In [1]:
```
import warnings
warnings.simplefilter(action ="ignore")
warnings.filterwarnings('ignore')
from collections import Counter
# Import the necessary packages
import numpy as np
import pandas as pd
# Data visualization
import matplotlib.pyplot as plt
import seaborn as sns
# Algorithms
from sklearn.model_selection import cross_val_score
from sklearn.preprocessing import StandardScaler
from sklearn.model_selection import train_test_split
from sklearn import linear_model
from sklearn.linear_model import LogisticRegression
from sklearn.ensemble import RandomForestClassifier
from sklearn.ensemble import GradientBoostingClassifier
from sklearn.model_selection import GridSearchCV
from sklearn.metrics import classification_report, confusion_matrix
from sklearn.metrics import mean_squared_error
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.preprocessing import LabelEncoder, OneHotEncoder, PolynomialFeature
from sklearn.pipeline import Pipeline, make_pipeline
from sklearn.model_selection import GridSearchCV
from sklearn.metrics import accuracy_score, recall_score, f1_score, confusion_ma
from sklearn.linear_model import LogisticRegression
from sklearn.svm import SVC
from sklearn.tree import DecisionTreeClassifier
from sklearn.metrics import plot_confusion_matrix, auc
import pickle
```

In [2]:
```
# Load Dataset
data = pd.read_csv('data/fetal_health.csv')
# previewing the DataFrame
data.head()
```

Out[2]:

	baseline value	accelerations	fetal_movement	uterine_contractions	light_decelerations	severe_dece
0	120.0	0.000	0.0	0.000	0.000	
1	132.0	0.006	0.0	0.006	0.003	
2	133.0	0.003	0.0	0.008	0.003	
3	134.0	0.003	0.0	0.008	0.003	
4	132.0	0.007	0.0	0.008	0.000	

5 rows × 22 columns

In [3]:
```
data = data.rename(columns = {'baseline value':'baseline_value',
                              'abnormal_short_term_variability':'percentage_of_t
                              'prolongued_decelerations': 'prolonged_deceleratio
```

In [4]:
```
# engineering new categorical target column for Normal and At Risk fetal health
```

```
data.loc[data['fetal_health']==1.000, 'fh_outcome'] = 'Normal'
data.loc[data['fetal_health']==2.000, 'fh_outcome'] = 'Risk'
data.loc[data['fetal_health']==3.000, 'fh_outcome'] = 'Risk'
data = data.drop(columns='fetal_health', axis=1)
data.head()
```

Out[4]:

	baseline_value	accelerations	fetal_movement	uterine_contractions	light_decelerations	sever
0	120.0	0.000	0.0	0.000	0.000	
1	132.0	0.006	0.0	0.006	0.003	
2	133.0	0.003	0.0	0.008	0.003	
3	134.0	0.003	0.0	0.008	0.003	
4	132.0	0.007	0.0	0.008	0.000	

5 rows × 22 columns

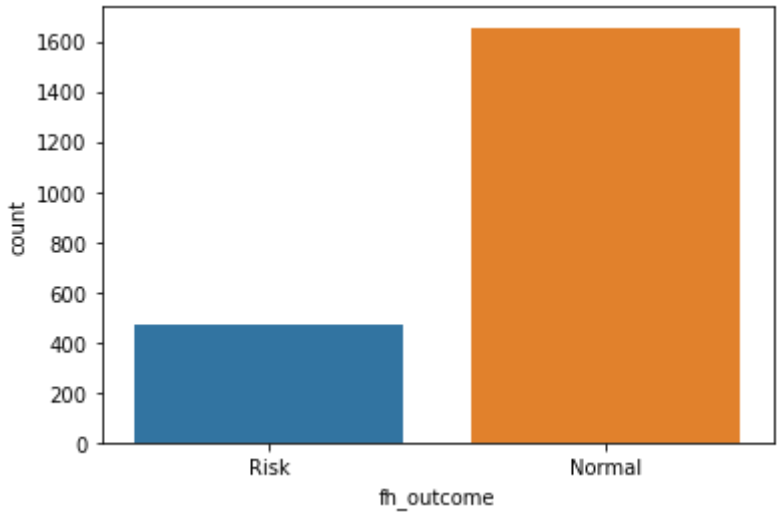
In [5]:
```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2126 entries, 0 to 2125
Data columns (total 22 columns):
 #   Column                                                  Non-Null Count  Dt
ype
---  ------                                                  --------------  --
---
 0   baseline_value                                          2126 non-null   fl
oat64
 1   accelerations                                           2126 non-null   fl
oat64
 2   fetal_movement                                          2126 non-null   fl
oat64
 3   uterine_contractions                                    2126 non-null   fl
oat64
 4   light_decelerations                                     2126 non-null   fl
oat64
 5   severe_decelerations                                    2126 non-null   fl
oat64
 6   prolonged_decelerations                                 2126 non-null   fl
oat64
 7   percentage_of_time_with_abnormal_short_term_variability 2126 non-null   fl
oat64
 8   mean_value_of_short_term_variability                    2126 non-null   fl
oat64
 9   percentage_of_time_with_abnormal_long_term_variability  2126 non-null   fl
oat64
 10  mean_value_of_long_term_variability                     2126 non-null   fl
oat64
 11  histogram_width                                         2126 non-null   fl
oat64
 12  histogram_min                                           2126 non-null   fl
oat64
 13  histogram_max                                           2126 non-null   fl
oat64
 14  histogram_number_of_peaks                               2126 non-null   fl
oat64
 15  histogram_number_of_zeroes                              2126 non-null   fl
oat64
 16  histogram_mode                                          2126 non-null   fl
oat64
```

```
 17  histogram_mean                                    2126 non-null   fl
oat64
 18  histogram_median                                  2126 non-null   fl
oat64
 19  histogram_variance                                2126 non-null   fl
oat64
 20  histogram_tendency                                2126 non-null   fl
oat64
 21  fh_outcome                                        2126 non-null   ob
ject
dtypes: float64(21), object(1)
memory usage: 365.5+ KB
```

In [6]:
```
sns.countplot(x = 'fh_outcome', data = data)
```

Out[6]:
```
<AxesSubplot:xlabel='fh_outcome', ylabel='count'>
```



In [7]:
```
# setting target and features
y = data['fh_outcome']
X = data.drop(columns='fh_outcome')
# splitting the data into train and test sets
# using stratify parameter to make sure class ratios
# are distributed evenly across train and test sets
X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.3, stratify=y, random_state=42)
```

In [8]:
```
y_train
```

Out[8]:
```
468       Risk
123     Normal
1099    Normal
1804    Normal
381     Normal
         ...
1370      Risk
1926    Normal
410       Risk
1449    Normal
229     Normal
Name: fh_outcome, Length: 1488, dtype: object
```

In [9]:
```
Lb = LabelEncoder()
Lb.fit(y_train)
y_train = Lb.transform(y_train)
y_test = Lb.transform(y_test)
```

In [10]:
```
# Fit SMOTE to training data
from imblearn.over_sampling import SMOTE
X_train_resampled, y_train_resampled = SMOTE().fit_resample(X_train, y_train)
```

Build Pipeline

In [12]:
```
pipe_SVM = Pipeline([
                     ('Scl', StandardScaler()),
                     ('Svm', SVC())
                    ])
```

In [13]:
```
param_grid_SVM = [{"Svm__C":[0.5,1,2,3,5],
                   "Svm__tol":[0.001,0.0001,0.00001],
                   "Svm__kernel":['linear','rbf','poly'],
#                    "Svm__gamma":['scale','auto',0.1,0.01,0.001,0.0001]
                  }]
```

In [14]:
```
GridSearchCV_SVM = GridSearchCV(estimator=pipe_SVM,
                                param_grid=param_grid_SVM,
                                cv=10,
                                verbose=1,
                                n_jobs=-1,
                                scoring = 'recall',
                                return_train_score=True
                                )
```

In [15]:
```
grid = GridSearchCV_SVM.fit(X_train_resampled,y_train_resampled)
```

```
Fitting 10 folds for each of 45 candidates, totalling 450 fits
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 8 concurrent workers.
[Parallel(n_jobs=-1)]: Done  34 tasks      | elapsed:    1.5s
[Parallel(n_jobs=-1)]: Done 320 tasks      | elapsed:    5.3s
[Parallel(n_jobs=-1)]: Done 450 out of 450 | elapsed:    7.2s finished
```

In [16]:
```
grid.score(X_test, y_test)
```

Out[16]:
```
0.9432624113475178
```

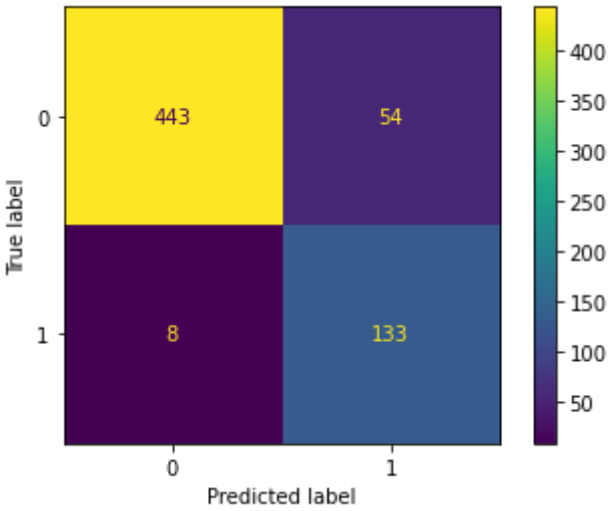
In [17]:
```
y_pred_train = grid.predict(X_train_resampled)
y_pred = grid.predict(X_test)
```

In [18]:
```
print(classification_report(y_train_resampled, y_pred_train))
print(classification_report(y_test, y_pred))
```

```
              precision    recall  f1-score   support

           0       0.99      0.90      0.95      1158
           1       0.91      0.99      0.95      1158

    accuracy                           0.95      2316
   macro avg       0.95      0.95      0.95      2316
weighted avg       0.95      0.95      0.95      2316

              precision    recall  f1-score   support

           0       0.98      0.89      0.93       497
           1       0.71      0.94      0.81       141
```

```
              precision    recall  f1-score   support

    accuracy                         0.90       638
   macro avg       0.85      0.92      0.87       638
weighted avg       0.92      0.90      0.91       638
```

In [19]:
```
# plotting confusion matrix
plot_confusion_matrix(grid, X_test, y_test)
plt.show()
```



Onehotencoder()/Labelencoder()/imputer()