

Fetal Health Data Set-Model Tuning

```
In [2]: import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

In [3]: data=pd.read_csv('C:\\Users\\User\\Desktop\\Fetal_health.csv')

In [4]: data.head()
```

	baseline_value	accelerations	fetal_movement	uterine_contractions	light_decelerations	severe_decelerations	prolongued_decelerations	abnormal_short_term_variability	mean_value_of_short_term_variability
0	120.0	0.000	0.0	0.000	0.000	0.0	0.0	73.0	0.1
1	132.0	0.006	0.0	0.006	0.003	0.0	0.0	17.0	2.1
2	133.0	0.003	0.0	0.008	0.003	0.0	0.0	16.0	2.1
3	134.0	0.003	0.0	0.008	0.003	0.0	0.0	16.0	2.1
4	132.0	0.007	0.0	0.008	0.000	0.0	0.0	16.0	2.1

5 rows × 22 columns

```
In [5]: data.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

In [6]: data.describe()
```

	baseline_value	accelerations	fetal_movement	uterine_contractions	light_decelerations	severe_decelerations	prolongued_decelerations	abnormal_short_term_variability	mean_value_of_short_term_variability
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	0.11
std	9.940844	0.003866	0.046666	0.002946	0.002960	0.000057	0.000590	17.192814	0.11
min	106.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	12.000000	0.11
25%	126.000000	0.000000	0.000000	0.002000	0.000000	0.000000	0.000000	32.000000	0.11
50%	133.000000	0.002000	0.000000	0.004000	0.000000	0.000000	0.000000	49.000000	0.11
75%	140.000000	0.006000	0.003000	0.007000	0.003000	0.000000	0.000000	61.000000	0.11
max	160.000000	0.019000	0.481000	0.015000	0.015000	0.001000	0.005000	87.000000	0.11

8 rows × 22 columns

```
In [7]: # Checking null vlaues
data.isnull().sum()
```

	baseline_value	accelerations	fetal_movement	uterine_contractions	light_decelerations	severe_decelerations	prolongued_decelerations	abnormal_short_term_variability	mean_value_of_short_term_variability
count	2126	2126	2126	2126	2126	2126	2126	2126	2126
mean	133.303857	0.003178	0.009481	0.004366	0.001889	0.000003	0.000159	46.990122	0.11
std	9.940844	0.003866	0.046666	0.002946	0.002960	0.000057	0.000590	17.192814	0.11
min	106.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	12.000000	0.11
25%	126.000000	0.000000	0.000000	0.002000	0.000000	0.000000	0.000000	32.000000	0.11
50%	133.000000	0.002000	0.000000	0.004000	0.000000	0.000000	0.000000	49.000000	0.11
75%	140.000000	0.006000	0.003000	0.007000	0.003000	0.000000	0.000000	61.000000	0.11
max	160.000000	0.019000	0.481000	0.015000	0.015000	0.001000	0.005000	87.000000	0.11

8 rows × 22 columns

```
In [7]: # Checking null vlaues
data.isnull().sum()

Out[7]: 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

In [8]: #Checking duplicates
duplicate = data[data.duplicated()]

In [9]: print(duplicate)
```

	baseline_value	accelerations	fetal_movement	uterine_contractions	light_decelerations	severe_decelerations	prolongued_decelerations	abnormal_short_term_variability	mean_value_of_short_term_variability
68	149.0	0.007	0.000	0.000	0.004	0.0	0.0	73.0	0.1
234	123.0	0.000	0.000	0.000	0.000	0.0	0.0	17.0	2.1
306	145.0	0.000	0.020	0.000	0.000	0.0	0.0	16.0	2.1
324	135.0	0.000	0.000	0.000	0.000	0.0	0.0	16.0	2.1
333	144.0	0.000	0.019	0.000	0.000	0.0	0.0	16.0	2.1
787	123.0	0.003	0.003	0.000	0.000	0.0	0.0	16.0	2.1
791	123.0	0.003	0.004	0.000	0.000	0.0	0.0	16.0	2.1
798	146.0	0.000	0.000	0.000	0.000	0.0	0.0	16.0	2.1
849	138.0	0.002	0.000	0.000	0.004	0.0	0.0	16.0	2.1
1113	122.0	0.000	0.000	0.000	0.000	0.0	0.0	16.0	2.1
1114	122.0	0.000	0.000	0.000	0.000	0.0	0.0	16.0	2.1
1115	122.0	0.000	0.000	0.000	0.000	0.0	0.0	16.0	2.1
1458	148.0	0.005	0.000	0.000	0.002	0.0	0.0	16.0	2.1

