Maternal Health Risk Data Set

FINAL PROJECT ON MACHINE LEARNING MODEL

PROGRAMMING IN PYTHON

Project Overview

The main purpose of doing this project was to get a proper idea about the machine learning and the model in machine learning. This project mainly focuses on the data set of Maternal health risk.Firstly,diffrent types of python libraries are used for a good output. while working with the datasets a target was selected to focus and predict accuracy.The models used for the accuracy of the dataset can become a important asset to understand the maternal health risk in the rural areas.Lastly, it can be said that the project's main target was to increase knowledge about diffrent algorithms and outcomes.

In [1]:
```
#importing necessary python libraries
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn import metrics
%matplotlib inline
```

Dataset reference

https://archive.ics.uci.edu/ml/datasets/Maternal+Health+Risk+Data+Set

Total Dataset Information

The data was colleted from Maternal Health Risk Data Set . Data has been collected from different hospitals, community clinics, maternal health cares from the rural areas of Bangladesh through the IoT based risk monitoring system. The Age, Systolic Blood Pressure as SystolicBP, Diastolic BP as DiastolicBP, Blood Sugar as BS, Body Temperature as BodyTemp, HeartRate and RiskLevel. All these are the responsible and significant risk factors for maternal mortality, that is one of the main concern of SDG of UN.

Attribute information:

1. AGE: Age-Any ages in years when a women during pregnant. 2.SBP: SystolicBP-Upper value of Blood Pressure in mmHg,another significant attribute during pregnancy. 3.DBP: DiastolicBP-Lower value of Blood Pressure in mmHg,another significant attribute during pregnancy. 4.BS: Blood glucose levels is in terms of a molar concentration, mmol/L. 5.BTM: BodyTemprature troughout the pregnancy. 6.HRT: HeartRate-A normal resting heart rate in beats per minute. 7.RLV: Risk Level: Predicted Risk Intensity Level during pregnancy considering the previous attribute.

   Relevant Papers:

1. Ahmed M., Kashem M.A., Rahman M., Khatun S. (2020) Review and Analysis of Risk Factor of Maternal Health in Remote Area Using the Internet of Things (IoT). In: Kasruddin Nasir A. et al. (eds) InECCE2019. Lecture Notes in Electrical Engineering, vol 632. Springer, Singapore. [Web Link]
2. IoT based Risk Level Prediction Model for Maternal Health Care in the Context of Bangladesh, STI-2020, [under publication in IEEE]

In [2]:
```
#importing the datasets from csv file in a dataframe and showing

dF=pd.read_csv('r_dataset.csv')
dF
```

Out[2]:

	AGE	SBP	DBP	BS	BTM	HRT	RLV
0	25	130	80	15.0	98.0	86	high risk
1	35	140	90	13.0	98.0	70	high risk
2	29	90	70	8.0	100.0	80	high risk
3	30	140	85	7.0	98.0	70	high risk
4	35	120	60	6.1	98.0	76	low risk
...	...	...	...	...	...	...	...
1009	22	120	60	15.0	98.0	80	high risk
1010	55	120	90	18.0	98.0	60	high risk
1011	35	85	60	19.0	98.0	86	high risk
1012	43	120	90	18.0	98.0	70	high risk
1013	32	120	65	6.0	101.0	76	mid risk

1014 rows × 7 columns

Data Preprocessing

Data preprocessing is one of the most important task for making a machine learing model.Because, with dirty data it most probabiliy will generate inaccurate outputs. There are several techniques of data cleaning like checking the null value, checking if one of the columns of data set has most empty values.In our data set the gone through a cleaning process to make sure that the data set is clean.Therefore, the processes of data cleaning is shown below to check and identify if their is any null values.

In [3]:
```
#Exploring the Details of the dataset
dF.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1014 entries, 0 to 1013
Data columns (total 7 columns):
 #   Column  Non-Null Count  Dtype
---  ------  --------------  -----
 0   AGE     1014 non-null   int64
 1   SBP     1014 non-null   int64
 2   DBP     1014 non-null   int64
 3   BS      1014 non-null   float64
 4   BTM     1014 non-null   float64
 5   HRT     1014 non-null   int64
 6   RLV     1014 non-null   object
dtypes: float64(2), int64(4), object(1)
memory usage: 55.6+ KB
```

In [4]:
```
#Replace null values with 'null'

new_dF = dF.fillna('null')
new_dF
```

Out[4]:

	AGE	SBP	DBP	BS	BTM	HRT	RLV
0	25	130	80	15.0	98.0	86	high risk
1	35	140	90	13.0	98.0	70	high risk
2	29	90	70	8.0	100.0	80	high risk
3	30	140	85	7.0	98.0	70	high risk
4	35	120	60	6.1	98.0	76	low risk
...	...	...	...	...	...	...	...
1009	22	120	60	15.0	98.0	80	high risk
1010	55	120	90	18.0	98.0	60	high risk
1011	35	85	60	19.0	98.0	86	high risk
1012	43	120	90	18.0	98.0	70	high risk
1013	32	120	65	6.0	101.0	76	mid risk

1014 rows × 7 columns

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

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1014 entries, 0 to 1013
Data columns (total 7 columns):
 #   Column  Non-Null Count  Dtype
---  ------  --------------  -----
 0   AGE     1014 non-null   int64
 1   SBP     1014 non-null   int64
 2   DBP     1014 non-null   int64
 3   BS      1014 non-null   float64
 4   BTM     1014 non-null   float64
 5   HRT     1014 non-null   int64
 6   RLV     1014 non-null   object
dtypes: float64(2), int64(4), object(1)
memory usage: 55.6+ KB
```

Exploratory Data Analysis

Exploritory Data analysis is also one of the important part for machine learning model creation. Because with the help of data analysis we can get an initial idea of the prediction of the dataset and chose a target.In the selected dataset the risk level was selected as target. Diffrent plotting with bar grap, scatter graph were discribe the co-relation between the target variable to form a relation with the variables. Overall, the data analysis part helped us a little bit to choose the feature matrix for accuracy models.

In [6]:
```
#Returning All the keys from the dataset
new_dF.keys()
```

Out[6]:
```
Index(['AGE', 'SBP', 'DBP', 'BS', 'BTM', 'HRT', 'RLV'], dtype='object')
```

In [7]:
```
#Replacing the Target Column True/False value with high/low/mild blood suger levels information
new_dF['RLV']=new_dF['RLV'].replace(['high risk','mid risk','low risk'],['True','True or False','False'])
new_dF
```

Out[7]:

	AGE	SBP	DBP	BS	BTM	HRT	RLV
0	25	130	80	15.0	98.0	86	True
1	35	140	90	13.0	98.0	70	True
2	29	90	70	8.0	100.0	80	True
3	30	140	85	7.0	98.0	70	True
4	35	120	60	6.1	98.0	76	False
...	...	...	...	...	...	...	...
1009	22	120	60	15.0	98.0	80	True
1010	55	120	90	18.0	98.0	60	True
1011	35	85	60	19.0	98.0	86	True
1012	43	120	90	18.0	98.0	70	True
1013	32	120	65	6.0	101.0	76	True or False

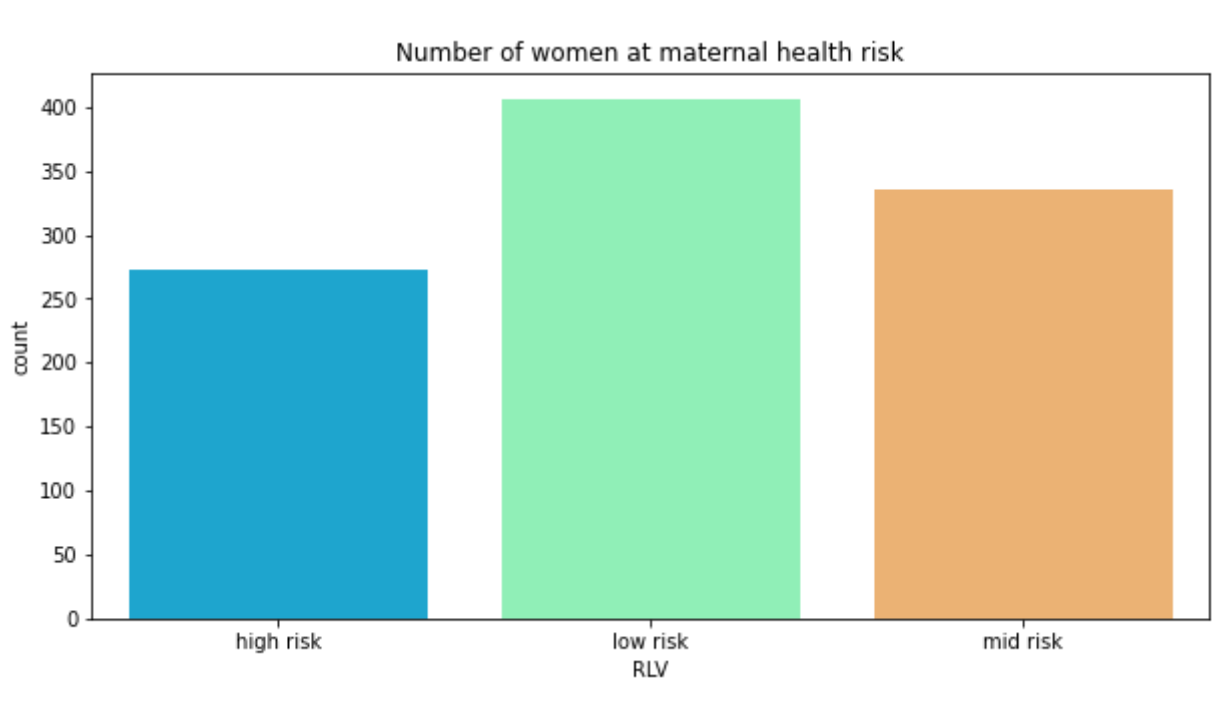
1014 rows × 7 columns

The risk factors are determined by BS(blood glucose) and changing high risk, low risk and mild risk to a true,false output, in maternal health cares from the rural areas of Bangladesh.

In [8]:
```
# showing the number of woman at  diffrent risk levels
#data distribution of the target variable(RLV)

plt.figure(figsize=(10,5))
sns.countplot(x='RLV', data=dF, palette='rainbow')
plt.title('Number of women at maternal health risk')
```

Out[8]:
```
Text(0.5, 1.0, 'Number of women at maternal health risk')
```

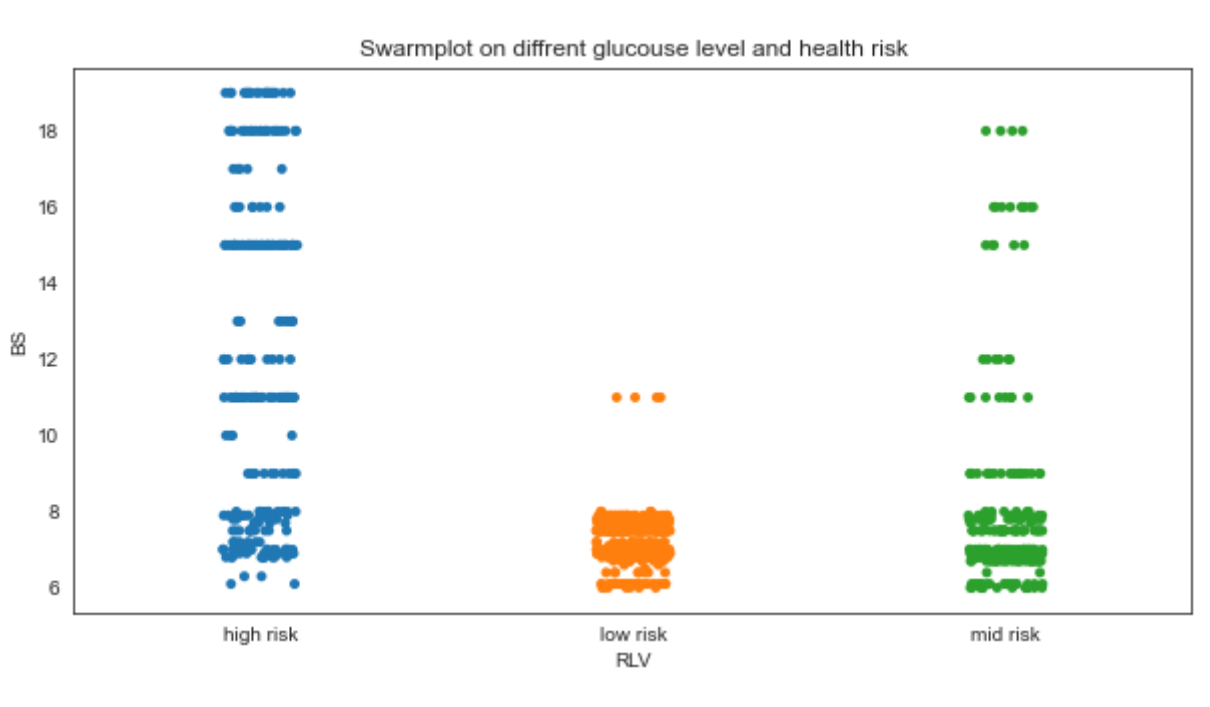


Data exploration Models

The target varible which is risk level and the corelation between the age and blood glucouse were determined by diffrent graps to understand co-relations.

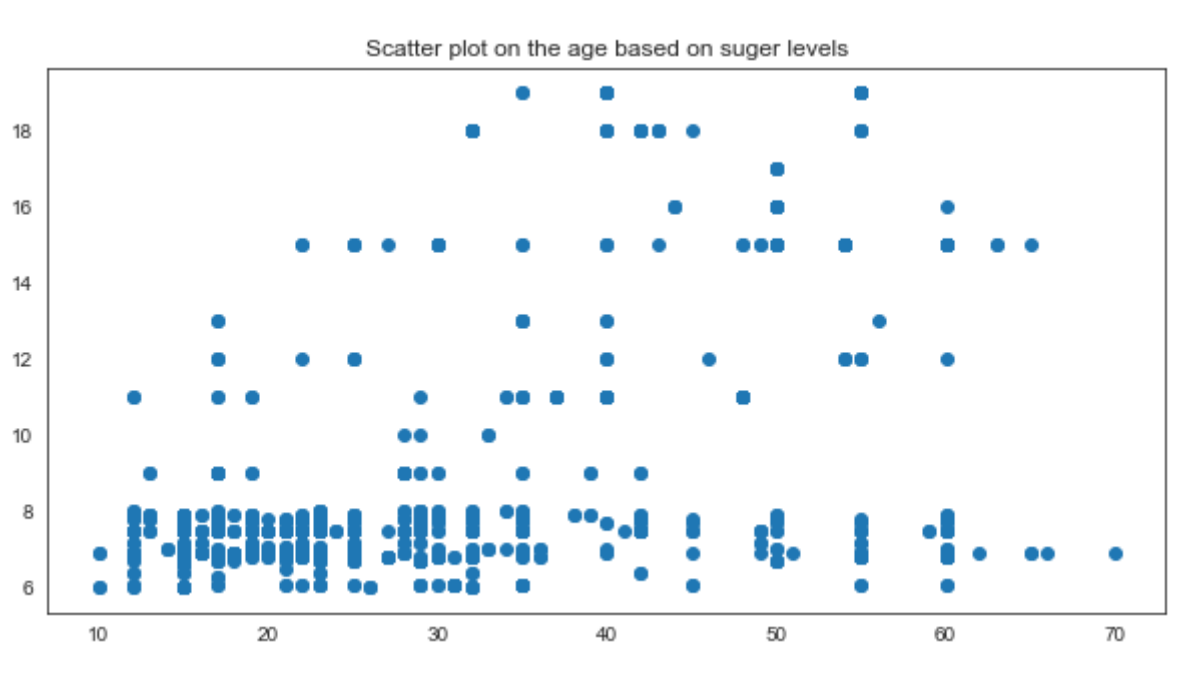
In [9]:
```
#plotting the Data of the blood glucouse levels and the level of health risk
sns.set_style("white")
plt.figure(figsize=(10,5))
sns.stripplot(x='RLV', y='BS', data=dF)
plt.title("Swarmplot on diffrent glucouse level and health risk")
```

Out[9]:
```
Text(0.5, 1.0, 'Swarmplot on diffrent glucouse level and health risk')
```



In [10]:
```
#plotting the Data if women with high glucouse level and their risk level.
plt.figure(figsize=(10,5))
plt.scatter(dF['AGE'],dF['BS'])
plt.title("Scatter plot on the age based on suger levels")
```

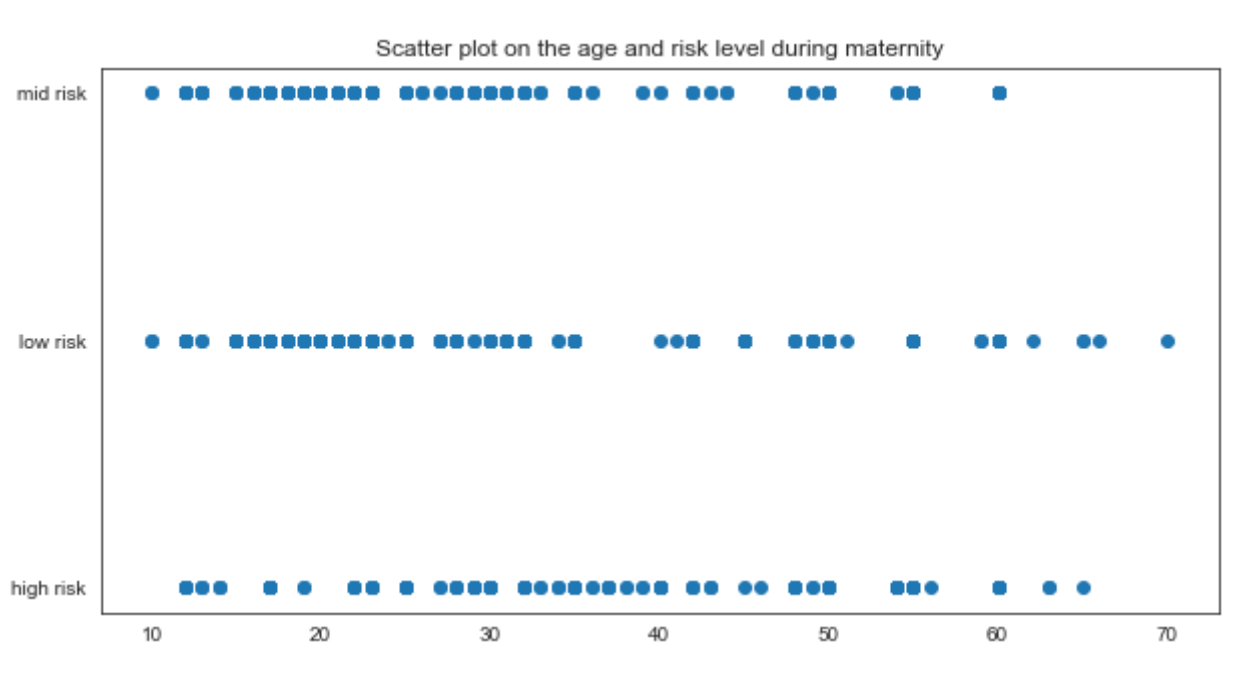
Out[10]:
```
Text(0.5, 1.0, 'Scatter plot on the age based on suger levels')
```



In [11]:
```
#plotting data to compare the ages and risk rate
plt.figure(figsize=(10,5))
plt.scatter (dF['AGE'],dF['RLV'])

plt.title("Scatter plot on the age and risk level during maternity")
```

Out[11]:
```
Text(0.5, 1.0, 'Scatter plot on the age and risk level during maternity')
```



Outcome from the plotting

Here, thought the ploting of the data based on the risk levels and the glucous level in the blood it can be determined that women with higher blood glucose level have a higher risk than a woman with lower blood glucouse. Then the age and risk level of the womens were taken ito considation for a generating a idea with the help of the plotting.

Model Development

After the initial data preprocessing and analysis also with the creation of feature matrix and separating the target variable it is time for creating the machine learning model.

Total 5 algorithms have been used to test the accuracy of this model. They are listed below:

- naive bayes
- knn
- decision tree
- logistic regression
- svm

All of them were used to test the model.

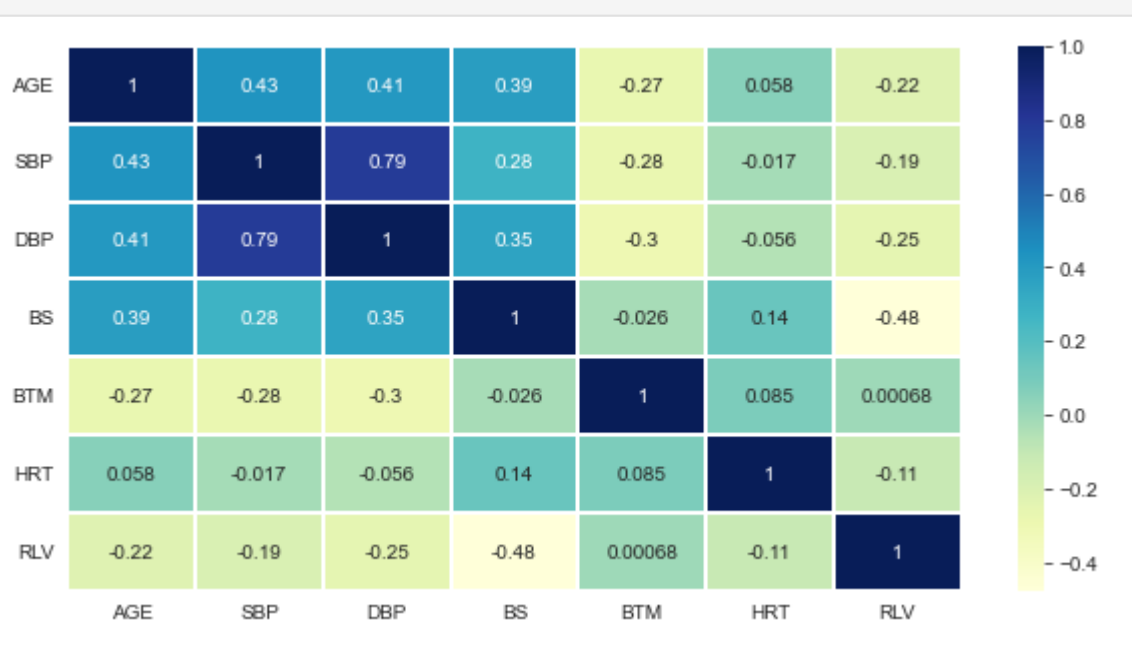
In [13]:
```
#creating the correlation Matrix for the whole Dataset

from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelEncoder

from sklearn.metrics import classification_report, confusion_matrix, precision_recall_curve, auc, roc_curve
from sklearn.tree import DecisionTreeClassifier, export_graphviz

labelencoder=LabelEncoder() #it is used to convert the categorical data into machine readable form
for column in dF.columns:
    dF[column] = labelencoder.fit_transform(dF[column])


plt.figure(figsize=(10,5))
sns.heatmap(dF.corr(),linewidths=.1,cmap="YlGnBu", annot=True)
plt.yticks(rotation=0);
```



Total 2 columns have been selected from the dataset based on correlation

SBP(Systolic blood pressure)-systolic blood pressure of 160 mm or higher can produce complication in maternal health. so, The women are at higher risk. This column is important information for the well-being and health of maternal health and predicting future risk patterns.

DBP(Diastolic blood pressure)-diastolic blood pressure of 110 mm Hg or higher can produce higher risk level during maternal health. This column is important information for the well-being and health of maternal health and predicting future risk patterns.

In [14]:
```
#creating the updated metrix using important colums
up_df=dF[['SBP', 'DBP']]
up_df
```

Out[14]:

	SBP	DBP
0	15	10
1	17	13
2	7	7
3	17	11
4	13	2
...	...	...
1009	13	2
1010	13	13
1011	6	2
1012	13	13
1013	13	4

1014 rows × 2 columns

In [15]:
```
#seprating the target variable
y=dF['RLV']
y
```

Out[15]:
```
0       0
1       0
2       0
3       0
4       1
       ..
1009    0
1010    0
1011    0
1012    0
1013    2
Name: RLV, Length: 1014, dtype: int64
```

In [16]:
```
#using data splitting functions from scikit-learn
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(up_df, y, test_size = 0.3, random_state = 16)
print("X_training shape: ", X_train.shape)
print("X_testing shape: ", X_test.shape)
perc = {}
```

```
X_training shape:  (709, 2)
X_testing shape:  (305, 2)
```

Traing and Testing Dataset

Training- 709 out of 1014 data has been choosen for training the model
Testing- 305 out of 1014 data has been choosen for the testing model.

ploting where necessary and data spliting

Data splitting is very important because without it the model cannot be generated. Data needs to be spitted into two parts one is for training the dataset and another part is for testing the dataset.In our project there were total 1014 instances of data. where 709 of 1014 is used for training the model. the rest 305 data were used for testing the model's accuracy. The spitting mostly depends on user as both training and testing is equally important for the model to be accurate.

In [17]:
```
#Naive bayes
# importing the necessary package to use the naive bayes algorithm
from sklearn.naive_bayes import GaussianNB
model_nb = GaussianNB()
model_nb.fit(X_train, y_train)
y_prediction_nb = model_nb.predict(X_test)
score_nb = metrics.accuracy_score(y_prediction_nb, y_test).round(4)
print('The accuracy of the model based on Naive Bayes Algorithm is {}%'.format(score_nb*100))
perc['Naive-bayes'] = score_nb
```

```
The accuracy of the model based on Naive Bayes Algorithm is 50.82%
```

In [18]:
```
#Knn
# importing the necessary package to use the knn algorithm
from sklearn.neighbors import KNeighborsClassifier
model_knn = KNeighborsClassifier(n_neighbors=3)
model_knn.fit(X_train, y_train)
y_prediction_knn = model_knn.predict(X_test)
score_knn = metrics.accuracy_score(y_prediction_knn, y_test).round(4)
print('The accuracy of the model based on K Nearest Neighbour Algorithm is {}%'.format(score_knn*100))
perc['Knn'] = score_knn
```

```
The accuracy of the model based on K Nearest Neighbour Algorithm is 54.75%
```

In [19]:
```
# importing the necessary package to use the decision tree algorithm
from sklearn.tree import DecisionTreeClassifier
model_dt = DecisionTreeClassifier(random_state=4)
model_dt.fit(X_train, y_train)
y_prediction_dt = model_dt.predict(X_test)
score_dt = metrics.accuracy_score(y_prediction_dt, y_test).round(4)
print('The accuracy of the model based on Decision Tree Algorithm is {}%'.format(score_dt*100))
perc['Decision-tree'] = score_dt
```

```
The accuracy of the model based on Decision Tree Algorithm is 58.36%
```

In [20]:
```
#logistic regression algorithm
# importing the necessary package to use the logistic regression algorithm
from sklearn.linear_model import LogisticRegression
model_lr = LogisticRegression()
model_lr.fit(X_train, y_train)
y_prediction_lr = model_lr.predict(X_test)
score_lr = metrics.accuracy_score(y_prediction_lr, y_test).round(4)
print('The accuracy of the model based on Logistic Regression Algorithm is {}%'.format(score_lr*100))
perc['Logistic-regression'] = score_lr
```

```
The accuracy of the model based on Logistic Regression Algorithm is 55.74%
```

In [21]:
```
#svm
# importing the necessary package to use the svm algorithm
from sklearn import svm
model_svm = svm.SVC()
model_svm.fit(X_train, y_train)
y_prediction_svm = model_svm.predict(X_test)
score_svm = metrics.accuracy_score(y_prediction_svm, y_test).round(4)
print('The accuracy of the model based on Support Vector Machine Algorithm is {}%'.format(score_svm*100))
perc['SVM'] = score_svm
```

```
The accuracy of the model based on Support Vector Machine Algorithm is 58.69%
```

In [51]:
```
max_accuracy = max(perc.values())
max_model = max(perc, key=perc.get)
print(f'The maximum accuracy from all these algorithms is {max_model} with {max_accuracy*100}%')
```

```
The maximum accuracy from all these algorithms is SVM with 58.69%
```

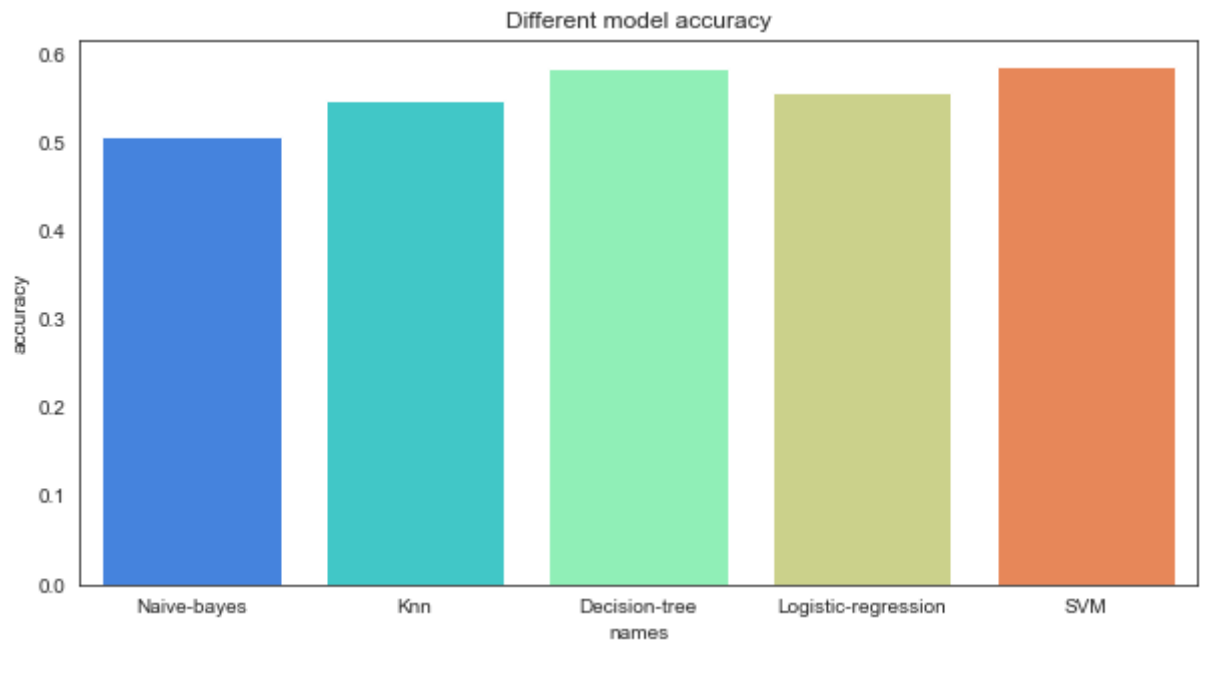
In [48]:
```
score = pd.DataFrame(list(perc.keys()), columns=['names'])
score['accuracy'] = list(perc.values())
score
```

Out[48]:

	names	accuracy
0	Naive-bayes	0.5082
1	Knn	0.5475
2	Decision-tree	0.5836
3	Logistic-regression	0.5574
4	SVM	0.5869

In [50]:
```
plt.figure(figsize=(10,5))
sns.barplot(x='names', y='accuracy', data=score, palette='rainbow')
plt.title('Different model accuracy')
```

Out[50]:
```
Text(0.5, 1.0, 'Different model accuracy')
```



Comparison of Models

The 5 models used for the accuracy prediction of the data sets were used to determine the percentage of accuracy of 50.82% based on Support Naive Bayes Algorithm again the higest accuracy percentage of 58.69% based on SVM algorithm, again the second highest decision percentage for the decision tree Algorithm was 58.36%. based on the model accuracy percentage were determined to produce accuracy value by plotting the 5 models and plotting them on a bar graph to understand more clearly that SVM has a little more accuracy from the other 4 models. SVM or Support Vector Machine is a linear model for classification and regression problems for the problem. The reason SVM can be useful for the datasets can solve linear and non-linear problems and work well for a more accurate result.

Discussion

This model is created based on the Maternal Health Risk Data Set which has the data of high,low and mild risk. This model has an accuracy of 50.82%. Which is not the best result or outcome in this data set which has 1014 numbers of instances. This desions where made by diffrent modeles chossing the most relevent data through diffrent plotting and graphs. still as the accuracy is not even near the 70% it might not be an ideal result. which might be the solved if their where more attributes and data sets. If we had more columns which had more correlatation with the other variable the prediction could have been more better. As, it was based on a real life dataset, the prediction can also be called a realistic if it predicts the correct result.