Developing a fetal classifier.

A data science project by Kenneth Imade

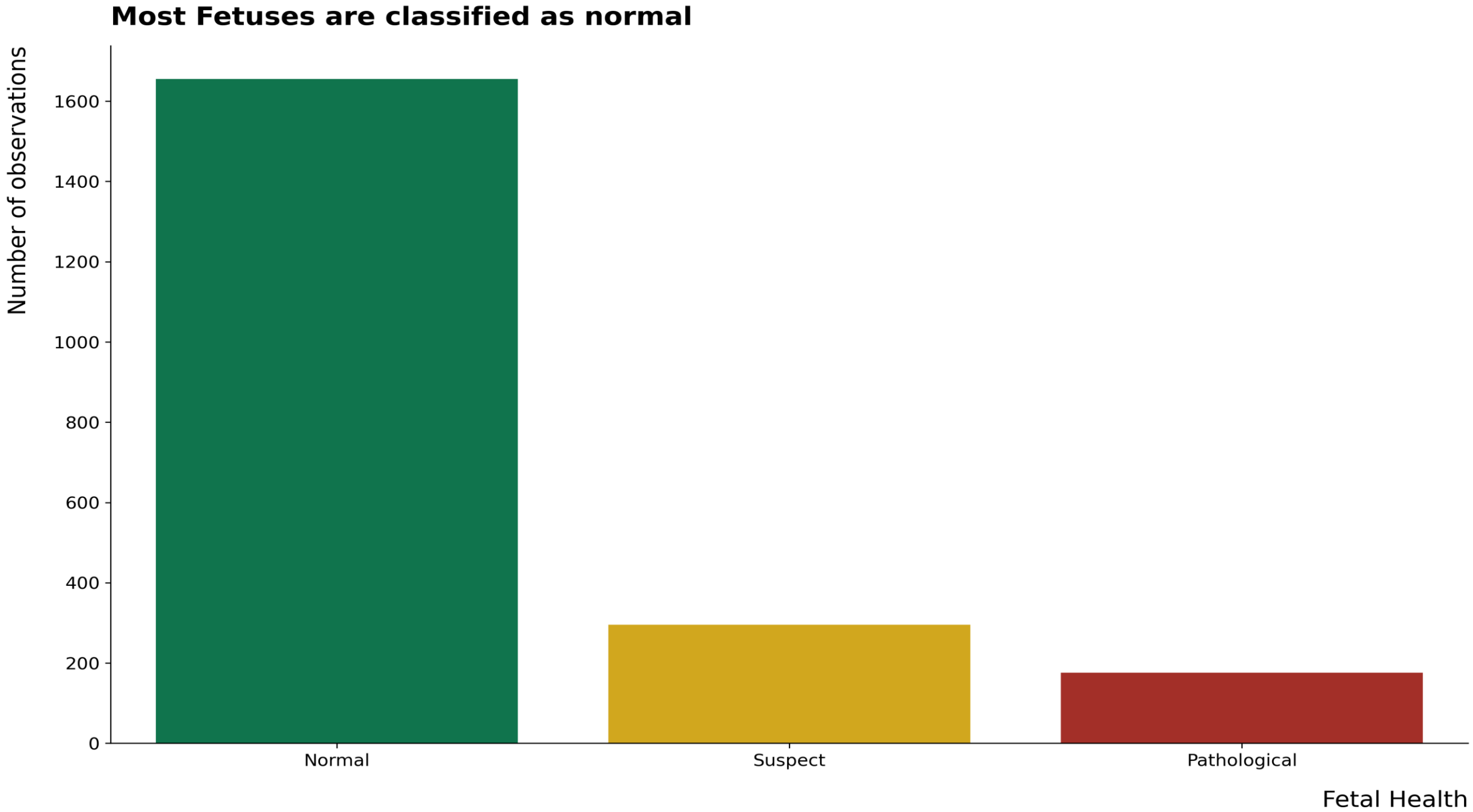
Photo by Volodymyr Hryshchenko on Unsplash