13 rows × 22 columns

```
In [10]: data.drop_duplicates(inplace=True)

In [11]: data

Out[11]:
```

	baseline_value	accelerations	fetal_movement	uterine_contractions	light_decelerations	severe_decelerations	prolongued_decelerations	abnormal_short_term_variability	mean_value_of_short_term_variability
0	120.0	0.000	0.000	0.000	0.000	0.0	0.0	73.0	0.1
1	132.0	0.006	0.000	0.006	0.003	0.0	0.0	17.0	2.1
2	133.0	0.003	0.000	0.008	0.003	0.0	0.0	16.0	2.1
3	134.0	0.003	0.000	0.008	0.003	0.0	0.0	16.0	2.1
4	132.0	0.007	0.000	0.008	0.000	0.0	0.0	16.0	2.1
...
2121	140.0	0.000	0.000	0.007	0.000	0.0	0.0	79.0	0.1
2122	140.0	0.001	0.000	0.007	0.000	0.0	0.0	78.0	0.1
2123	140.0	0.001	0.000	0.007	0.000	0.0	0.0	79.0	0.1
2124	140.0	0.001	0.000	0.006	0.000	0.0	0.0	78.0	0.1
2125	142.0	0.002	0.002	0.008	0.000	0.0	0.0	74.0	0.1

2113 rows × 22 columns

Data Visualization and EDA

```
In [12]: data['fetal_health'].value_counts()
```

fetal_health	count
1	1646
2	292
3	0

Name: fetal_health, dtype: int64

```
In [13]: def data_plot(col):
sns.set_style(style='darkgrid')
sns.countplot(data=data,x=col)
plt.title('Sample of each class')
plt.show()

data_plot('fetal_health')
```

```
In [54]: #heat map
plt.figure(figsize=(10,5))
sns.heatmap(data_corr(),annot=True)

Out[54]:
```

```
In [14]: data.hist(figsize=(20,15),color='#d12f65')
plt.show()
```

```
In [15]: x=data.drop(['fetal_health'],axis=1)

In [16]: y=data['fetal_health']

In [17]: # split data into train and test
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test=train_test_split(x,y)

In [18]: from sklearn.svm import SVC
classifier=SVC()
classifier.fit(x_train,y_train)

Out[18]: SVC()

In [19]: # y prediction
y_pred=classifier.predict(x_test)

In [20]: # finding accuracy score
from sklearn.metrics import accuracy_score
accuracy_score(y_test, y_pred)

Out[20]: 0.8638941398865785

In [21]: param_grid={'C':[0.001,0.01,0.1,1,10,100], 'gamma':[0.001,0.01,0.1,1,10,100]}

In [22]: # performing GridSearchCV
from sklearn.model_selection import GridSearchCV
grid_search_cv=GridSearchCV(SVC(),param_grid,scoring='accuracy',n_jobs=-1)

In [23]: grid_search_cv.fit(x_train,y_train)

Out[23]: GridSearchCV(estimator=SVC(), n_jobs=-1,
param_grid={'C': [0.001, 0.01, 0.1, 1, 10, 100],
'gamma': [0.001, 0.01, 0.1, 1, 10, 100]},
scoring='accuracy')

In [24]: grid_search_cv.best_score_

Out[24]: 0.9634861414367288

In [25]: from sklearn.model_selection import RandomizedSearchCV

In [26]: rand_search_cv=RandomizedSearchCV(SVC(),param_distributions=param_grid,scoring='accuracy',n_jobs=-1)

In [27]: rand_search_cv.fit(x_train,y_train)

Out[27]: RandomizedSearchCV(estimator=SVC(), n_jobs=-1,
param_distributions={'C': [0.001, 0.01, 0.1, 1, 10, 100],
'gamma': [0.001, 0.01, 0.1, 1, 10, 100]},
scoring='accuracy')

In [31]: rand_search_cv.best_score_

Out[31]: 0.8573174148478377

In [29]: rand_search_cv.best_params_

Out[29]: {'gamma': 0.01, 'C': 1}

In [30]: grid_search_cv.best_params_

Out[30]: {'C': 10, 'gamma': 0.001}

In [32]: #Confusion matrix
from sklearn.metrics import confusion_matrix
confusion_svm=confusion_matrix(y_test,y_pred)
sns.heatmap(confusion_svm,annot=True)

Out[32]:
```

```
In [38]: #classification report
from sklearn.metrics import classification_report
print(classification_report(y_test,y_pred))
```

	precision	recall	f1-score	support
1	0.90	0.96	0.93	426
2	0.55	0.35	0.43	66
3	0.81	0.68	0.74	37
accuracy				529
macro avg	0.75	0.66	0.66	529
weighted avg	0.85	0.86	0.85	529

In []: