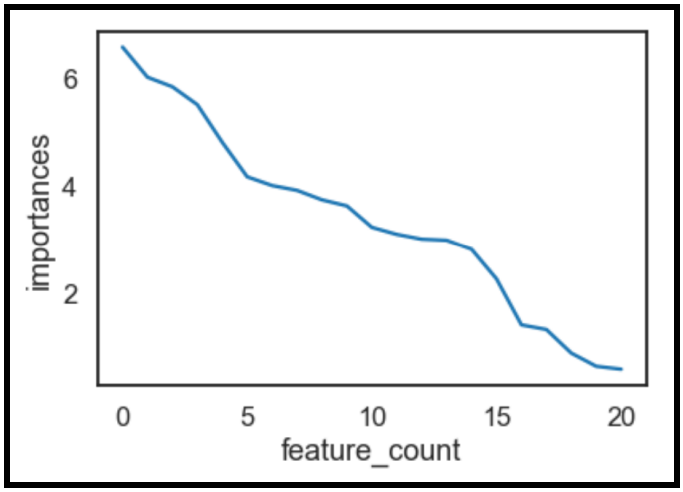
```
('SMOTE', 0) 21 coefficients
('SMOTE', 1) 21 coefficients
('SMOTE', 2) 20 coefficients
('ADASYN', 0) 21 coefficients
('ADASYN', 1) 21 coefficients
('ADASYN', 2) 21 coefficients
('Tomek', 0) 20 coefficients
('Tomek', 1) 21 coefficients
('Tomek', 2) 21 coefficients
('ENN', 0) 21 coefficients
('ENN', 1) 17 coefficients
('ENN', 2) 21 coefficients
('SMOTE-ENN', 0) 21 coefficients
('SMOTE-ENN', 1) 20 coefficients
('SMOTE-ENN', 2) 21 coefficients
('SMOTE-Tomek', 0) 21 coefficients
('SMOTE-Tomek', 1) 21 coefficients
('SMOTE-Tomek', 2) 20 coefficients
```

	accuracy	precision	recall	f1
SMOTE_train	0.872769	0.875972	0.872769	0.873594
SMOTE_test	0.884013	0.771559	0.839959	0.801334
ADASYN_train	0.842775	0.844003	0.842440	0.842650
ADASYN_test	0.860502	0.710539	0.812465	0.749829
Tomek_train	0.907522	0.838079	0.790211	0.811726
Tomek_test	0.912226	0.874072	0.780913	0.821231
ENN_train	0.961921	0.882245	0.831897	0.853625
ENN_test	0.876176	0.780352	0.695062	0.694500
SMOTE-ENN_train	0.904409	0.908020	0.904928	0.905888
SMOTE-ENN_test	0.868339	0.754966	0.846337	0.790582
SMOTE-Tomek_train	0.874640	0.878609	0.874642	0.875553
SMOTE-Tomek_test	0.882445	0.769707	0.839288	0.799832

‘SMOTE,’ ‘ADASYN,’ ‘SMOTE-ENN’ and ‘SMOTE-Tomek’ have recall scores that are close to one another and higher than the rest of the sampling methods. In the confusion matrices below we are looking for the model that has the fewest false negatives for classes 1 and 2 (circled in red). From this perspective ‘ADASYN’ performs best as it has 17 false negatives, while the others have between 19 and 45 false negatives.



In order to rank the feature importances, we find the three coefficients for each of the features (one for each class), then sum the absolute values of the coefficients. At this point we will only be using the ADASYN sampling method for analysis of feature importance. We see from the line graph below there is no “elbow” point where we can easily separate the most important features from the least important features. Rather the importance appears to decrease in a linear fashion.



index	0	1	2	importances
LB	-0.425946	-2.627816	3.531076	6.584838
AC_per	2.412613	-1.588212	-2.023590	6.024415
Width	3.397336	0.397322	-2.051907	5.846565
Median	1.076123	2.252326	-2.184910	5.513358
Mean	-1.523743	1.882615	-1.407443	4.813801
DS_per	0.059617	-4.021459	-0.090879	4.171955
Variance	-2.293145	-0.523247	1.192700	4.009092
Max	-2.245252	0.414360	1.259683	3.919295
Min	2.258244	0.088806	-1.395966	3.743015
DL_per	0.962175	1.218267	-1.449366	3.629808
MSTV	1.091960	-1.987877	-0.150817	3.230654
Mode	0.887287	-0.356619	-1.854671	3.098577
MLTV	0.526683	1.473650	-1.008628	3.008961

The rankings are above, and show the top five features are:

1. LB - Baseline value
2. AC_per - Frequency of accelerations
3. Width - Width of histogram
4. Median - Median of histogram
5. Mean - Mean of histogram

However, “top five” is an arbitrary perspective, and as mentioned above, there are no features we can ignore without potentially hindering model performance. This does create an impediment to model interpretability, as there are 63 coefficients (21 features x 3 classes) to consider at any given time.

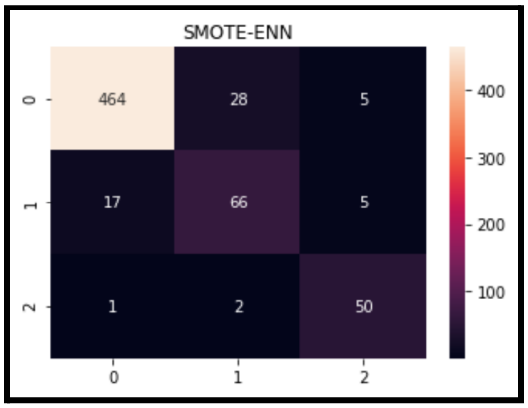
MODEL #2 - DECISION TREE

The second model to be used will be based on the Decision Tree method. If using a single tree, the dimensions of it will indicate the degree to which it would be useful in practice. In addition, we can still determine the relative importance of the features for purposes of interpretability.

As with Logistic Regression, we will be trying out the different oversampling and undersampling methods to determine which provides the best performance. In addition to measuring recall and the absolute number of false negatives for classes 1 and 2, we will also prioritize the models which have the smallest tree dimensions in terms of number of nodes and tree depth.

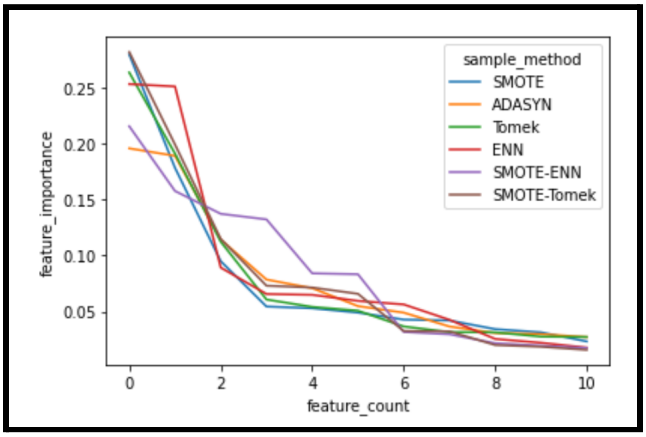
The first tree created placed no limits on depth or number of features to be used. The error metrics show the model is overfitting on the training set as the training recall score is 1 or almost 1 for all sampling methods, which is not matched by the test recall scores. However, the SMOTE-ENN method shows a high recall score on the test set. The confusion matrix for the SMOTE-ENN method shows 20 false negatives for classes 1 and 2. In addition the shape of the tree based on SMOTE-ENN is one of the least complex, second only to the ENN.

	accuracy	precision	recall	f1
SMOTE_train	0.999712	0.999712	0.999712	0.999712
SMOTE_test	0.910658	0.854469	0.830810	0.840491
ADASYN_train	0.999714	0.999712	0.999718	0.999715
ADASYN_test	0.915361	0.858367	0.850871	0.853319
Tomek_train	0.999298	0.999705	0.998084	0.998892
Tomek_test	0.909091	0.844797	0.836990	0.840250
ENN_train	1.000000	1.000000	1.000000	1.000000
ENN_test	0.888715	0.809908	0.782783	0.771800
SMOTE-ENN_train	1.000000	1.000000	1.000000	1.000000
SMOTE-ENN_test	0.909091	0.827830	0.875666	0.850084
SMOTE-Tomek_train	0.999711	0.999711	0.999711	0.999711
SMOTE-Tomek_test	0.907524	0.846115	0.832586	0.838653



	sampler	nodes	depth
0	SMOTE	281	18
1	ADASYN	321	19
2	Tomek	157	14
3	ENN	47	9
4	SMOTE-ENN	151	15
5	SMOTE-Tomek	273	18