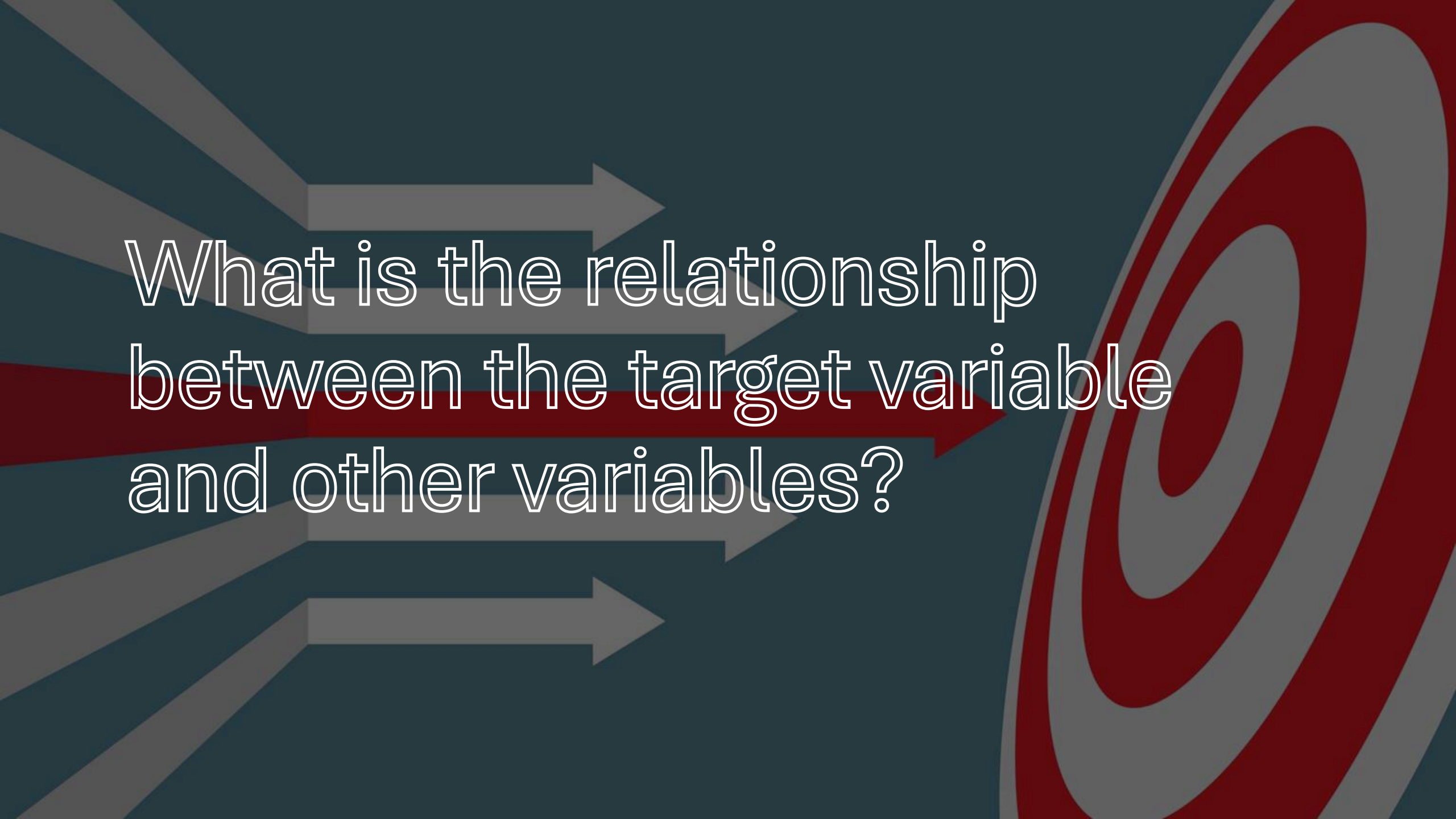
Objectives

- To understand the features that determine the health status of a fetus
- To develop a machine learning model capable of classifying fetus health into three categories; normal, suspect, or pathological

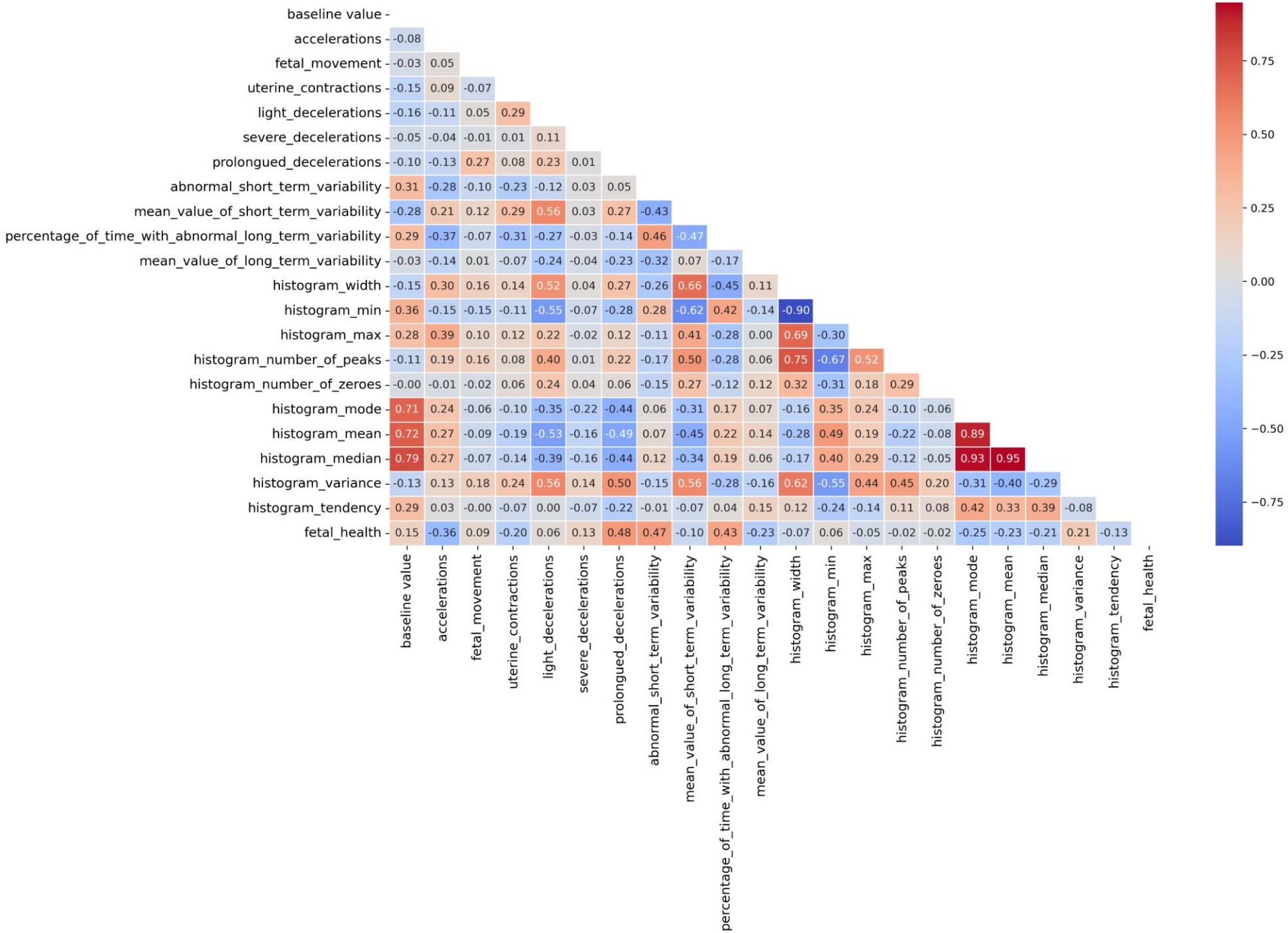
Exploratory Data Analysis


What is the distribution
of the target variable?

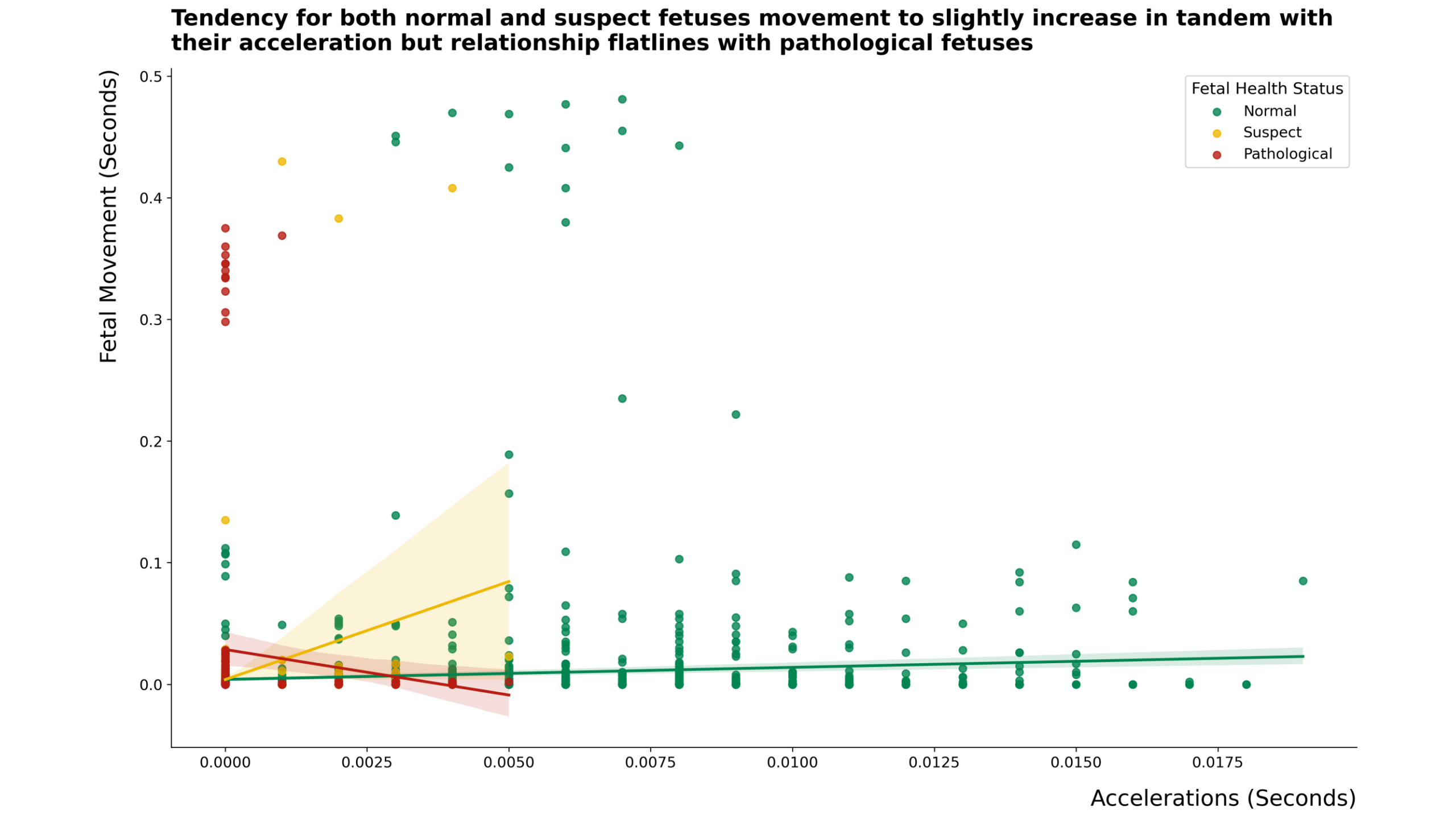

Most Fetuses are classified as normal
Number of observations
1600
1400
1200
1000
800
600
400
200
0
Normal
Suspect
Pathological
Fetal Health

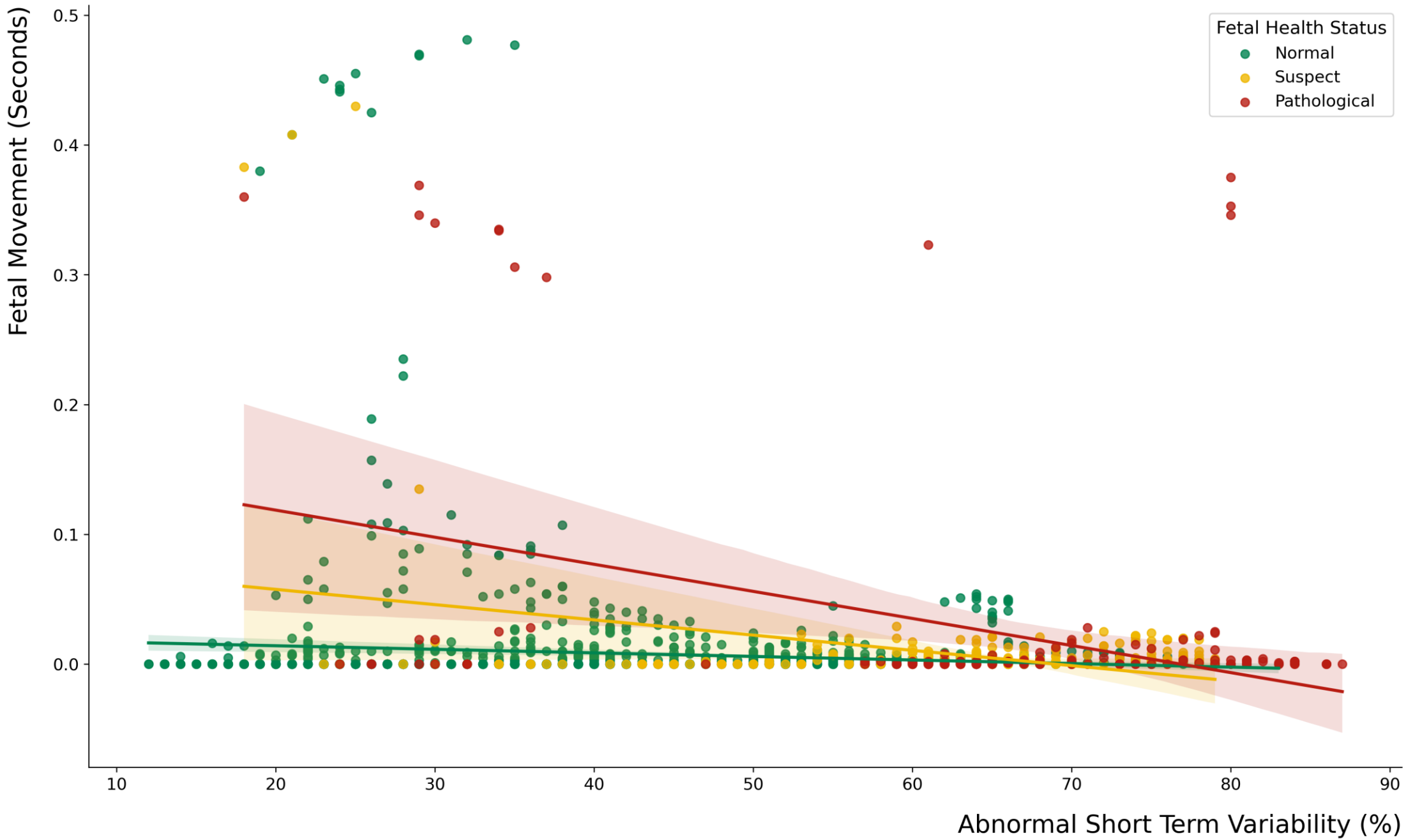

What is the relationship between the target variable and other variables?

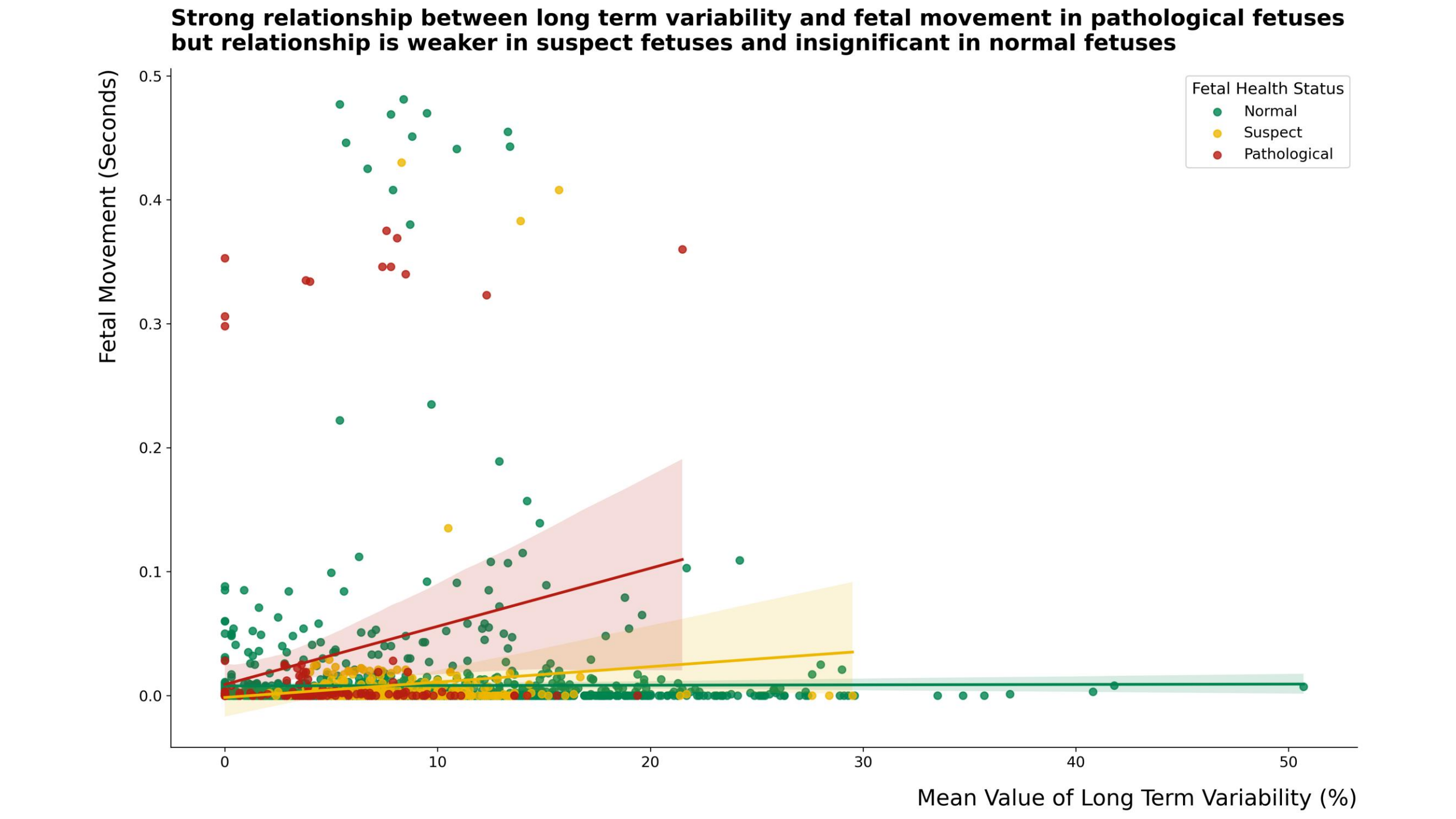
Strong correlation exists between fetal health and other factors

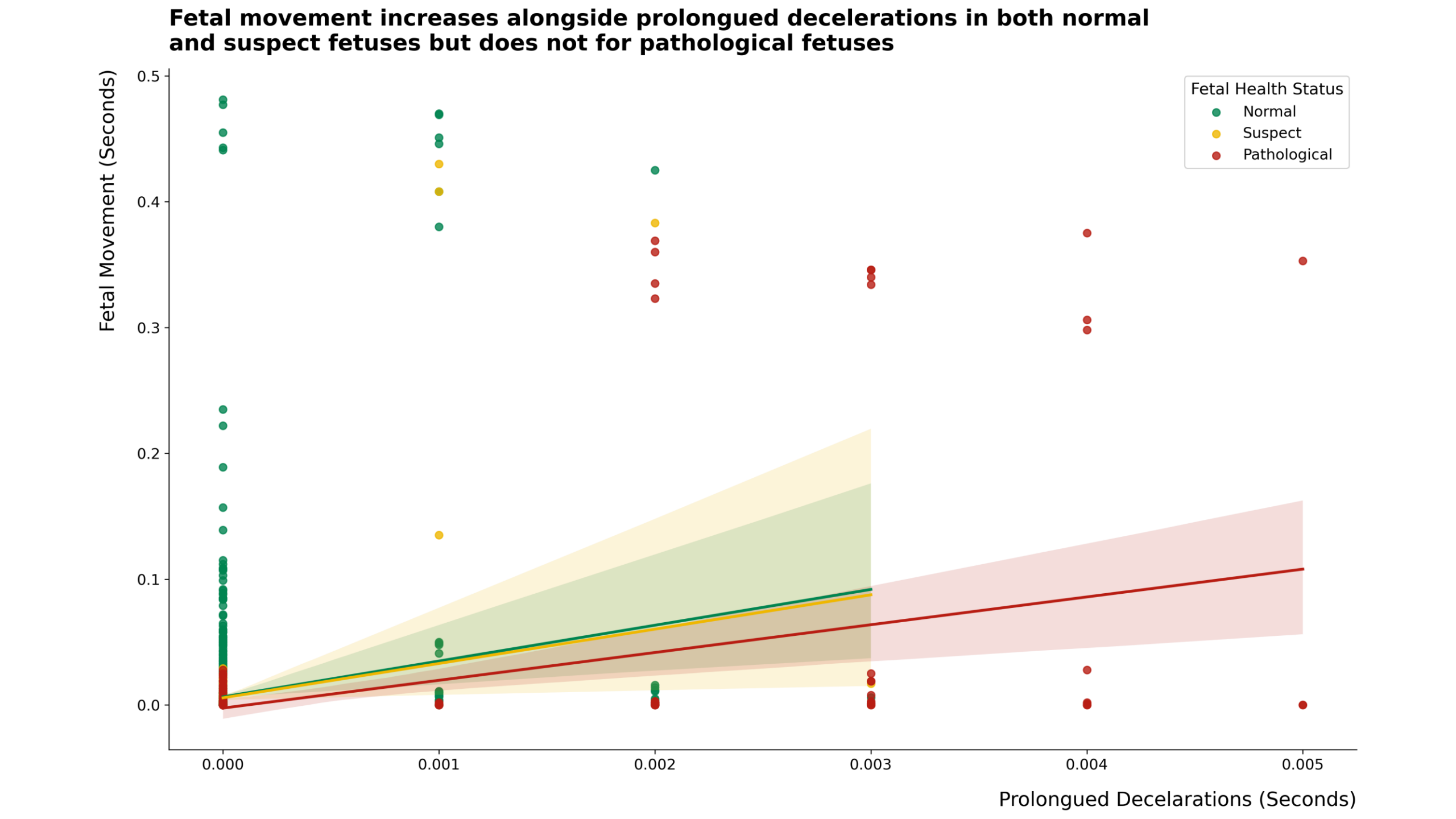


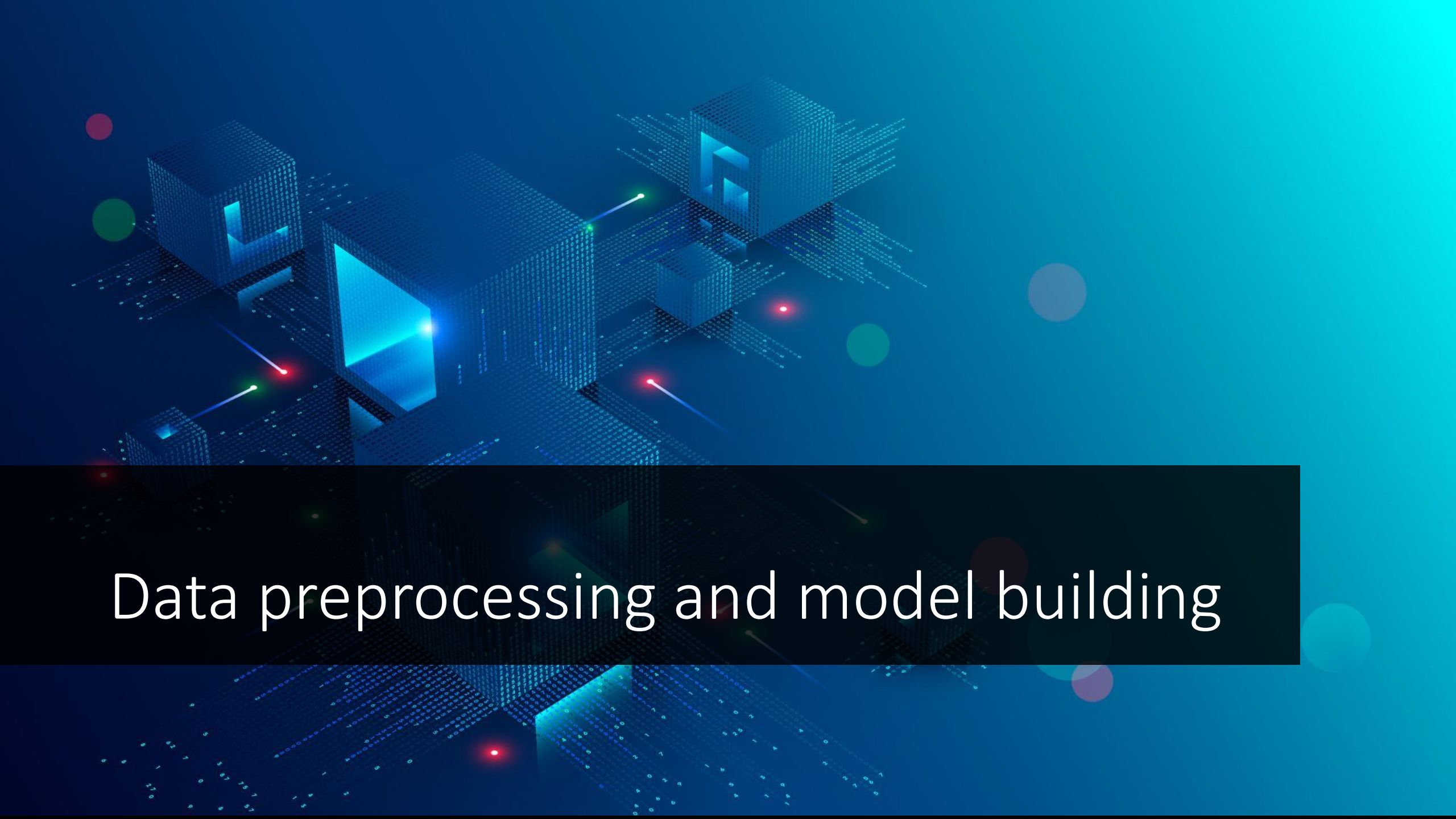

How does Fetal health change with respect to these variables?


Tendency for both normal and suspect fetuses movement to slightly increase in tandem with their acceleration but relationship flatlines with pathological fetuses
Fetal Movement (Seconds)
0.5
0.4
0.3
0.2
0.1
0.0
0.0000
0.0025
0.0050
0.0075
0.0100
0.0125
0.0150
0.0175
Accelerations (Seconds)
Fetal Health Status
Normal
Suspect
Pathological

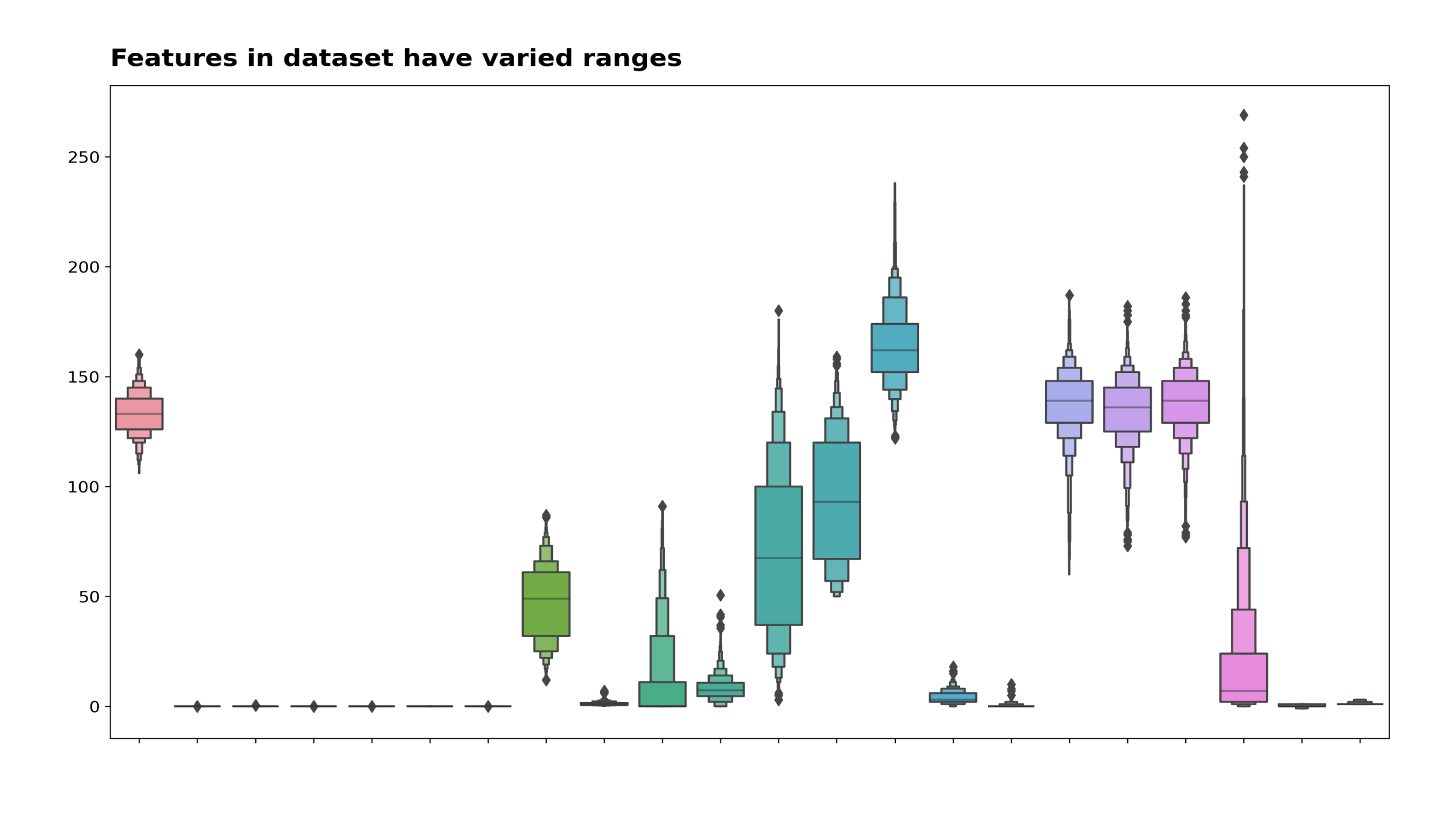

Increased short term fetal heart rate suggests a decrease in fetal movement across all fetal health categories
Fetal Movement (Seconds)
0.5
0.4
0.3
0.2
0.1
0.0
Fetal Health Status
Normal
Suspect
Pathological
10
20
30
40
50
60
70
80
90
Abnormal Short Term Variability (%)


Strong relationship between long term variability and fetal movement in pathological fetuses but relationship is weaker in suspect fetuses and insignificant in normal fetuses
Fetal Movement (Seconds)
0.5
0.4
0.3
0.2
0.1
0.0
0
10
20
30
40
50
Mean Value of Long Term Variability (%)
Fetal Health Status
Normal
Suspect
Pathological

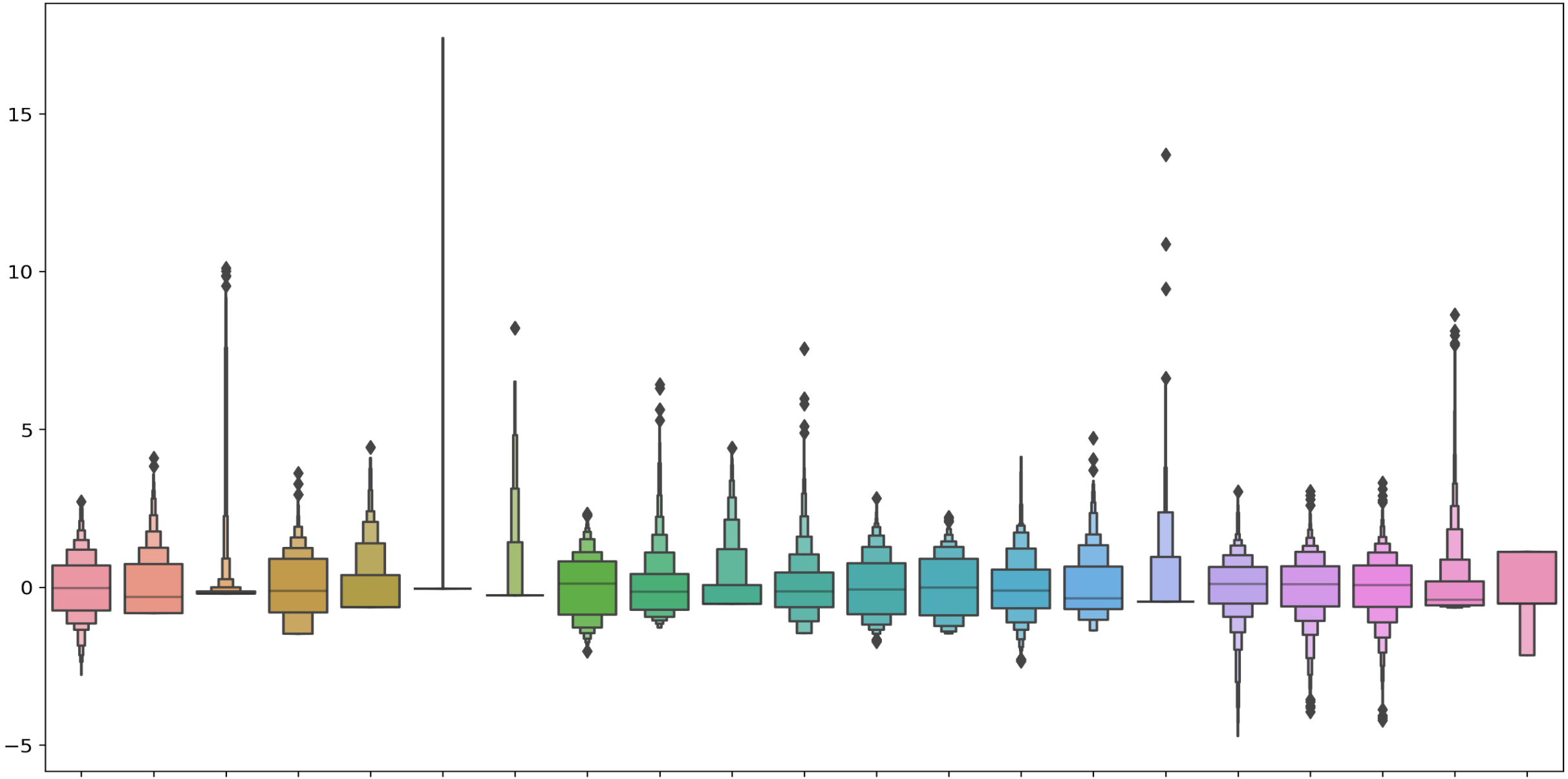

Fetal movement increases alongside prolongued decelerations in both normal and suspect fetuses but does not for pathological fetuses
Fetal Movement (Seconds)
0.5
0.4
0.3
0.2
0.1
0.0
Fetal Health Status
Normal
Suspect
Pathological
0.000
0.001
0.002
0.003
0.004
0.005
Prolongued Decelarations (Seconds)



Data preprocessing and model building


Features in dataset have varied ranges
250
200
150
100
50
0

Feature ranges standardised for model building



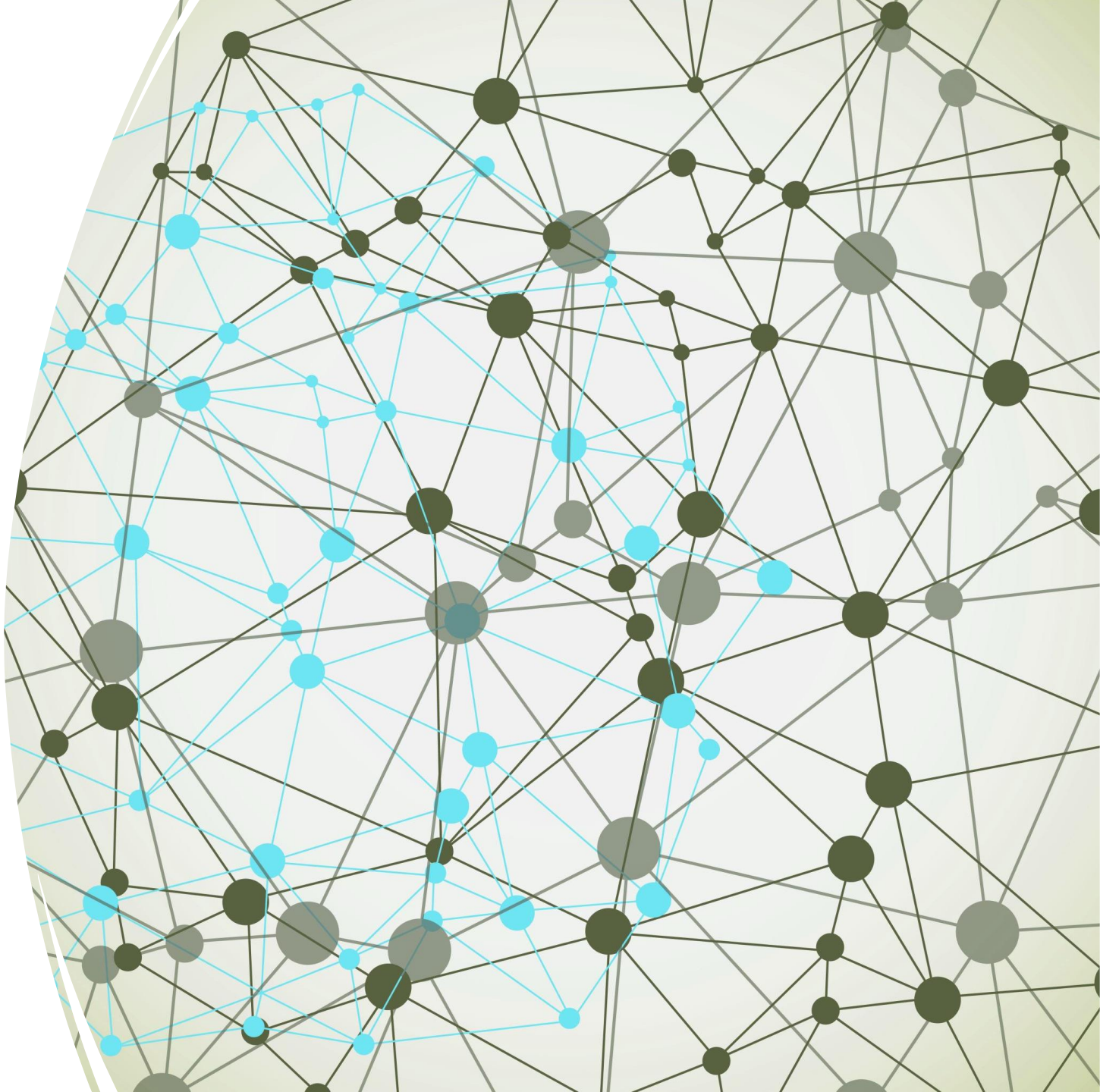
Choosing the best model

```
lr_model = Pipeline([('lr_classifier', LogisticRegression(random_state=42))])
dt_model = Pipeline([('dt_classifier', DecisionTreeClassifier(random_state=42))])
rf_model = Pipeline([('rf_classifier', RandomForestClassifier(random_state=42))])
knn_model = Pipeline([('knn_classifier', KNeighborsClassifier())])
svc_model = Pipeline([('svc_classifier', SVC())])

pipelines = [lr_model, dt_model, rf_model, knn_model, svc_model]
pipe_dict = {0: 'Logistic Regression', 1: 'Decision Tree', 2: 'Random Forest', 3: 'KNN', 4: 'SVC'}

for pipe in pipelines:
    pipe.fit(X_train, y_train)

#cross validation on accuracy
cv_results_accuracy = []
for i, model in enumerate(pipelines):
    cv_score = cross_val_score(model, X_train,y_train, cv=10 )
    cv_results_accuracy.append(cv_score)
    print("%s: %f " % (pipe_dict[i], cv_score.mean()))
```

