Car Class Prediction

Problem Description:

- The target variable is the class of the car which may be one of, 0 –bus, 1– Opel Manta, 2 – Saab, 3 – Van.
- The people who are going to buy a new automobile for their constant activities will be having these types of problem.
- Using advanced machine learning techniques appropriate for statistical analysis, we can help solve the challenge mentioned above. This project aims to build a model that classifies a car's make and model-given attributes of the automotive.

Data Set:

- To Build such a model, we take the data from the source which is the Dataset file: 'cars_class.csv'
- This is a multi-class classification data set.
- The data set has 719 samples.
- There are 18 numerical features

Features of the Data:

- Comp: Compactness
- Circ: Circularity
- D.Circ: Distance Circularity
- Rad.Ra: Radius ratio
- Pr. Axis.Ra: pr. axis aspect ratio
- Max.L. Ra: max. length aspect ratio
- Scat.Ra: scatter ratio
- Elong: elongatedness
- Pr. Axis.Rect: pr. axis rectangularity
- Max.L. Rect: max. length rectangularity

- Sc. Var.Maxis: scaled variance along major axis
- Sc. Var.maxis: scaled variance along minor axis
- Ra. Gyr: scaled radius of gyration
- Skew. Maxis: skewness about major axis
- Skew. maxis: skewness about minor axis
- Kurt. maxis: kurtosis about minor axis
- Kurt. Maxis: kurtosis about major axis
- Holl.Ra: hollows ratio

Data Exploration:

- The first 5 rows of data are as follows
- **data.head(5)**

index	ID	Comp	Circ	D.Circ	Rad.Ra	Pr.Axis.Ra	Max.L.Ra	Scat.Ra	Elong	Pr.Axis.Rect	Max.L.Rect	Sc.Var.Maxis	Sc.Var.maxis	Ra.Gyr	Skew.Maxis	Skew.Maxis	Kurt.Maxis	Kurt.Maxis	Holl.Ra	Class
0	1	88	39	70	166	66	7	148	44	19	134	167	332	143	69	5	13	193	201	0
1	2	85	35	64	129	57	6	116	57	17	125	138	200	123	65	1	23	196	203	3
2	3	91	41	84	141	57	9	149	45	19	143	170	330	158	72	9	14	189	199	3
3	4	102	54	98	177	56	10	219	31	25	171	219	706	223	72	5	17	186	196	1
4	5	87	39	74	152	58	6	151	44	19	136	174	337	140	70	1	33	187	196	2

- **data.shape**

  The shape of the data is (719,20)

  data. dtypes

- ID - int64
- Comp - int64
- Circ - int64
- D.Circ - int64
- Rad.Ra - int64
- Pr. Axis.Ra - int64
- Max.L. Ra - int64
- Scat.Ra - int64
- Elong - int64
- Pr. Axis.Rect - int64
- Max.L. Rect - int64
- Sc. Var.Maxis - int64

- Sc. Var.maxis - int64
- Ra. Gyr - int64
- Skew.Maxis - int64
- Skew.maxis - int64
- Kurt.maxis - int64
- Kurt.Maxis - int64
- Holl.Ra - int64
- Class - int64

The Summary of the statistics values in the data are:

- **data.describe()**

index	ID	Comp	Circ	D.Circ	Rad.Ra	Pr.Axis.Ra	Max.L.Ra	Scat.Ra	Elong	Pr.Axis.Rect	Max.L.Rect	Sc.Var.Maxis	Sc.Var.maxis	Ra.Gyr	Skew.Maxis	Skew.maxis	Kurt.maxis	Kurt.Maxis	Holl.Ra	Class
count	719.0	719.0	719.0	719.0	719.0	719.0	719.0	719.0	719.0	719.0	719.0	719.0	719.0	719.0	719.0	719.0	719.0	719.0	719.0	719.0
mean	360.0	93.44	44.85	81.72	168.58	61.85	8.63	168.14	41.08	20.53	148.03	188.17	436.22	174.73	72.68	6.34	12.39	188.8	195.41	1.47
std	207.7	8.11	6.15	15.53	33.81	8.26	4.92	32.94	7.76	2.56	14.56	31.24	174.96	32.15	7.54	4.86	8.74	6.05	7.24	1.13
min	1.0	73.0	33.0	40.0	105.0	47.0	2.0	112.0	26.0	17.0	118.0	130.0	184.0	109.0	59.0	0.0	0.0	176.0	181.0	0.0
25%	180.5	87.0	40.0	70.0	141.0	57.0	6.0	146.0	33.0	19.0	137.0	167.0	317.0	149.0	68.0	2.0	5.5	184.0	190.0	0.0
50%	360.0	93.0	44.0	79.0	166.0	61.0	8.0	157.0	43.0	20.0	146.0	178.0	362.0	174.0	72.0	6.0	11.0	188.0	196.0	1.0
75%	539.5	99.0	49.0	96.0	194.5	65.0	10.0	197.5	46.0	23.0	159.0	216.0	584.5	198.0	76.0	9.0	18.0	193.0	201.0	2.0
max	719.0	119.0	59.0	110.0	333.0	138.0	55.0	265.0	61.0	29.0	188.0	320.0	1018.0	262.0	135.0	22.0	41.0	204.0	211.0	3.0

- There are no missing values from the given dataset.
- **data.isnull().sum()**

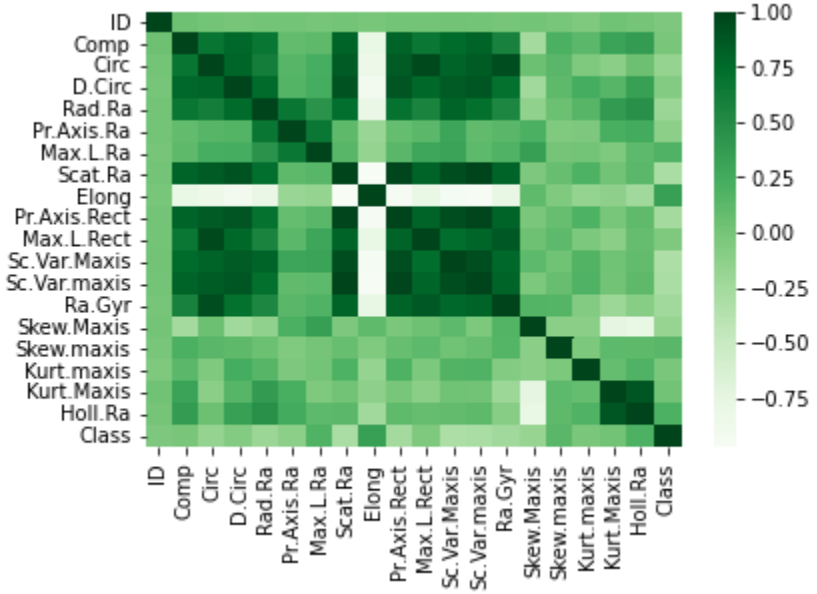
  ID 0
  Comp 0
  Circ 0
  D.Circ 0
  Rad.Ra 0
  Pr.Axis.Ra 0
  Max.L.Ra 0
  Scat.Ra 0
  Elong 0
  Pr.Axis.Rect 0
  Max.L.Rect 0
  Sc.Var.Maxis 0
  Sc.Var.maxis 0

Ra.Gyr	0
Skew.Maxis	0
Skew.maxis	0
Kurt.maxis	0
Kurt.Maxis	0
Holl.Ra	0
Class	0

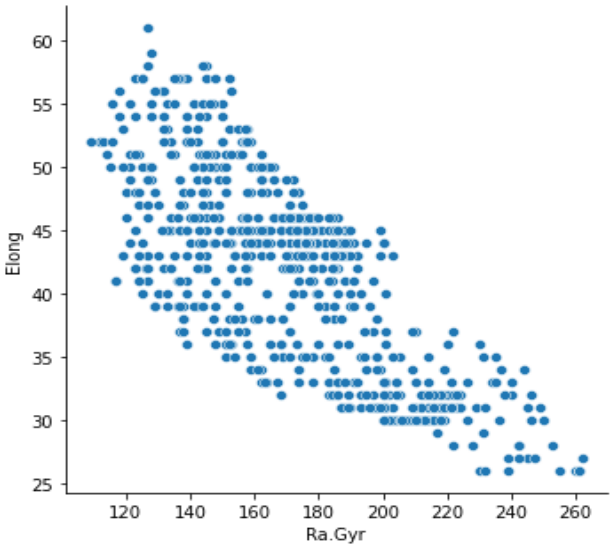
Feature Engineering:

- I feel that there should be no dropping the features because the all features which are present in the data are the parameters when the automobile is tested to get into the market.
- These parameters are very useful in getting the size and shape of the Automobile.
- So, I have used the correlation between the features and I have dropped some feature for making a better model.

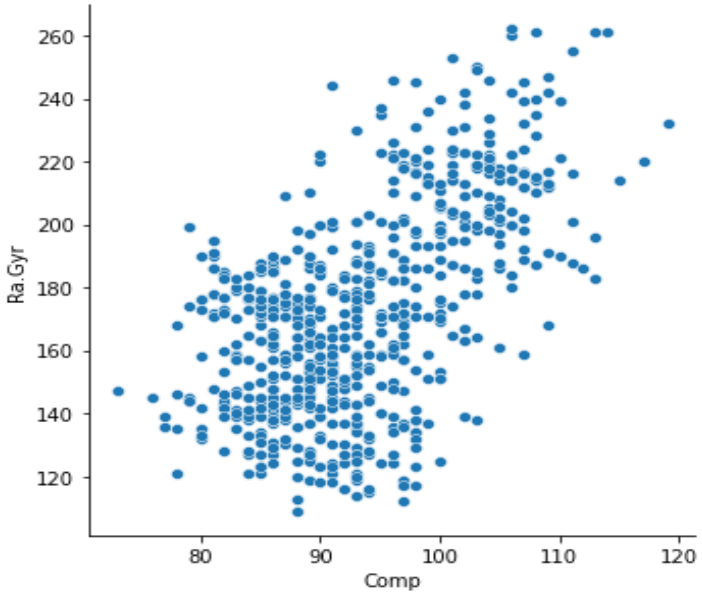
```
correlation = data.corr()
sns.heatmap(correlation, xticklabels= correlation.columns , yticklabels=
correlation.columns, cmap = 'Greens')
```



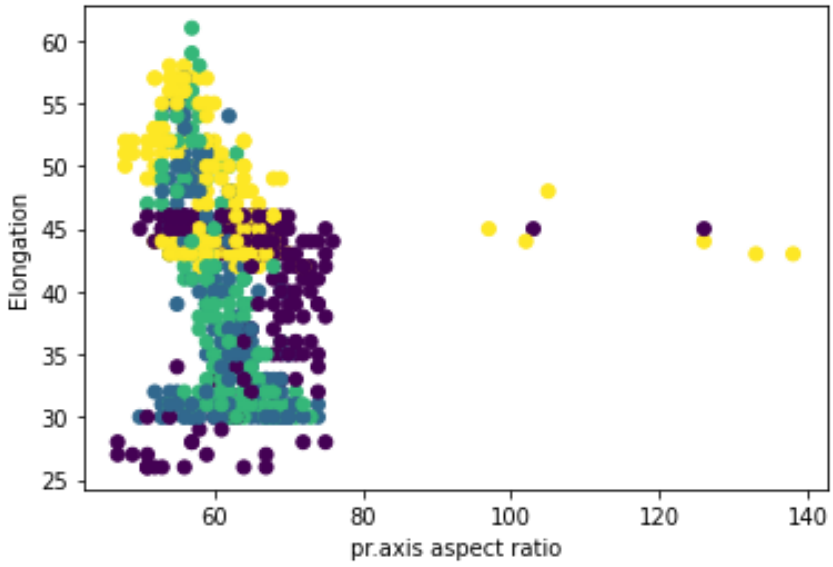
- The relation between Radius of Gyration and Elongation feature are:



- The relation between Circularity and Elongation is:

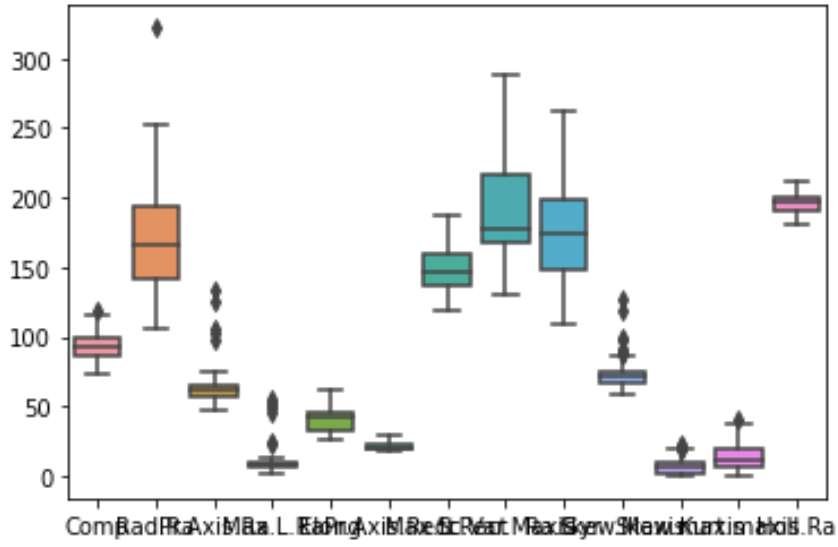


- The relation between Pr. aspect ratio and elongation on the basis of the class of the automobile.

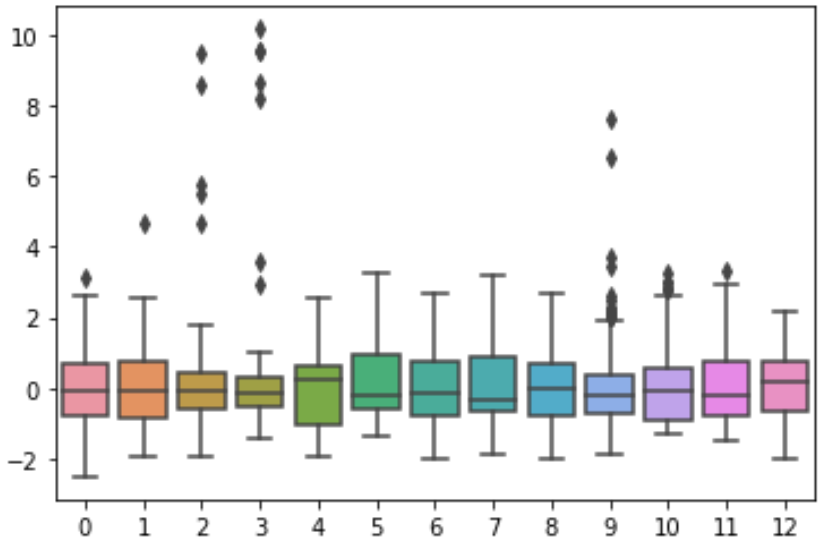


Data Pre-Processing:

- By the means of correlation between the features of the data.
- I have deleted some feature for better training of the data and to get better results in our problem.
- The Deleted features are:
- 'ID', ‘Circ’, ‘Scat.Ra’, ‘D. Circ’, ‘Kurt.Maxis’, ‘Sc. Var.maxis'
- Now, the data has been divided into 70% training data and 20% testing data.
- Mean and Variance of the various features in the data as follows:



- By using the StandardScaler The features are now standardized to mean 0 and variance 1.



Data Modelling:

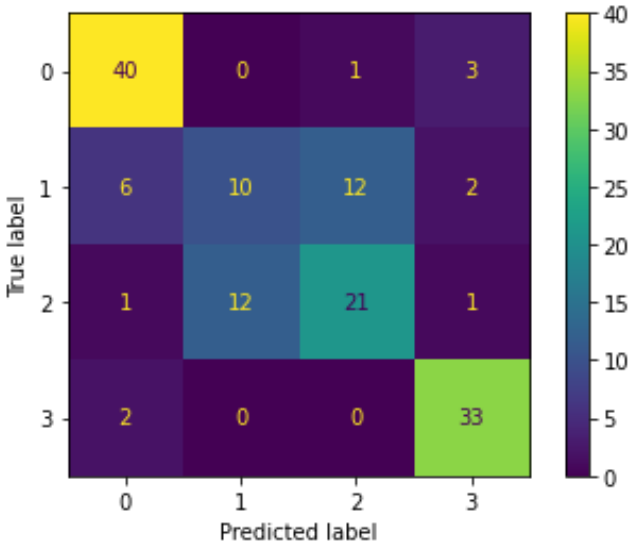
- As this is a classification problem because we need to find the class of the automobile.
- I have used several ML techniques to get better results.
- Logistic Regression
- SGDClassifier
- Support Vector Classifier
- KNeighborsClassifier
- DecisionTreeClassifier
- RandomForestClassifier
- GradientBoostingClassifier
- GaussianNB

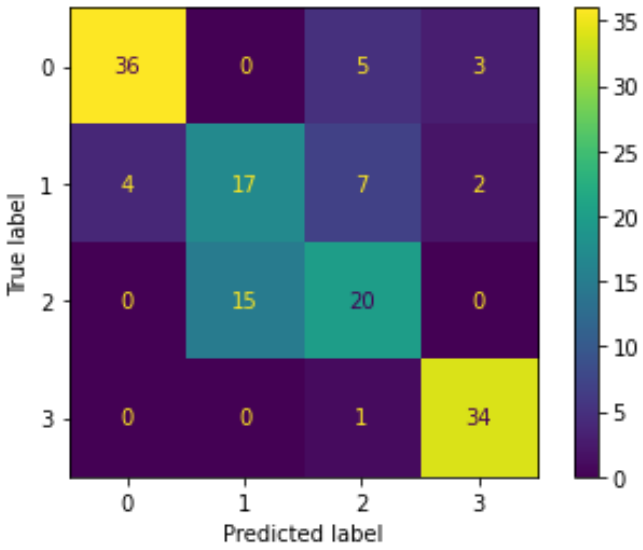
The Accuracy scores of the testing data for the above techniques respectively are:

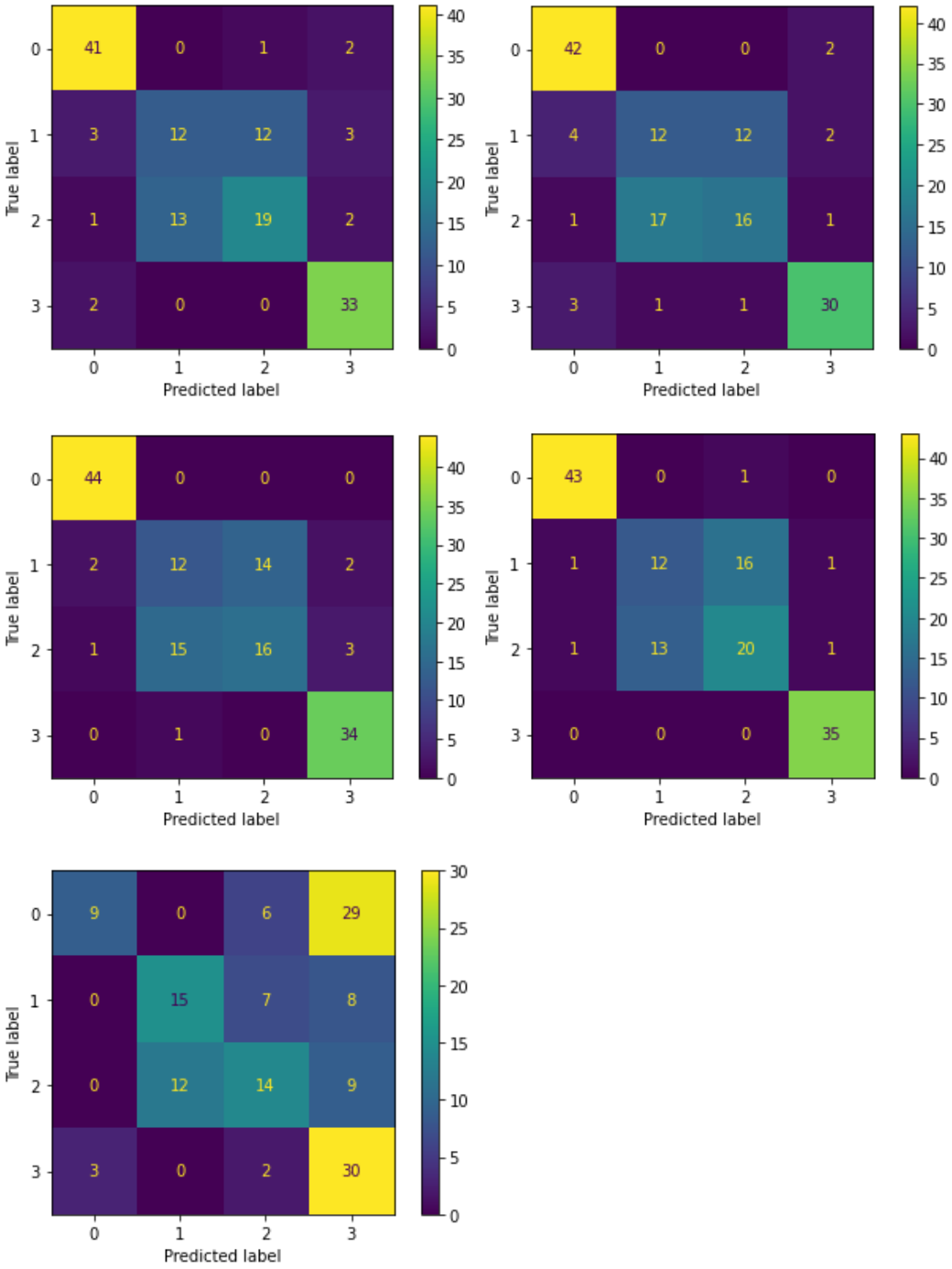
- 0.7222222222222222
- 0.7430555555555556

- 0.7291666666666666
- 0.6944444444444444
- 0.6736111111111112
- 0.7361111111111112
- 0.7638888888888888
- 0.4722222222222222

The Confusion matrices for the data is:

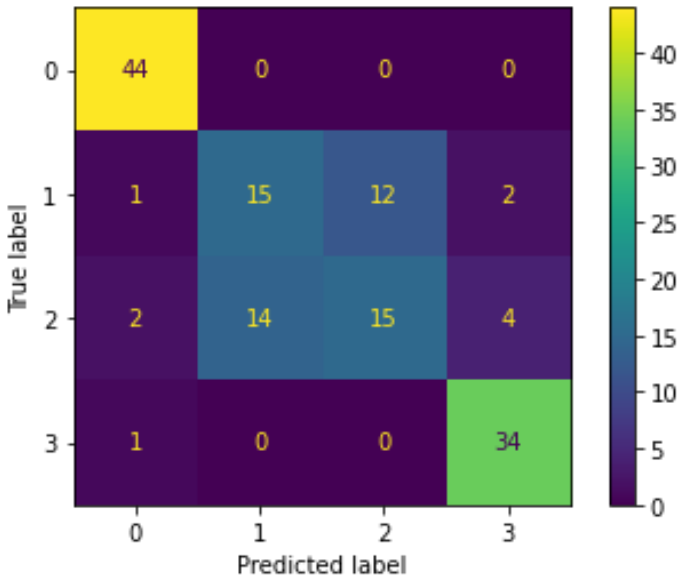




- 

- By, the testing scores I have selected Support Vector Classifier, Random Forest Classifier and Gradient Boosting Classifier for Hyper-Parameter Tuning.
- After Parameter Tuning the final ML Technique is Random Forest Classifier.
- final_model = RandomForestClassifier(max_depth=8, n_estimators= 112)

- final_model.fit(X_train, y_train)
- final_model.score(X_train, y_train)
- 0.9843478260869565
- final_model.score(X_test, y_test)
- 0.75
- The Confusion Matrix for the final model is:

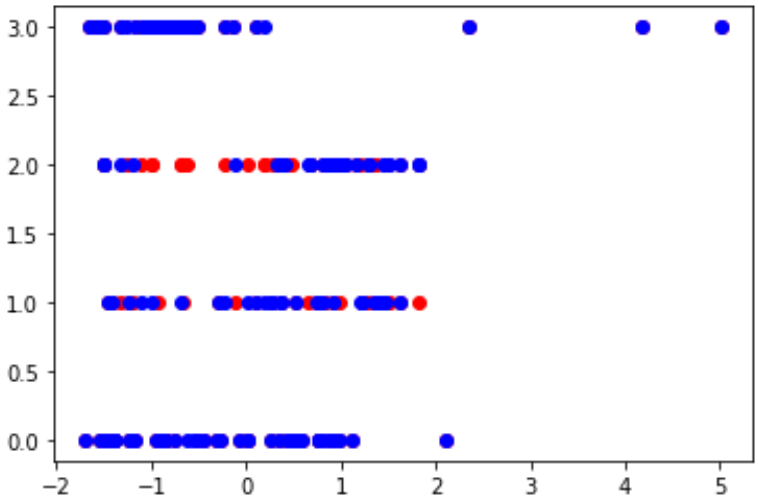


- The importance of the features after modelling the data is:

- [0.06734188, 0.0586756, 0.06596381, 0.15073984, 0.11087278, 0.06130449, 0.0986932, 0.11940415, 0.0532953, 0.06241242, 0.0400713, 0.04590767, 0.06531757]

- The scatter graph of the predicated data to the test data is as follows:

- Test Data – Blue
- Predicated Data - Red


3.0
2.5
2.0
1.5
1.0
0.5
0.0
−2
−1
0
1
2
3
4
5