



FETAL HEALTH

CLASSIFICATION USING

RANDOM FOREST

FELICIA SALIM



A solid yellow rectangular bar.

OBJECTIVE

This project uses the Fetal Health dataset from Kaggle to build a classification model that classifies fetal health status into three target classes: Normal, Suspicious, and Pathological. The dataset is made up of 2126 records and 22 features.

Dataset: <https://www.kaggle.com/datasets/andrewmvd/fetal-health-classification>



TECH STACK



LOAD DATASET

```
#Load dataset
import pandas as pd

fetal_health = pd.read_csv('fetal_health.csv')

x = fetal_health.drop(['fetal_health'], axis=1) #features
y = fetal_health['fetal_health'] #target class

print('Dataframe: ', fetal_health.shape)
fetal_health.head(10)
```



Dataframe: (2126, 22)

	baseline value	accelerations	fetal_movement	uterine_contractions	light_decelerations	severe_decelerations	prolongued_decelerations	abnormal_short_term_variability
0	120.0	0.000	0.0	0.000	0.000	0.0	0.000	73.0
1	132.0	0.006	0.0	0.006	0.003	0.0	0.000	17.0
2	133.0	0.003	0.0	0.008	0.003	0.0	0.000	16.0
3	134.0	0.003	0.0	0.008	0.003	0.0	0.000	16.0
4	132.0	0.007	0.0	0.008	0.000	0.0	0.000	16.0
5	134.0	0.001	0.0	0.010	0.009	0.0	0.002	26.0
6	134.0	0.001	0.0	0.013	0.008	0.0	0.003	29.0
7	122.0	0.000	0.0	0.000	0.000	0.0	0.000	83.0
8	122.0	0.000	0.0	0.002	0.000	0.0	0.000	84.0
9	122.0	0.000	0.0	0.003	0.000	0.0	0.000	86.0

10 rows × 22 columns

DATASET OVERVIEW

```
fetal_health.info()
```



```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2126 entries, 0 to 2125
Data columns (total 22 columns):
#   Column                                                                 Non-Null Count  Dtype
---  -
0   baseline value                                                         2126 non-null   float64
1   accelerations                                                         2126 non-null   float64
2   fetal_movement                                                         2126 non-null   float64
3   uterine_contractions                                                  2126 non-null   float64
4   light_decelerations                                                   2126 non-null   float64
5   severe_decelerations                                                  2126 non-null   float64
6   prolonged_decelerations                                               2126 non-null   float64
7   abnormal_short_term_variability                                       2126 non-null   float64
8   mean_value_of_short_term_variability                                  2126 non-null   float64
9   percentage_of_time_with_abnormal_long_term_variability              2126 non-null   float64
10  mean_value_of_long_term_variability                                    2126 non-null   float64
11  histogram_width                                                        2126 non-null   float64
12  histogram_min                                                          2126 non-null   float64
13  histogram_max                                                          2126 non-null   float64
14  histogram_number_of_peaks                                              2126 non-null   float64
15  histogram_number_of_zeroes                                             2126 non-null   float64
16  histogram_mode                                                         2126 non-null   float64
17  histogram_mean                                                         2126 non-null   float64
18  histogram_median                                                       2126 non-null   float64
19  histogram_variance                                                     2126 non-null   float64
20  histogram_tendency                                                     2126 non-null   float64
21  fetal_health                                                           2126 non-null   float64
dtypes: float64(22)
memory usage: 365.5 KB
```

DATASET OVERVIEW

```
fetal_health['fetal_health'].value_counts()
```



count	
fetal_health	
1.0	1655
2.0	295
3.0	176

dtype: int64

1.0 - Normal
2.0 - Suspicious
3.0 - Pathological

NULL VALUE CHECK

```
#Check for null values
fetal_health.isnull().sum()
```



	0
baseline value	0
accelerations	0
fetal_movement	0
uterine_contractions	0
light_decelerations	0
severe_decelerations	0
prolongued_decelerations	0
abnormal_short_term_variability	0
mean_value_of_short_term_variability	0
percentage_of_time_with_abnormal_long_term_variability	0
mean_value_of_long_term_variability	0
histogram_width	0
histogram_min	0
histogram_max	0
histogram_number_of_peaks	0
histogram_number_of_zeroes	0
histogram_mode	0
histogram_mean	0
histogram_median	0
histogram_variance	0
histogram_tendency	0
fetal_health	0

dtype: int64

No null values in
this dataset



STATISTICAL SUMMARY

```
fetal_health.describe()
```



	baseline value	accelerations	fetal_movement	uterine_contractions	light_decelerations	severe_decelerations	prolongued_decelerations	abnormal_short_term_variability	mean_value_of_short_term_variability	percentage_of_time_wit
count	2126.000000	2126.000000	2126.000000	2126.000000	2126.000000	2126.000000	2126.000000	2126.000000	2126.000000	
mean	133.303857	0.003178	0.009481	0.004366	0.001889	0.000003	0.000159	46.990122	1.332785	
std	9.840844	0.003866	0.046666	0.002946	0.002960	0.000057	0.000590	17.192814	0.883241	
min	106.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	12.000000	0.200000	
25%	126.000000	0.000000	0.000000	0.002000	0.000000	0.000000	0.000000	32.000000	0.700000	
50%	133.000000	0.002000	0.000000	0.004000	0.000000	0.000000	0.000000	49.000000	1.200000	
75%	140.000000	0.006000	0.003000	0.007000	0.003000	0.000000	0.000000	61.000000	1.700000	
max	160.000000	0.019000	0.481000	0.015000	0.015000	0.001000	0.005000	87.000000	7.000000	

8 rows x 22 columns

MODEL TRAINING, TESTING & EVALUATION

```
#Split Data
from sklearn.model_selection import train_test_split

x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.2, random_state=42)
```

```
from sklearn.ensemble import RandomForestClassifier

#training
model = RandomForestClassifier()
model.fit(x_train, y_train)
```

```
#testing
y_pred = model.predict(x_test)
```

```
from sklearn.metrics import accuracy_score, confusion_matrix, classification_report

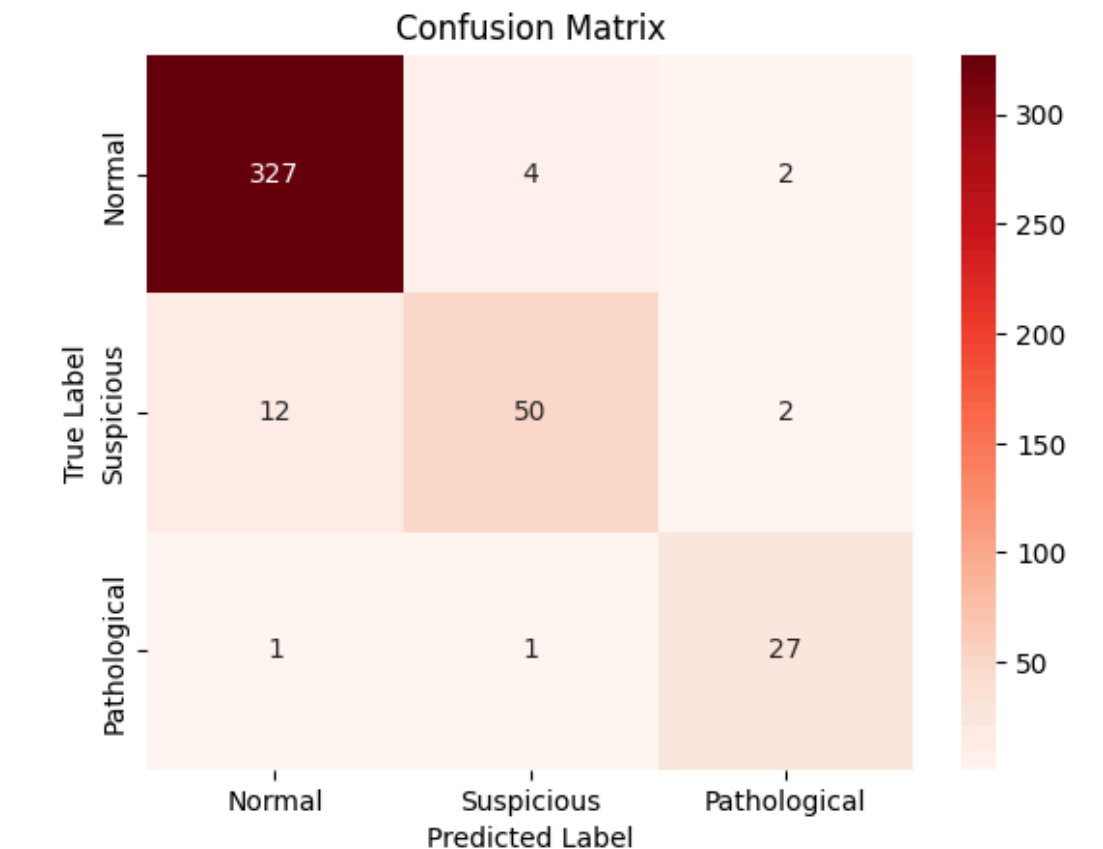
print('Accuracy Score: ', accuracy_score(y_test, y_pred))

sns.heatmap(confusion_matrix(y_test, y_pred), annot=True, fmt='d', cmap='Reds',
            xticklabels=['Normal', 'Suspicious', 'Pathological'],
            yticklabels=['Normal', 'Suspicious', 'Pathological'])
plt.title('Confusion Matrix')
plt.xlabel('Predicted Label')
plt.ylabel('True Label')
plt.show()

print('Classification Report: \n', classification_report(y_test, y_pred))
```



Accuracy Score: 0.9483568075117371



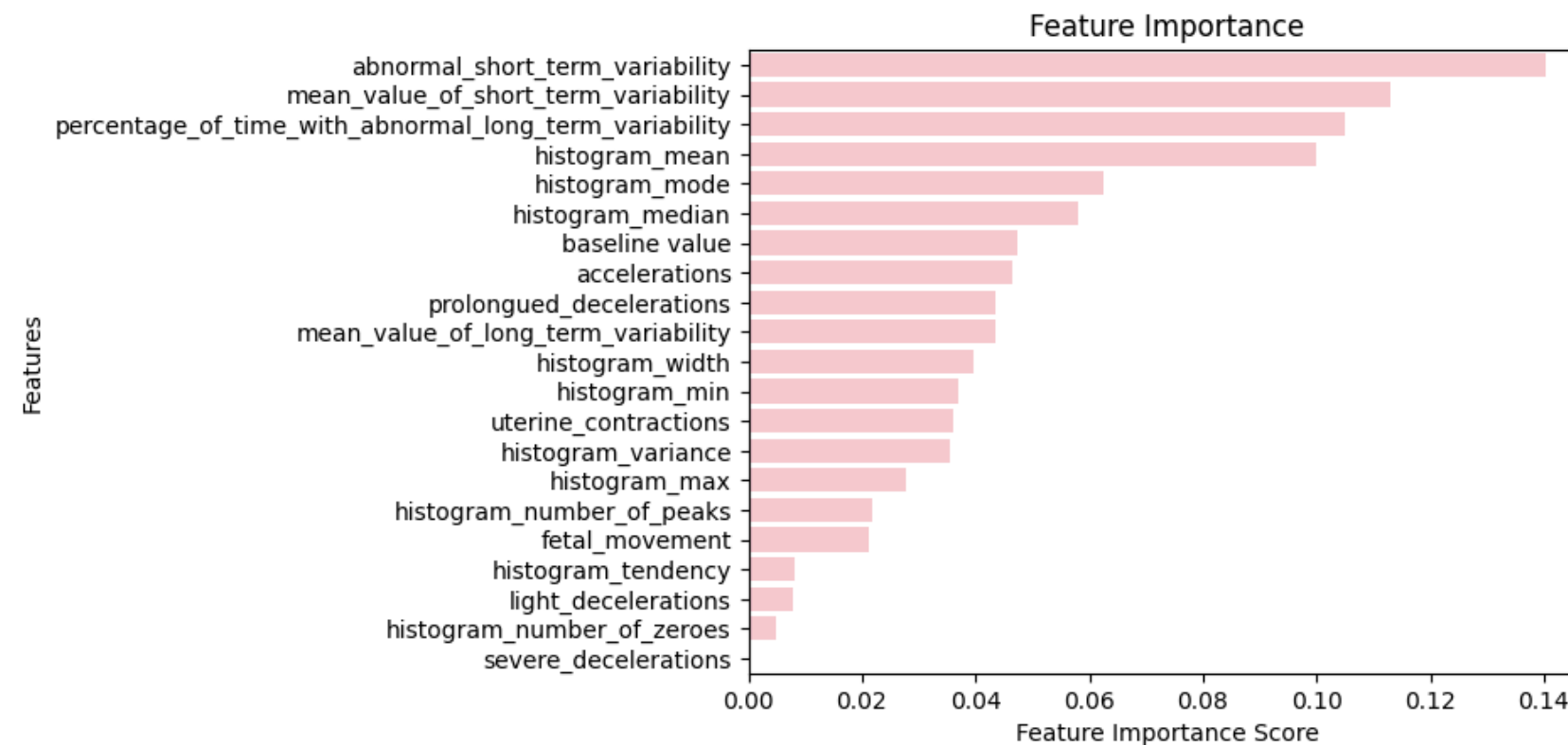
Classification Report:

	precision	recall	f1-score	support
1.0	0.96	0.98	0.97	333
2.0	0.91	0.78	0.84	64
3.0	0.87	0.93	0.90	29
accuracy			0.95	426
macro avg	0.91	0.90	0.90	426
weighted avg	0.95	0.95	0.95	426

MODEL TRAINING, TESTING & EVALUATION

```
feature_importance = model.feature_importances_
importance = pd.Series(feature_importance, index=x_train.columns).sort_values(ascending=False)
sns.barplot(x=importance, y=importance.index, color='pink')
plt.xlabel('Feature Importance Score')
plt.ylabel('Features')
plt.title("Feature Importance")
plt.show()
```

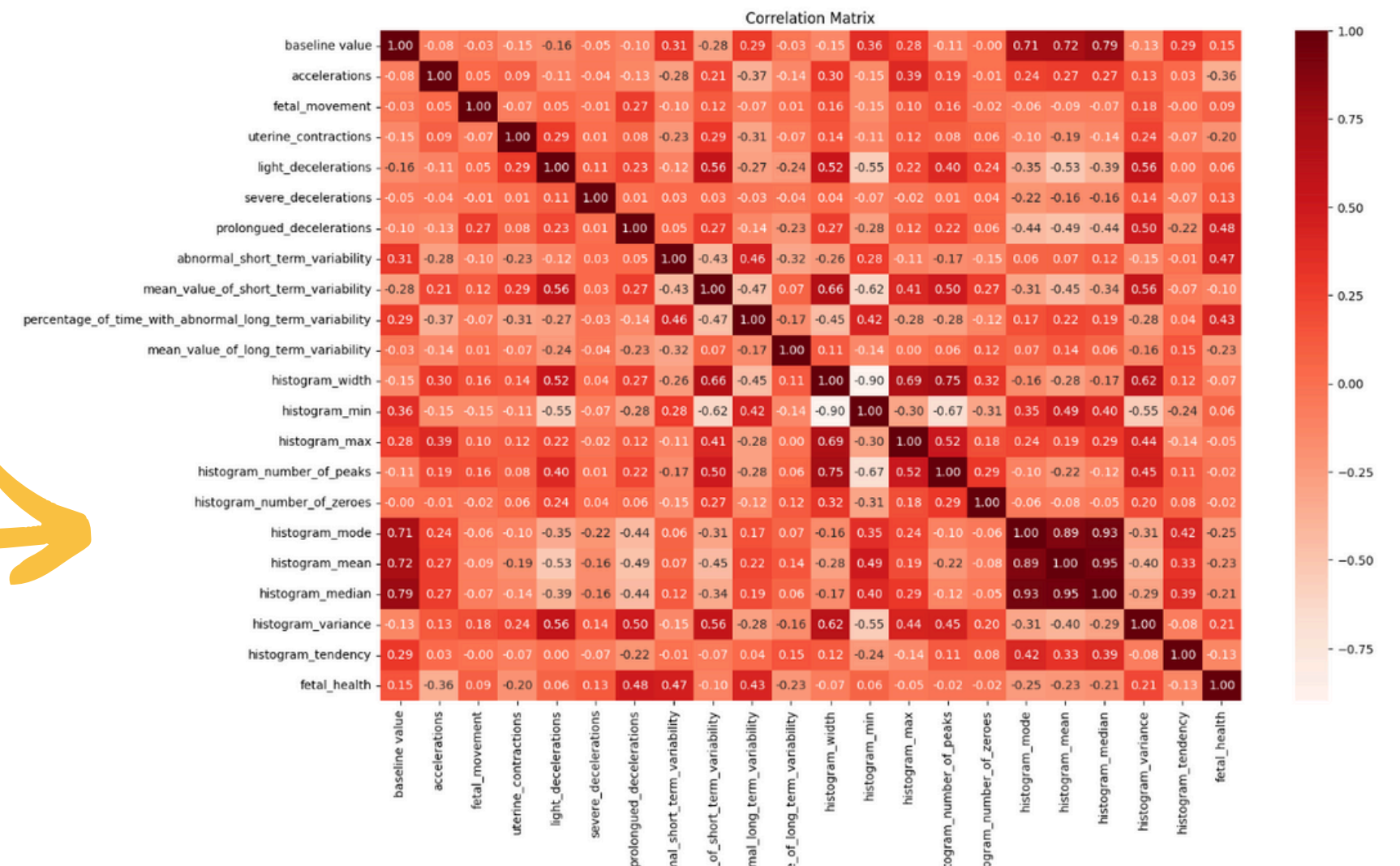
Feature Importance refers to how valuable each feature is in predicting the target variable. Higher feature importance score means that the feature strongly impacts the prediction model.



MODEL TRAINING, TESTING & EVALUATION

```
import seaborn as sns
import matplotlib.pyplot as plt

#Correlation Matrix
plt.figure(figsize=(16, 10))
sns.heatmap(fetal_health.corr(), annot=True, fmt='.2f', cmap='Reds')
plt.title('Correlation Matrix')
plt.show()
```



Correlation Matrix shows the relationships between a feature with another feature.



CONCLUSION

Result shows that there are 404 correctly predicted value out of 426 testing data:

- 327 data predicted as normal,
- 50 data predicted as suspicious, and
- 27 data predicted as pathological.

The Random Forest model shows an accuracy score of 94.8%



THANK YOU

