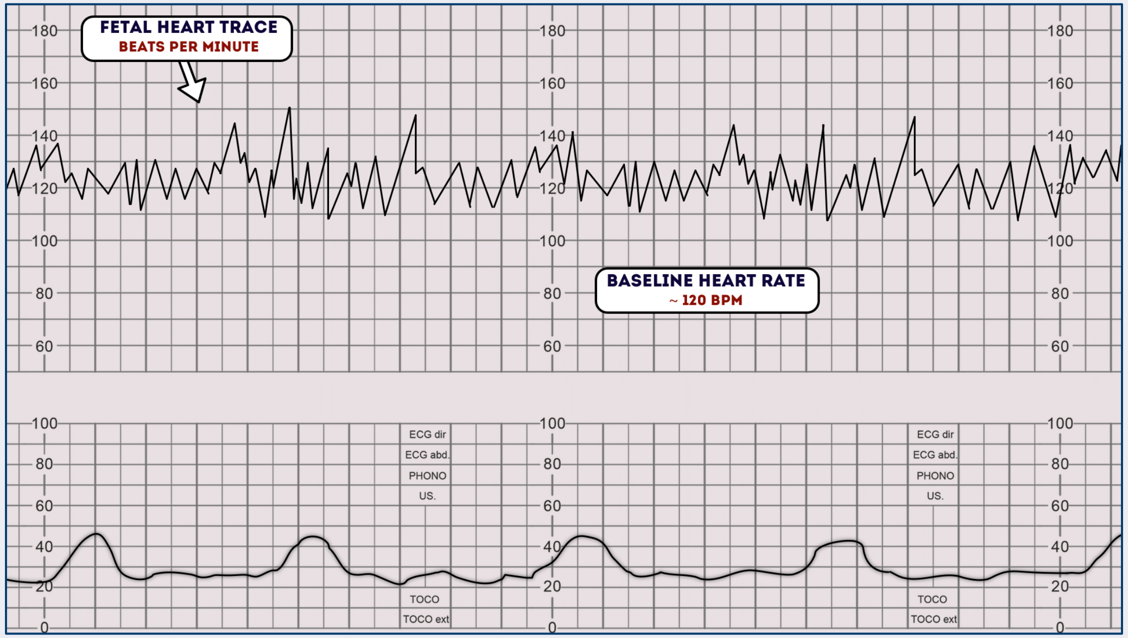
Fetal Cardiotocograph (CTG) Classification

Kate Stadelman

Cardiotocographs (CTGs)



How to Read a CTG

- Monitor fetal heart rate & uterine contractions
- Cost-effective way to help **reduce fetal mortality**

CTG Pros & Cons

- Detect fetal compromise
- Allow mothers to seek early treatment

- Require specific training for interpretation
- Same test interpreted differently based on reviewer

science.engineering.analytics.insights DATA

Research Question

How do Random Forest, Logistic Regression, and K-Nearest Neighbors compare in predicting a normal vs. abnormal CTG given test metrics?

CTG Dataset

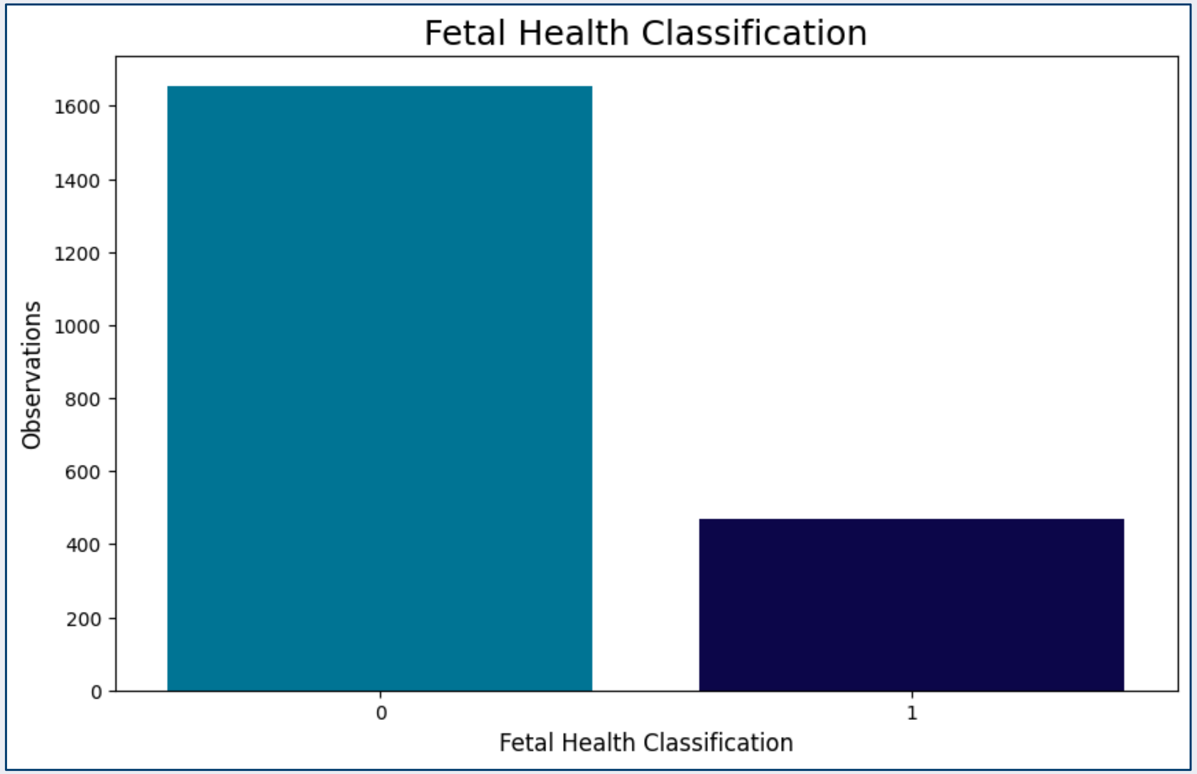
2,126 CTG Records ❖ 21 Numeric Input Features

Numeric Input Feature Categories
Uterine Contractions
Fetal Movements
Fetal Heart Rate (FHR) Baseline
FHR Accelerations & Decelerations
FHR Short- & Long-Term Variability
FHR Histogram Descriptive Statistics

Output Class Labels
Normal (Class 0)
Abnormal (Class 1)

CTG reading classification by three expert obstetricians

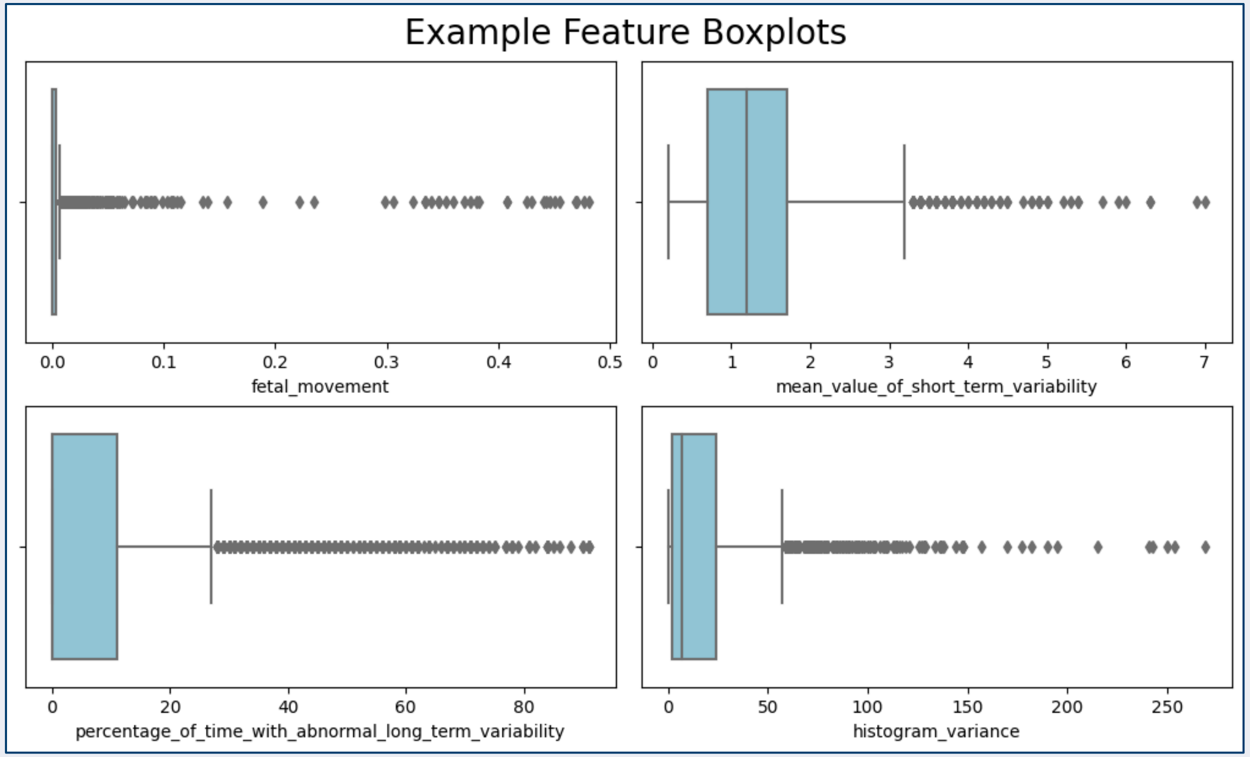
Exploratory Data Analysis



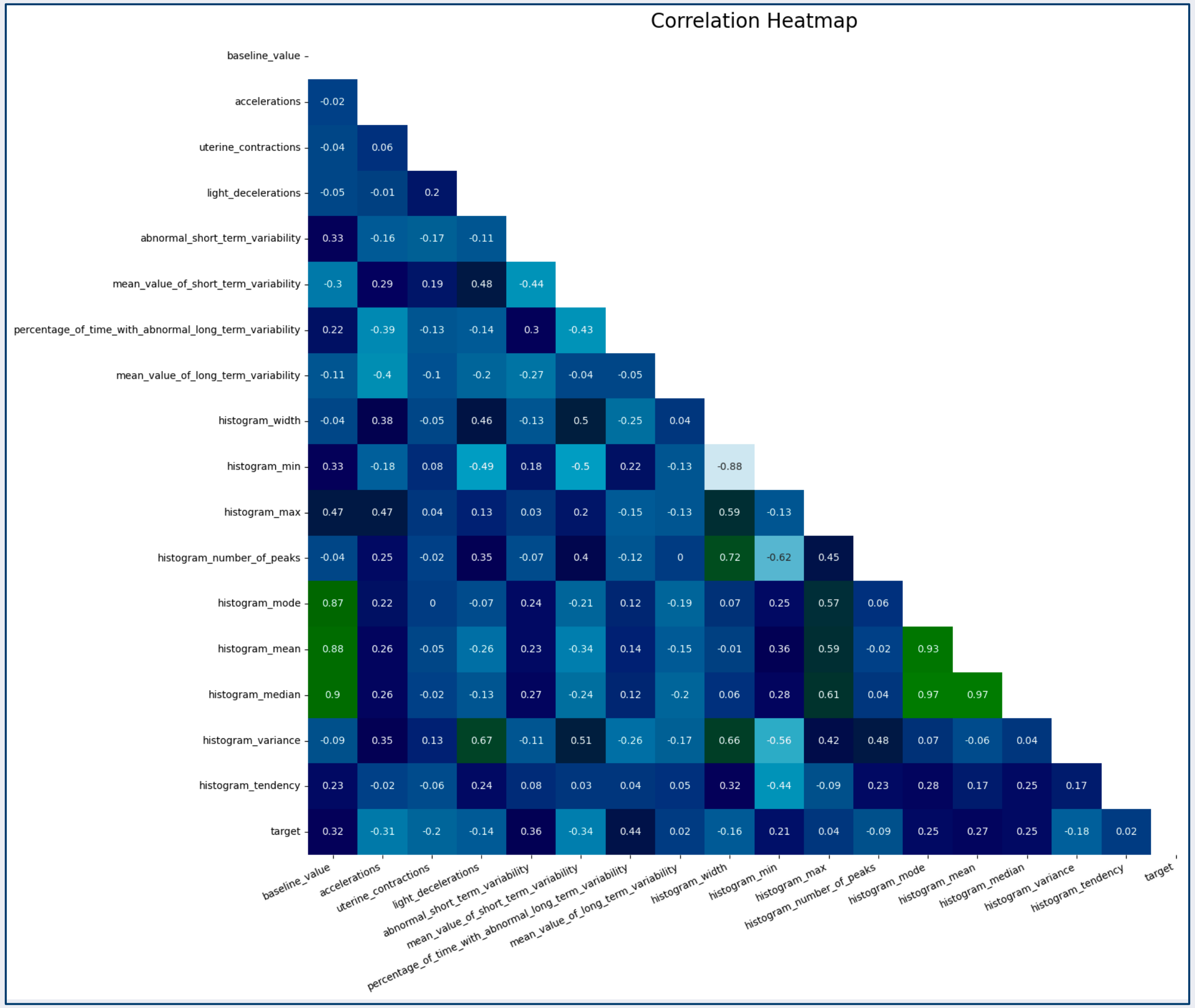
- Imbalanced Output Classes
- False Negatives are more harmful than False Positives
- ROC Area Under Curve & Recall are better measures than Accuracy

Exploratory Data Analysis

- No missing values
- Several features with many outliers
- Outliers removed
  - Below Q1 – 1.5 × IQR
  - Above Q3 +1.5 × IQR



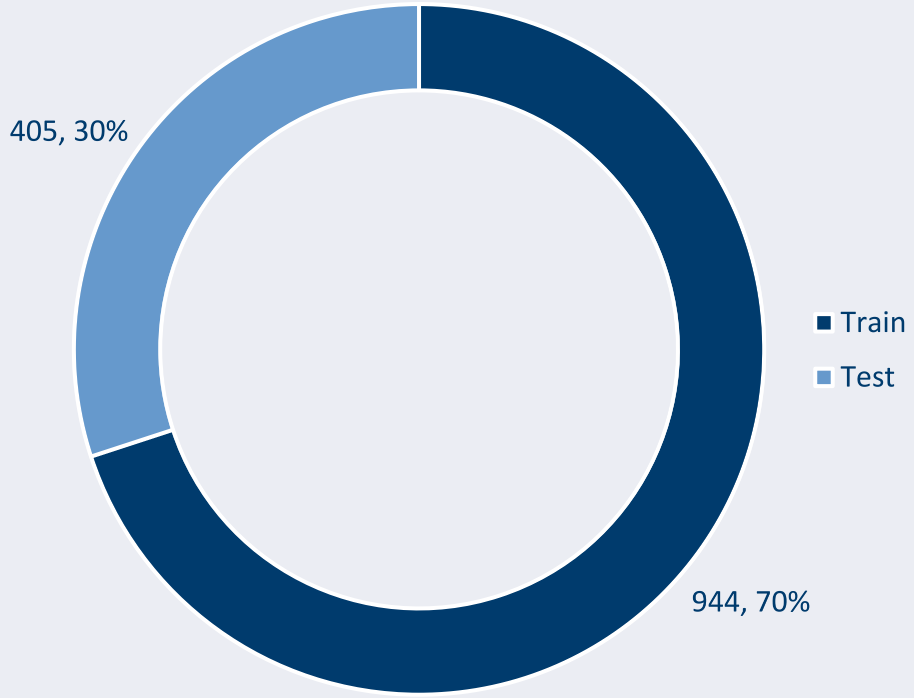
Exploratory Data Analysis



- Highly correlated features
- Principal Component Analysis used to extract seven features
- Cumulative explained variance: 99.2%

Data Scaling & Splitting

- Min-max normalization used to maintain original variance
- Observations randomly split between Train (70%) and Test (30%)



Model Building & Optimization

Random Forest

- Initial model is overfitted
- Poor Recall & F1-Scores, likely due to output class imbalance

Baseline Model Scores

Dataset	Accuracy	ROC AUC	Recall	F1-Score
Train	1.000	1.000	1.000	1.000
Test	0.948	0.880	0.796	0.677

```
# Random Forest
rf_clf = RandomForestClassifier(random_state=36)
rf_clf.fit(X_train, y_train)
```

```
# Random Forest
rf = RandomForestClassifier(random_state=36)
rf_pg = {
    'n_estimators': [50, 70, 100],
    'max_depth': [2, 3, 5],
    'oob_score': [True, False],
    'class_weight': [{0:.09, 1:.91}, {0:.1, 1:.9}]
}
rf_grid = GridSearchCV(rf, param_grid=rf_pg, cv=10, scoring='recall')
with joblib.parallel_backend('threading', n_jobs=4):
    rf_grid.fit(X_train, y_train)
```

