IMPORT LIBRARIES

In [1]:

```
import pandas as pd
import seaborn as sb
import matplotlib.pyplot as plt
import numpy as np
import math
import plotly.animation
```

```
#pygal
!pip install pygal

! pip install plotly

! pip install plotly
```

In [2]:

```
fetal = pd.read_csv("fetal_health.csv")
fetal.transpose()
```

Out[2]:

	0	1	2	3	
baseline value	120.0	132.000	133.000	134.000	132.00
accelerations	0.0	0.006	0.003	0.003	0.00
fetal_movement	0.0	0.000	0.000	0.000	0.00
uterine_contractions	0.0	0.006	0.008	0.008	0.00
light_decelerations	0.0	0.003	0.003	0.003	0.00
severe_decelerations	0.0	0.000	0.000	0.000	0.00
prolongued_decelerations	0.0	0.000	0.000	0.000	0.00
abnormal_short_term_variability	73.0	17.000	16.000	16.000	16.00
mean_value_of_short_term_variability	0.5	2.100	2.100	2.400	2.40
percentage_of_time_with_abnormal_long_term_variability	43.0	0.000	0.000	0.000	0.00
mean_value_of_long_term_variability	2.4	10.400	13.400	23.000	19.90
histogram_width	64.0	130.000	130.000	117.000	117.00
histogram_min	62.0	68.000	68.000	53.000	53.00
histogram_max	126.0	198.000	198.000	170.000	170.00
histogram_number_of_peaks	2.0	6.000	5.000	11.000	9.00
histogram_number_of_zeroes	0.0	1.000	1.000	0.000	0.00
histogram_mode	120.0	141.000	141.000	137.000	137.00
histogram_mean	137.0	136.000	135.000	134.000	136.00
histogram_median	121.0	140.000	138.000	137.000	138.00
histogram_variance	73.0	12.000	13.000	13.000	11.00
histogram_tendency	1.0	0.000	0.000	1.000	1.00
fetal_health	2.0	1.000	1.000	1.000	1.00

22 rows × 2126 columns

In [3]:

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

Out[3]:

```
1.0    1655
2.0     295
3.0     176
Name: fetal_health, dtype: int64
```

In [4]:

```
!pip install pandas_profiling
from pandas_profiling import ProfileReport
```

```
nda3\lib\site-packages (from matplotlib>=3.2.0->pandas_profiling) (2020.6.
20)
Requirement already satisfied: pytz>=2017.2 in c:\users\prath\anaconda3\li
b\site-packages (from pandas!=1.0.0,!=1.0.1,!=1.0.2,!=1.1.0,>=0.25.3->pand
as_profiling) (2020.1)
Requirement already satisfied: MarkupSafe>=0.23 in c:\users\prath\anaconda
3\lib\site-packages (from jinja2>=2.11.1->pandas_profiling) (1.1.1)
Requirement already satisfied: chardet<4,>=3.0.2 in c:\users\prath\anacond
a3\lib\site-packages (from requests>=2.24.0->pandas_profiling) (3.0.4)
Requirement already satisfied: idna<3,>=2.5 in c:\users\prath\anaconda3\li
b\site-packages (from requests>=2.24.0->pandas_profiling) (2.10)
Requirement already satisfied: urllib3!=1.25.0,!=1.25.1,<1.26,>=1.21.1 in
c:\users\prath\anaconda3\lib\site-packages (from requests>=2.24.0->pandas_
profiling) (1.25.11)
Requirement already satisfied: pyyaml in c:\users\prath\anaconda3\lib\site
-packages (from confuse>=1.0.0->pandas_profiling) (5.3.1)
Requirement already satisfied: networkx>=2.4 in c:\users\prath\anaconda3\l
ib\site-packages (from visions[type_image_path]==0.6.0->pandas_profiling)
(2.5)
Requirement already satisfied: imagehash; extra == "type_image_path" in
```

In [5]:

```
pr = ProfileReport(fetal)
```

In [6]:

```
pr.to_notebook_iframe()
```

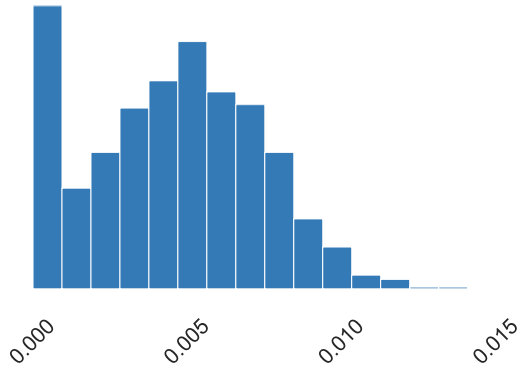
Summarize dataset: 100% 35/35 [00:51<00:00, 1.48s/it, Completed]

Generate report structure: 100% 1/1 [02:52<00:00, 172.12s/it]

Render HTML: 100% 1/1 [00:08<00:00, 8.55s/it]

Infinite (%)	0.0%

Mean	0.004366415804
Minimum	0
Maximum	0.015
Zeros	332
Zeros (%)	15.6%
Memory size	16.7 KiB



Quantile statistics

Minimum	0
5-th percentile	0
Q1	0.002
median	0.004
Q3	0.007
95-th percentile	0.009
Maximum	0.015
Range	0.015

In [7]:

```
fetal.columns
```

Out[7]:

```
Index(['baseline value', 'accelerations', 'fetal_movement',
       'uterine_contractions', 'light_decelerations', 'severe_deceleration
s',
       'prolongued_decelerations', 'abnormal_short_term_variability',
       'mean_value_of_short_term_variability',
       'percentage_of_time_with_abnormal_long_term_variability',
       'mean_value_of_long_term_variability', 'histogram_width',
       'histogram_min', 'histogram_max', 'histogram_number_of_peaks',
       'histogram_number_of_zeroes', 'histogram_mode', 'histogram_mean',
       'histogram_median', 'histogram_variance', 'histogram_tendency',
       'fetal_health'],
      dtype='object')
```

In [8]:

```
fetal.describe().transpose()
```

Out[8]:

	count	mean	std	min	
baseline value	2126.0	133.303857	9.840844	106.0	1
accelerations	2126.0	0.003178	0.003866	0.0	
fetal_movement	2126.0	0.009481	0.046666	0.0	
uterine_contractions	2126.0	0.004366	0.002946	0.0	
light_decelerations	2126.0	0.001889	0.002960	0.0	
severe_decelerations	2126.0	0.000003	0.000057	0.0	
prolongued_decelerations	2126.0	0.000159	0.000590	0.0	
abnormal_short_term_variability	2126.0	46.990122	17.192814	12.0	
mean_value_of_short_term_variability	2126.0	1.332785	0.883241	0.2	
percentage_of_time_with_abnormal_long_term_variability	2126.0	9.846660	18.396880	0.0	
mean_value_of_long_term_variability	2126.0	8.187629	5.628247	0.0	
histogram_width	2126.0	70.445908	38.955693	3.0	
histogram_min	2126.0	93.579492	29.560212	50.0	
histogram_max	2126.0	164.025400	17.944183	122.0	1
histogram_number_of_peaks	2126.0	4.068203	2.949386	0.0	
histogram_number_of_zeroes	2126.0	0.323612	0.706059	0.0	
histogram_mode	2126.0	137.452023	16.381289	60.0	1
histogram_mean	2126.0	134.610536	15.593596	73.0	1
histogram_median	2126.0	138.090310	14.466589	77.0	1
histogram_variance	2126.0	18.808090	28.977636	0.0	
histogram_tendency	2126.0	0.320320	0.610829	-1.0	
fetal_health	2126.0	1.304327	0.614377	1.0	