In order to potentially reduce the overfitting and reduce the error, we will iterate through different configurations of max depth limits and feature limits. The line graph below indicates the feature importances flatten out at between 2 features and 6 features depending on the sampling technique. We therefore can eliminate many features to simplify our subsequent trees. The node and depth counts show we can also set a max depth = 17.



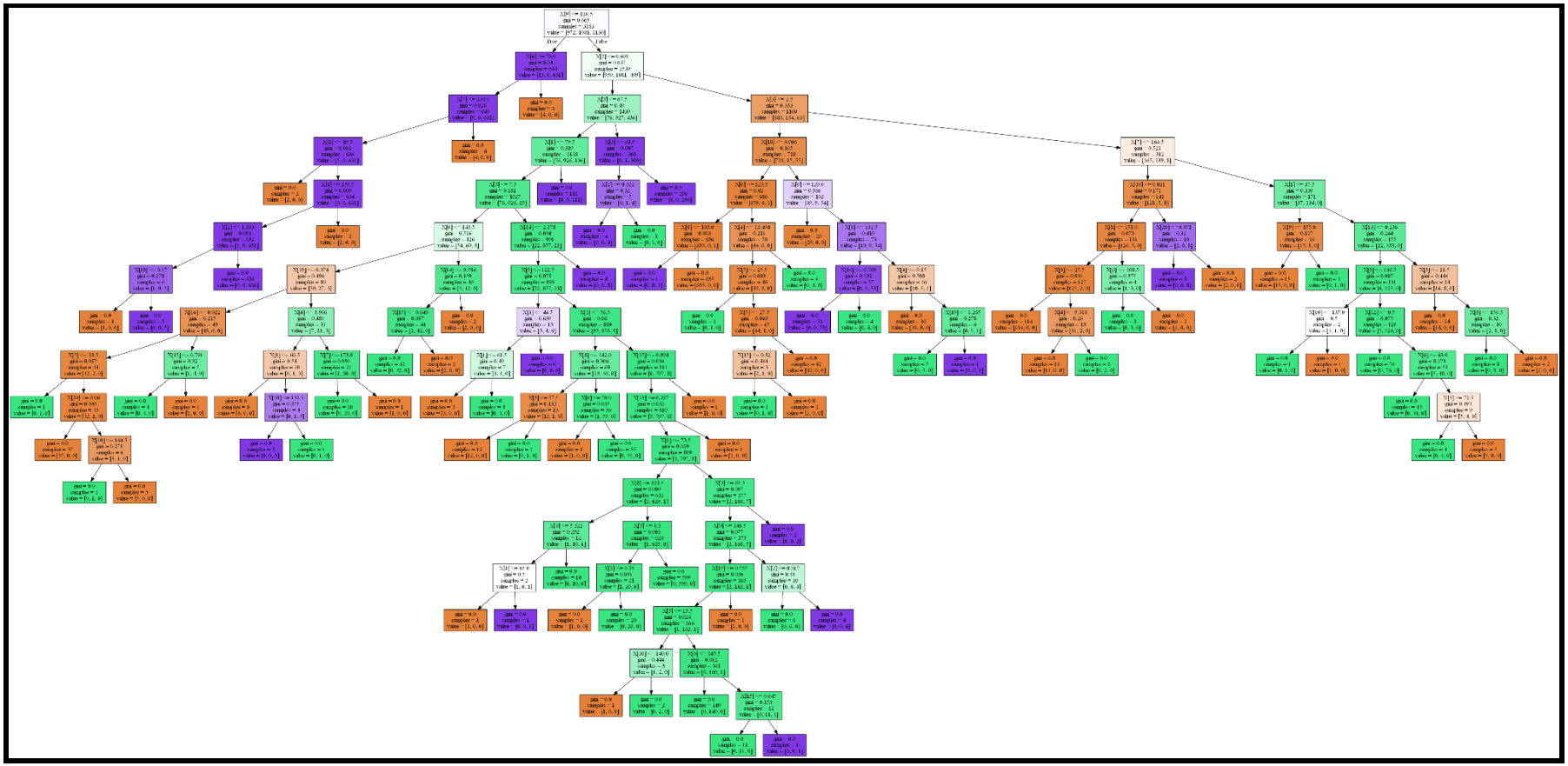
	sampler	nodes	depth
0	SMOTE	285	17
1	ADASYN	315	17
2	Tomek	131	11
3	ENN	51	9
4	SMOTE-ENN	161	17
5	SMOTE-Tomek	257	15

After setting these parameters and creating another tree we see the results below. They indicate that performance decreased and overfitting increased. In addition, the trees increased in the number of nodes and in only three scenarios reduced the depth. Therefore, it appears the initial tree is the best to use both in terms of increasing performance and reducing complexity.

	accuracy	precision	recall	f1
SMOTE_train	0.999712	0.999712	0.999712	0.999712
SMOTE_test	0.898119	0.811935	0.807973	0.808614
ADASYN_train	0.997140	0.997144	0.997158	0.997149
ADASYN_test	0.884013	0.789582	0.790046	0.789547
Tomek_train	0.999298	0.999705	0.998084	0.998892
Tomek_test	0.896552	0.798388	0.798567	0.795829
ENN_train	0.998311	0.999334	0.991914	0.995596
ENN_test	0.877743	0.789253	0.757499	0.752350
SMOTE-ENN_train	1.000000	1.000000	1.000000	1.000000
SMOTE-ENN_test	0.874608	0.749834	0.837205	0.786011
SMOTE-Tomek_train	0.982370	0.982488	0.982364	0.982389
SMOTE-Tomek_test	0.891850	0.807379	0.799018	0.803141

	sampler	nodes	depth
0	SMOTE	423	17
1	ADASYN	453	13
2	Tomek	231	16
3	ENN	95	11
4	SMOTE-ENN	323	15
5	SMOTE-Tomek	365	11

The full decision tree is shown below, along with the feature importances for the top 6 features on the next page:



feature_count	index	feature_importance
0	MSTV	0.262224
1	Mean	0.228821
2	ALTV	0.221880
3	ASTV	0.100305
4	Max	0.042580
5	DP_per	0.033569

MODEL #3 - DECISION TREE WITH GRADIENT BOOSTING

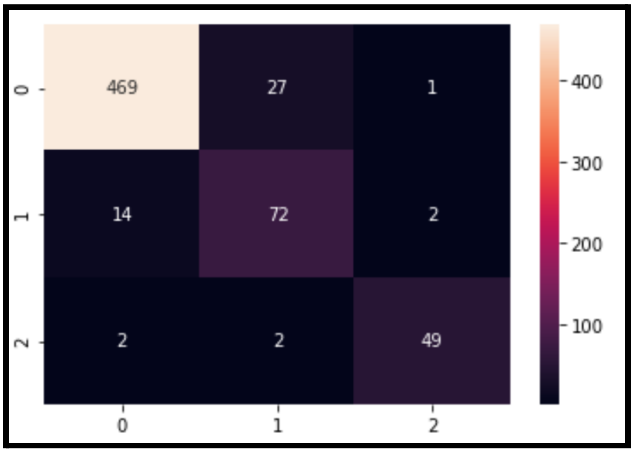
For the third model, we will create a decision tree using gradient boosting. Both should assist in reducing the overfitting seen in the decision trees above while also increasing performance. However, this will come at the cost of interpretability as we will only be able to view feature importance without being able to view an actual decision tree.

Using the SMOTE-ENN sampling method, the boosting algorithm was tuned to find the best performance with the following hyperparameters:

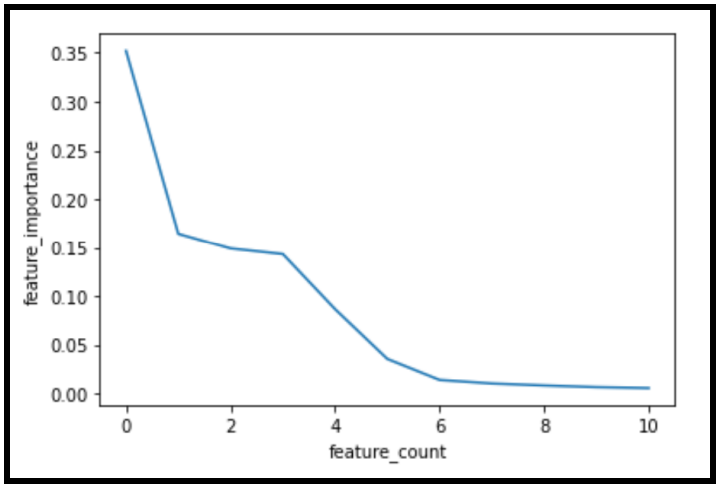
- Number of samples = 300
- Learning Rate = 0.7
- Subsample = 0.6

The error metrics show the performance improved by using gradient boosting compared to creating a single tree without any subsamples.

	SMOTEENN_train	SMOTEENN_test
accuracy	1.0	0.924765
precision	1.0	0.874063
recall	1.0	0.895457
f1	1.0	0.883477



Similar to the standalone trees in the previous model, we see in the line graph below there are six features that have most of the importance in our model.



feature_count	index	feature_importance
0	ALTV	0.351886
1	Mean	0.164357
2	ASTV	0.148982
3	AC_per	0.143202
4	DP_per	0.086649
5	MSTV	0.035643
6	Mode	0.014099

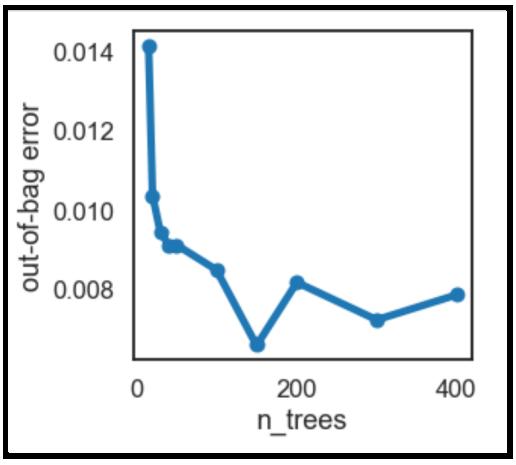
Unlike our logistic regression model, we *can* separate these first six features from the rest without losing much model performance, instead of choosing an arbitrary cutoff point to interpret the top features. These top six features are as follows:

1. ALTV - Percentage of time with abnormal long term variability
2. Mean - Mean of histogram
3. ASTV - Percentage of time with abnormal short term variability
4. AC_per - Frequency of accelerations
5. DP_per - Frequency of prolonged decelerations
6. MSTV - Mean value of short term variability
7. Mode - Mode of histogram

At this point we are not able to determine the impact each feature has on each class, but additional models could assist in this to provide more interpretability to the model.

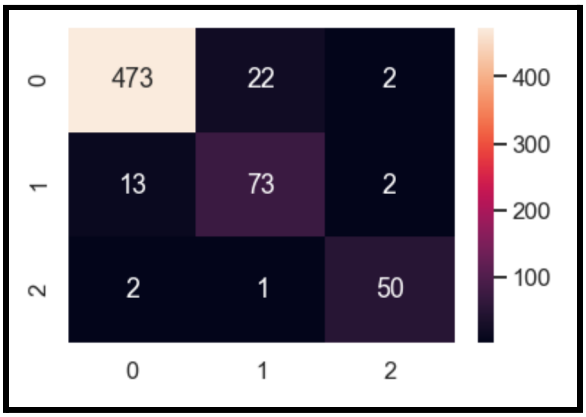
MODEL #4 - RANDOM FOREST

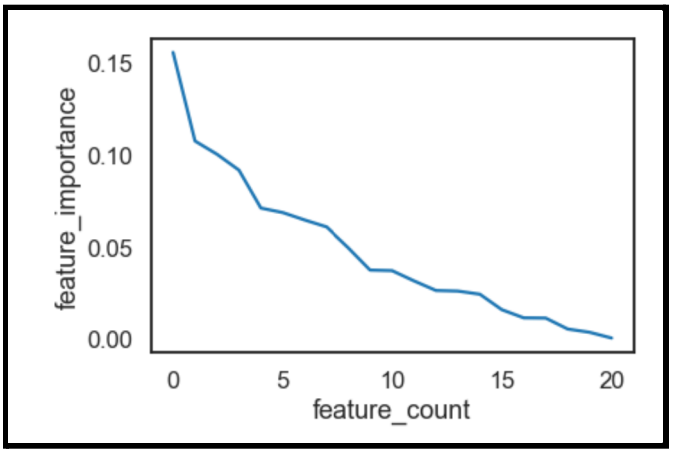
The next classifier I used was a random forest classifier. Using the SMOTE-ENN sampling method, I iterated on the number of trees to use in the model and evaluated the out-of-box error for each. The line graph below shows we saw the least error when using 150 trees.



When evaluating the results (shown below), we see the random forest classifier performed only slightly better than the gradient boosting model. In addition, when looking at the feature importances, there is no clear division point between important and non-important features as there was above. In other words, we lose simplicity and interpretability with the random forest model, and don't have enough of an increase in performance to show for it.

	SMOTE_ENN_train	SMOTE_ENN_test
accuracy	1.0	0.934169
precision	1.0	0.885202
recall	1.0	0.908217
f1	1.0	0.896155





The feature importance for the top six features is shown below. As with the logistic regression model, using a six-feature cut-off is arbitrary since there is no natural cut-off in the feature importances.

	feature_count	index	feature_importance
0	0	ALTV	0.155433
1	1	Mean	0.107380
2	2	ASTV	0.100207
3	3	MSTV	0.091600
4	4	DP_per	0.071027
5	5	AC_per	0.068569

CONCLUSIONS

Upon evaluating the four models above, the logistic regression model is the preferred model as it provides a high level of performance with very few false negatives for class 1 and 2, while maintaining a high level of interpretability due to the fact we have coefficient values for each feature and each class. If the goal of these models would be for real-time use in a labor and delivery room by a medical professional, there is a drawback to having 63 important coefficients to evaluate at once.

It is possible that with further modeling, the decision tree model with gradient boosting could provide a more ideal balance of predictive power and simplicity. If we could determine the impact of the six important features indicated above on the predicting class while maintaining the high level of predictive power then this model would be the preferred model. It could even be simple enough for use in a labor and delivery room to evaluate fetal health in real-time.

When evaluating individual features, we see that in most of our models, the two features that are among the most important are:

- Mean - Mean of histogram
    - Positive correlation with class 2, negative with classes 1 and 3
- AC_per - Frequency of accelerations
    - Positive correlation with class 1, negative with classes 2 and 3

Four additional features appear among the most important features in all the tree-based models. These are:

- ALTV - Percentage of time with abnormal long term variability
    - Positive correlation with class 1, negative with classes 2 and 3
- ASTV - Percentage of time with abnormal short term variability
    - Positive correlation with class 2, negative with classes 1 and 2
- MSTV - Mean value of long term variability
    - Positive correlation with class 1, negative with classes 2 and 3
- DP_per - Frequency of prolonged decelerations
    - Positive correlation with class 2 and 3, negative with class 1

POTENTIAL NEXT STEPS

As mentioned above, additional modelling can be done to increase the interpretability of the tree-based models, so we can evaluate whether or not they surpass the logistic regression model both in terms of predictive power and interpretability.

Another potential area of further study is to introduce polynomial features of degree two. This will certainly increase computing time and may decrease performance due to overfitting. However, it could also yield more powerful features that could allow for high predictive power with few features.