

```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.tree import DecisionTreeClassifier
from sklearn import metrics
from sklearn.model_selection import cross_val_score
#read data into dataframe
data = pd.read_csv ('train.csv')
```

```
#Examine the first few rows
print(data.head())
```

	baseline value	accelerations	fetal_movement	uterine_contractions	\
0	142.0	0.000	0.000	0.007	
1	122.0	0.000	0.000	0.006	
2	129.0	0.005	0.003	0.001	
3	136.0	0.006	0.000	0.008	
4	144.0	0.000	0.000	0.006	

	light_decelerations	severe_decelerations	prolongued_decelerations	\
0	0.000	0.0	0.0	
1	0.002	0.0	0.0	
2	0.000	0.0	0.0	
3	0.000	0.0	0.0	
4	0.000	0.0	0.0	

	abnormal_short_term_variability	mean_value_of_short_term_variability	\
0	58.0	0.4	
1	27.0	1.4	
2	34.0	1.7	
3	45.0	0.8	
4	32.0	1.0	

	percentage_of_time_with_abnormal_long_term_variability	...	histogram_min	\
0	9.0	...	136.0	
1	4.0	...	91.0	
2	0.0	...	78.0	
3	2.0	...	129.0	
4	0.0	...	122.0	

	histogram_max	histogram_number_of_peaks	histogram_number_of_zeroes	\
0	156.0	0.0	0.0	
1	144.0	4.0	0.0	
2	196.0	10.0	0.0	
3	158.0	2.0	0.0	
4	160.0	1.0	0.0	

	histogram_mode	histogram_mean	histogram_median	histogram_variance	\
0	148.0	147.0	149.0	1.0	
1	126.0	120.0	122.0	6.0	
2	137.0	136.0	137.0	6.0	

3	144.0	143.0	145.0	1.0
4	150.0	147.0	149.0	2.0

	histogram_tendency	fetal_health
0	0.0	1.0
1	0.0	1.0
2	0.0	1.0
3	0.0	1.0
4	1.0	1.0

[5 rows x 22 columns]

```
#split data into train and test sets
```

```
Y = data.fetal_health
```

```
X = data.drop('fetal_health',axis=1)
```

```
train_X, test_X, train_Y, test_Y = train_test_split(X,Y, test_size = 0.2)
```

```
print('train_X', train_X.shape)
```

```
print('train_Y', train_Y.shape)
```

```
print('test_X', test_X.shape)
```

```
print('test_Y', test_Y.shape)
```

```
train_X (1360, 21)
```

```
train_Y (1360,)
```

```
test_X (340, 21)
```

```
test_Y (340,)
```

```
#decide on the model
```

```
model = DecisionTreeClassifier()
```

```
# fit the model to the training set
```

```
model.fit(train_X, train_Y)
```

```
DecisionTreeClassifier()
```

```
from sklearn.metrics import accuracy_score
```

```
#using the trained model predict the output of the test_X input values
```

```
y_pred = model.predict(test_X)
```

```
# Model Accuracy, how often is the classifier correct?
```

```
print("Accuracy: {:.0f}%".format(accuracy_score(test_Y, y_pred)*100))
```

Accuracy: 93%

```
Validation_test = pd.read_csv('test.csv')
```

```
print('Validation_test', Validation_test.shape)
```