In [9]:

```
fetal.select_dtypes(exclude='object').columns
```

Out[9]:

```
Index(['baseline value', 'accelerations', 'fetal_movement',
       'uterine_contractions', 'light_decelerations', 'severe_deceleration
s',
       'prolongued_decelerations', 'abnormal_short_term_variability',
       'mean_value_of_short_term_variability',
       'percentage_of_time_with_abnormal_long_term_variability',
       'mean_value_of_long_term_variability', 'histogram_width',
       'histogram_min', 'histogram_max', 'histogram_number_of_peaks',
       'histogram_number_of_zeroes', 'histogram_mode', 'histogram_mean',
       'histogram_median', 'histogram_variance', 'histogram_tendency',
       'fetal_health'],
      dtype='object')
```

In [10]:

```
fetal.shape
```

Out[10]:

```
(2126, 22)
```

In [11]:

```
fetal.isnull().sum()
```

Out[11]:

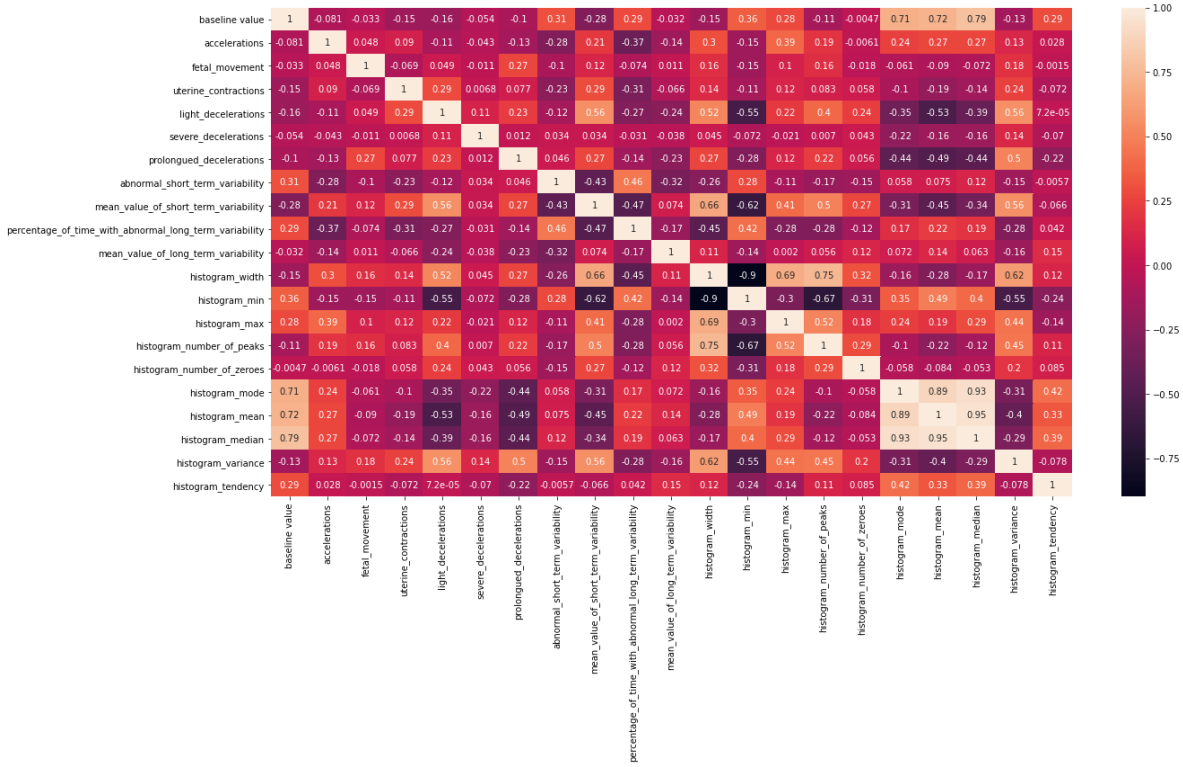
```
baseline value                                            0
accelerations                                             0
fetal_movement                                            0
uterine_contractions                                      0
light_decelerations                                       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 [32]:

```
correlation = fetal.corr()
correlation
plt.figure(figsize=(20,10))
sb.heatmap(correlation,annot=True)
```

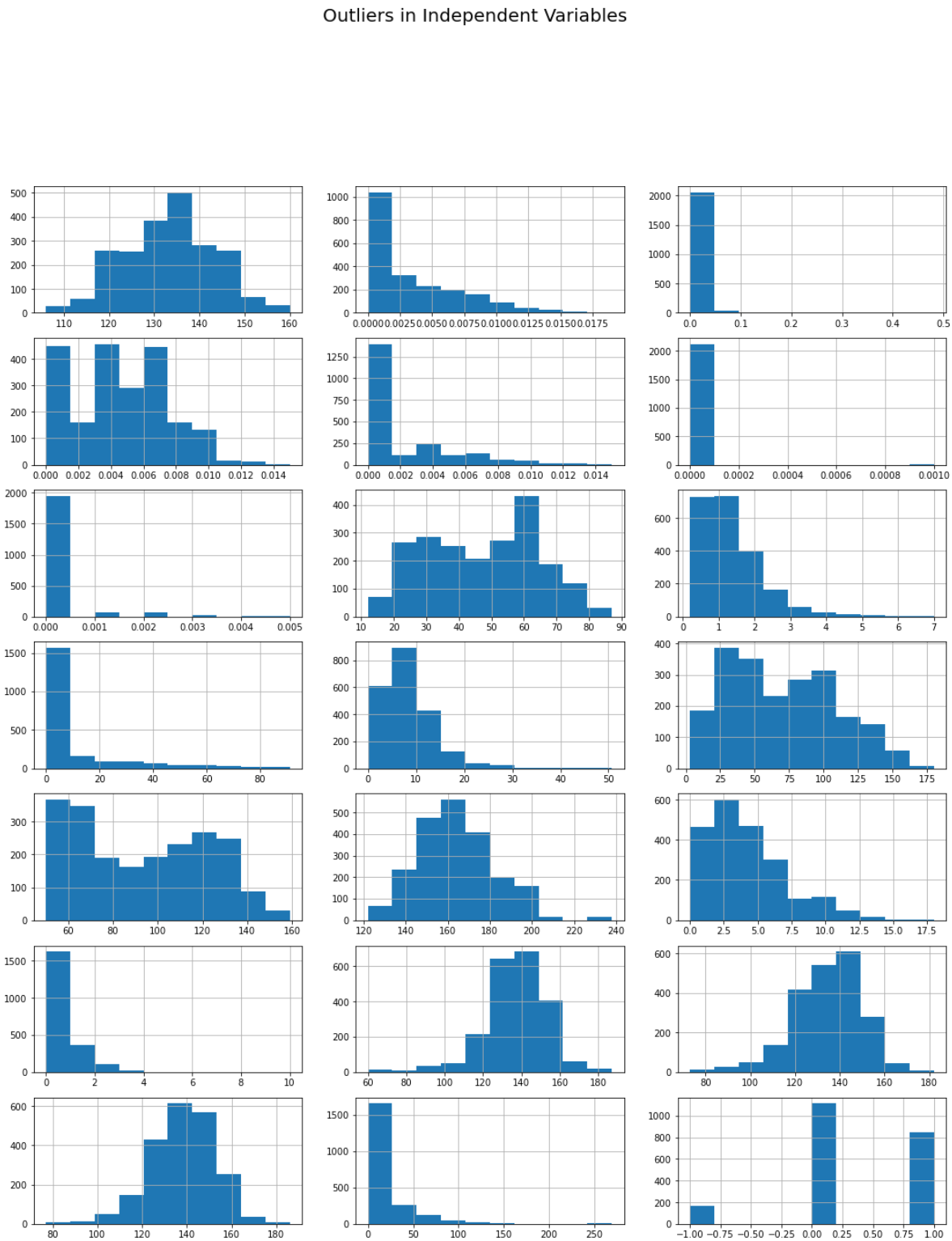
Out[32]:

<AxesSubplot:>



In [33]:

```
plt.figure(figsize=(18,21))
for i,col in enumerate(fetal.columns[:-1]):
    plt.subplot(7,3,i+1)
    fetal[col].hist()
plt.suptitle('Outliers in Independent Variables', size=20, y=1.01)
plt.show()
```



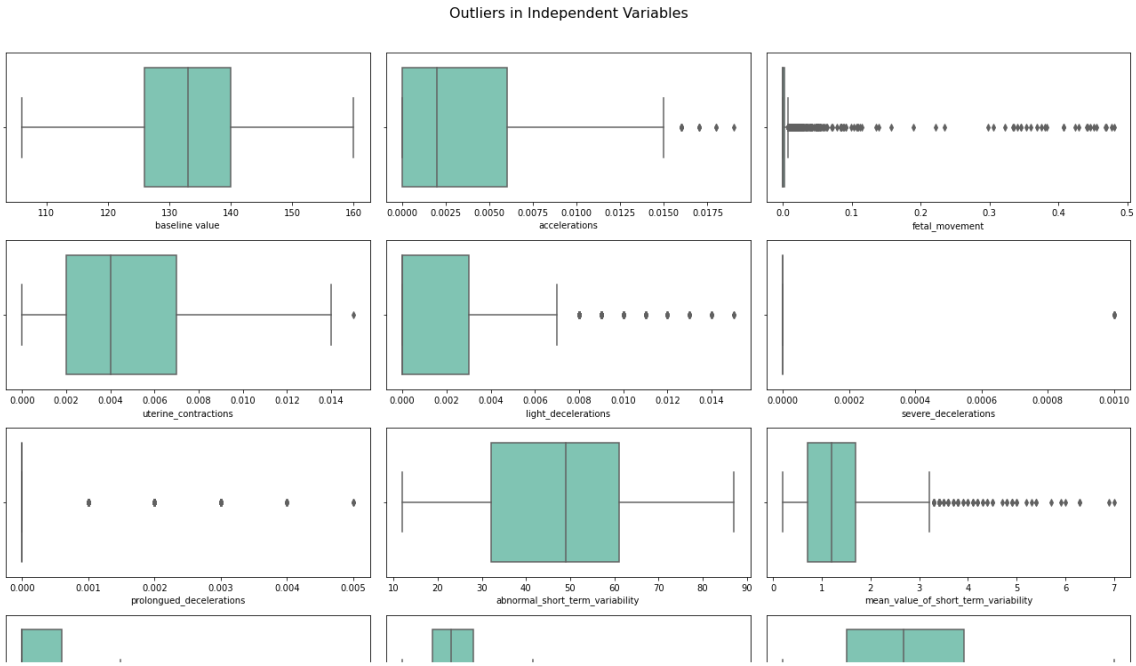
In [34]:

```
plt.figure(figsize=(18,21))
for i,col in enumerate(fetal.columns[:-1]):
    plt.subplot(7,3,i+1)
    sb.kdeplot(x=fetal[col],hue=fetal['fetal_health'])
plt.suptitle('Distribution of Independent Variables w.r.t. Dependent Variable', size=16, y=
plt.tight_layout()
plt.show()
```

outlier check

In [35]:

```
plt.figure(figsize=(18,21))
for i,col in enumerate(fetal.columns[:-1]):
    plt.subplot(7,3,i+1)
    sb.boxplot(x=fetal[col], color='#75cfb8')
plt.suptitle('Outliers in Independent Variables', size=16, y=1.01)
plt.tight_layout()
plt.show()
```



In [36]:

```
df = fetal.copy()
```

In [37]:

```
def removeOutlier(att, df):

    lowerbound = att.mean() - 3 * att.std()
    upperbound = att.mean() + 3 * att.std()

    print('lowerbound: ',lowerbound,' -------- upperbound: ', upperbound )

    df1 = df[(att > lowerbound) & (att < upperbound)]

    print((df.shape[0] - df1.shape[0]), ' number of outliers from ', df.shape[0] )
    print(' ***************************************************')

    df = df1.copy()

    return df
```

In [38]:

```
df = removeOutlier(df.histogram_variance, df)
df = removeOutlier(df.histogram_median, df)
df = removeOutlier(df.histogram_mean, df)
df = removeOutlier(df.histogram_mode, df)
df = removeOutlier(df.percentage_of_time_with_abnormal_long_term_variability, df)
df = removeOutlier(df.mean_value_of_short_term_variability, df)
```

```
lowerbound:  -68.12481771467543  -------- upperbound:  105.74099833555971
44  number of outliers from  2126
 ****************************************************
lowerbound:  96.21293175778905  -------- upperbound:  180.71886459187473
17  number of outliers from  2082
 ****************************************************
lowerbound:  92.08366378331934  -------- upperbound:  178.71246212467096
17  number of outliers from  2065
 ****************************************************
lowerbound:  97.08081110895039  -------- upperbound:  180.2912592035496
17  number of outliers from  2048
 ****************************************************
lowerbound:  -45.780257839676395  -------- upperbound:  66.39473346744597
57  number of outliers from  2031
 ****************************************************
lowerbound:  -1.1711686306789288  -------- upperbound:  3.781908245673867
30  number of outliers from  1974
 ****************************************************
```

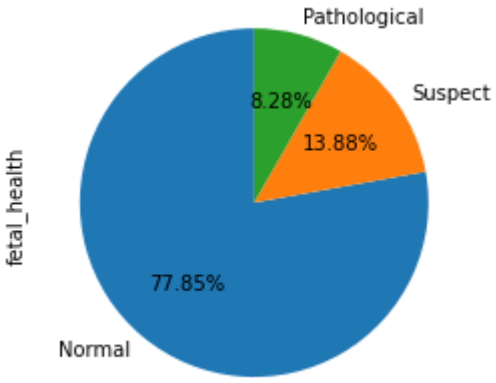
TARGET VARIABLE BALANCING

In [39]:

```
fetal['fetal_health'].value_counts().plot(kind='pie', autopct='%.2f%%',
                                          startangle=90,
                                         labels = ['Normal','Suspect','Pathological'])
```

Out[39]:

```
<AxesSubplot:ylabel='fetal_health'>
```



In [40]:

```
pip install imblearn
```

```
Requirement already satisfied: imblearn in c:\users\prath\anaconda3\lib\site
-packages (0.0)
Requirement already satisfied: imbalanced-learn in c:\users\prath\anaconda3
\lib\site-packages (from imblearn) (0.8.0)
Requirement already satisfied: numpy>=1.13.3 in c:\users\prath\anaconda3\lib
\site-packages (from imbalanced-learn->imblearn) (1.19.2)
Requirement already satisfied: joblib>=0.11 in c:\users\prath\anaconda3\lib
\site-packages (from imbalanced-learn->imblearn) (0.17.0)
Requirement already satisfied: scipy>=0.19.1 in c:\users\prath\anaconda3\lib
\site-packages (from imbalanced-learn->imblearn) (1.5.2)
Requirement already satisfied: scikit-learn>=0.24 in c:\users\prath\anaconda
3\lib\site-packages (from imbalanced-learn->imblearn) (0.24.1)
Requirement already satisfied: threadpoolctl>=2.0.0 in c:\users\prath\anacon
da3\lib\site-packages (from scikit-learn>=0.24->imbalanced-learn->imblearn)
(2.1.0)
Note: you may need to restart the kernel to use updated packages.
```

In [41]:

```
!pip install lightgbm
!pip install wheel
!pip install catboost
!pip install ipywidgets
```

```
Requirement already satisfied: lightgbm in c:\users\prath\anaconda3\lib\si
te-packages (3.2.1)
Requirement already satisfied: numpy in c:\users\prath\anaconda3\lib\site-
packages (from lightgbm) (1.19.2)
Requirement already satisfied: scikit-learn!=0.22.0 in c:\users\prath\anac
onda3\lib\site-packages (from lightgbm) (0.24.1)
Requirement already satisfied: scipy in c:\users\prath\anaconda3\lib\site-
packages (from lightgbm) (1.5.2)
Requirement already satisfied: wheel in c:\users\prath\anaconda3\lib\site-
packages (from lightgbm) (0.35.1)
Requirement already satisfied: threadpoolctl>=2.0.0 in c:\users\prath\anac
onda3\lib\site-packages (from scikit-learn!=0.22.0->lightgbm) (2.1.0)
Requirement already satisfied: joblib>=0.11 in c:\users\prath\anaconda3\li
b\site-packages (from scikit-learn!=0.22.0->lightgbm) (0.17.0)
Requirement already satisfied: wheel in c:\users\prath\anaconda3\lib\site-
packages (0.35.1)
Requirement already satisfied: catboost in c:\users\prath\anaconda3\lib\si
te-packages (0.25.1)
Requirement already satisfied: plotly in c:\users\prath\anaconda3\lib\site
```

In [42]:

```
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler

from sklearn.linear_model import LogisticRegression
from sklearn.neighbors import KNeighborsClassifier
from sklearn.tree import DecisionTreeClassifier
from sklearn.svm import LinearSVC, SVC
from sklearn.neural_network import MLPClassifier
from sklearn.ensemble import RandomForestClassifier, GradientBoostingClassifier
from xgboost import XGBClassifier
from lightgbm import LGBMClassifier
from catboost import CatBoostClassifier

import warnings
warnings.filterwarnings(action='ignore')
```

In [43]:

```
def preprocess_inputs(df):


    # Rename target values
    df['fetal_health'] = df['fetal_health'].replace({
        1.0: "NORMAL",
        2.0: "SUSPECT",
        3.0: "PATHOLOGICAL"
    })

    # Split df into X and y
    y = df['fetal_health']
    X = df.drop('fetal_health', axis=1)

    # Train-test split
    X_train, X_test, y_train, y_test = train_test_split(X, y, train_size=0.7, shuffle=True,

    # Scale X
    scaler = StandardScaler()
    scaler.fit(X_train)
    X_train = pd.DataFrame(scaler.transform(X_train), index=X_train.index, columns=X_train.
    X_test = pd.DataFrame(scaler.transform(X_test), index=X_test.index, columns=X_test.colu

    return X_train, X_test, y_train, y_test
```

In [44]:

```
k = np.sqrt(fetal.shape[0]).round()
k
```

Out[44]:

```
46.0
```

In [45]:

```
X_train, X_test, y_train, y_test = preprocess_inputs(fetal)
```

In [46]:

```
models = {
    "                    Logistic Regression": LogisticRegression(penalty='l2',solver='sag',
    "                    K-Nearest Neighbors": KNeighborsClassifier(  n_neighbors=3,weights=
    "                          Decision Tree": DecisionTreeClassifier(criterion='gini'),
    "                 Support Vector Machine ": SVC(kernel='poly'),
    "                  Support Vector Machine": SVC(kernel='rbf'),
    "                          Neural Network": MLPClassifier(),
    "                           Random Forest": RandomForestClassifier(),
    "                       Gradient Boosting": GradientBoostingClassifier(),
    "                                 XGBoost": XGBClassifier(eval_metric='mlogloss'),

}

for name, model in models.items():
    model.fit(X_train, y_train)
    print(name + " trained.")
```