Model Building & Optimization

Logistic Regression

- Poor Recall & F1-Scores, likely due to output class imbalance

```
# Logistic Regression
log_reg = LogisticRegression(solver='lbfgs',
                             max_iter=1000,
                             random_state=36)
log_reg.fit(X_train, y_train)
```

Baseline Model Scores

Dataset	Accuracy	ROC AUC	Recall	F1-Score
Train	0.912	0.699	0.427	0.515
Test	0.928	0.723	0.472	0.540

```
# Logistic Regression GridSearchCV
lr = LogisticRegression(max_iter=5000, random_state=36)
lr_pg = {
    'penalty': ['l1','l2'],
    'solver': ['liblinear','saga'],
    'class_weight': [{0:.01, 1:.99}, {0:.1, 1:.9}, {0:.15, 1:.85}],
    'C': [0.01, 0.05, 0.1, 0.25, 0.5, 0.75, 1]
}
lr_grid = GridSearchCV(lr, param_grid=lr_pg, cv=10, scoring='roc_auc')
lr_grid.fit(X_train, y_train)
```

science.engineering.analytics.insights.
DATA

Model Building & Optimization

K-Nearest Neighbors

- Poor Recall & F1-Scores, likely due to output class imbalance

```
# K-Nearest Neighbors
knn_clf = KNeighborsClassifier()
knn_clf.fit(X_train, y_train)
```

Baseline Model Scores

Dataset	Accuracy	ROC AUC	Recall	F1-Score
Train	0.932	0.762	0.544	0.636
Test	0.926	0.734	0.500	0.734

```
# KNN
knn = KNeighborsClassifier()
knn_pg = {
    'n_neighbors': [ i for i in range(5,15) ],
    'leaf_size': [10,30,50],
    'algorithm': ['ball_tree','kd_tree'],
    'p': [1,2],
    'metric':['minkowski','chebyshev']
}
knn_grid = GridSearchCV(knn, param_grid=knn_pg, cv=10, scoring='recall')
with joblib.parallel_backend('threading', n_jobs=4):
    knn_grid.fit(X_train, y_train)
```

Model Comparison

Training Data Final Scores

Model	Accuracy	ROC AUC	Recall	F1-Score
Random Forest	0.854	0.895	0.944	0.535
Logistic Regression	0.877	0.882	0.889	0.561
K-Nearest Neighbors	0.931	0.711	0.444	0.533

Test Data Final Scores

Model	Accuracy	ROC AUC	Recall	F1-Score
Random Forest	0.854	0.895	0.944	0.535
Logistic Regression	0.877	0.882	0.889	0.561
K-Nearest Neighbors	0.931	0.711	0.444	0.533

- False Negatives are more harmful than False Positives
- ROC AUC & Recall are most relevant metrics
- Random Forest outperforms other models
- Initial overfitting resolved by hyperparameter tuning

Discussion & Lessons Learned

- Random Forest is the best model with the highest ROC AUC and Recall.
  - Test ROC AUC: 89.5%
  - Test Recall: 94.4%

- Most influential variables in determining a Normal vs. Abnormal CTG:
  - FHR Histogram Mode
  - FHR Abnormal Short-Term Variability
  - FHR Percentage of Time with Abnormal Long-Term Variability

- Recommend increasing sample size in future research due to large number of outliers.

References

Ayres-de-Campos, D., Bernardes, J., Garrido, A., Marques-de-Sá, J., & Pereira-Leite, L. (2000). SisPorto 2.0: A Program for Automated Analysis of Cardiotocograms. *The Journal of Maternal-Fetal Medicine*. 9(5), 311-318. https://doi.org/10.1002/1520-6661(200009/10)9:5%3C311::AID-MFM12%3E3.0.CO;2-9

Potter, L. (2021, November 12). How to Read a CTG. *Geeky Medics*. https://geekymedics.com/how-to-read-a-ctg/

Prior, L, & Lees, C. (2019). Control and Monitoring of Fetal Growth. *Encyclopedia of Endocrine Diseases (Second Edition)*. https://www.sciencedirect.com/topics/nursing-and-health-professions/cardiotocograph