Hyper-parameter tuning.

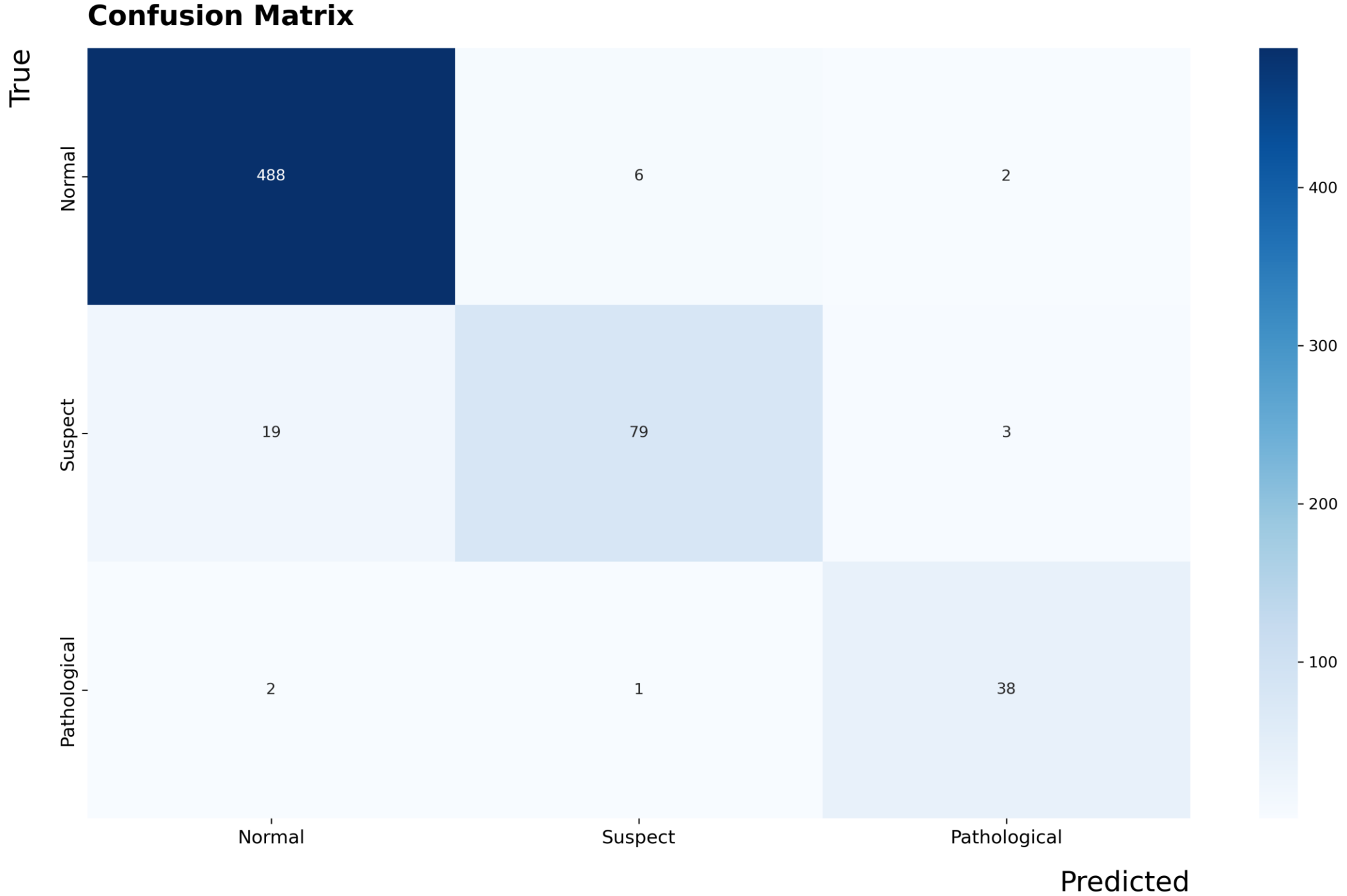
```
parameters = {
    'n_estimators': [100, 150, 200, 500, 700, 900],
    'max_features': ['auto', 'sqrt', 'log2'],
    'max_depth': [4,6,8,12,14,16],
    'criterion': ['gini', 'entropy'],
}

rfc_cv = GridSearchCV(estimator=RandomForestClassifier(), param_grid=parameters, cv= 5)
rfc_cv.fit(X_train, y_train)

rfc_cv.best_params_
```

Results



Classification Metrics

	Precision	Recall	F1-Score	Support
Normal (1.0)	0.96	0.98	0.97	496
Suspect (2.0)	0.92	0.78	0.84	101
Pathological (3.0)	0.88	0.93	0.90	41
Accuracy			0.95	638
Macro Avg	0.92	0.90	0.91	638
Weighted Avg	0.95	0.95	0.95	638


Confusion Matrix
True
Normal
Suspect
Pathological
488
6
2
19
79
3
2
1
38
Normal
Suspect
Pathological
Predicted
400
300
200
100