```
                    Logistic Regression trained.
                    K-Nearest Neighbors trained.
                          Decision Tree trained.
                 Support Vector Machine  trained.
                  Support Vector Machine trained.
                          Neural Network trained.
                           Random Forest trained.
                       Gradient Boosting trained.
                                 XGBoost trained.
```

In [47]:

```
for name, model in models.items():
    pred = model.predict(X_test)
    print(name + " predicted.")
```

```
                    Logistic Regression predicted.
                    K-Nearest Neighbors predicted.
                          Decision Tree predicted.
                 Support Vector Machine  predicted.
                  Support Vector Machine predicted.
                          Neural Network predicted.
                           Random Forest predicted.
                       Gradient Boosting predicted.
                                 XGBoost predicted.
```

In [48]:

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

In [49]:

```
for name, model in models.items():
    print(name + ": {:.2f}%".format(model.score(X_test, y_test) * 100))
```

```
                   Logistic Regression: 90.13%
                   K-Nearest Neighbors: 90.13%
                         Decision Tree: 91.38%
                Support Vector Machine : 89.03%
                 Support Vector Machine: 91.07%
                        Neural Network: 91.54%
                         Random Forest: 93.26%
                     Gradient Boosting: 94.98%
                               XGBoost: 94.98%
```

In [52]:

```
print(classification_report(y_test,pred))
```

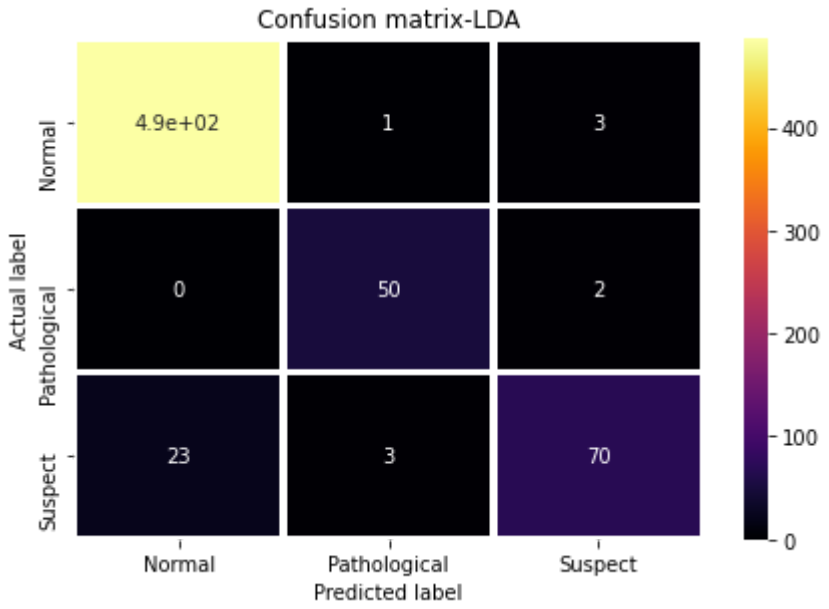
```
              precision    recall  f1-score   support

      NORMAL       0.95      0.99      0.97       490
PATHOLOGICAL       0.93      0.96      0.94        52
     SUSPECT       0.93      0.73      0.82        96

    accuracy                           0.95       638
   macro avg       0.94      0.89      0.91       638
weighted avg       0.95      0.95      0.95       638
```

In [53]:

```
#performance metrics
#confusion matrix-lda
cnf_matrix = confusion_matrix(y_test,pred)
cnf_matrix
sb.heatmap(pd.DataFrame(cnf_matrix), annot=True,linecolor='white',linewidths='2.6', cmap='i
           ,xticklabels=['Normal','Pathological','Suspect'], yticklabels=['Normal','Patholo
plt.tight_layout()
plt.title('Confusion matrix-LDA')
plt.ylabel('Actual label')
plt.xlabel('Predicted label')
```

Out[53]:

```
Text(0.5, 15.0, 'Predicted label')
```



In [ ]: