

PROJECT NAME:

Classify The right Customer

PROJECT DONE BY:

Sayed Ahmed

ABOUT:

In this project I have worked with a particular dataset. The problem is to detect discrete values or classes or a potential customer. I have used different classification algorithms to solve the problem. The code for each algorithm is added to this repository. I have compared the results of multiple classifiers and opted the best ones for this problem.

USED LIBRARIES AND IMPLEMENTED KNOWLEDGE:

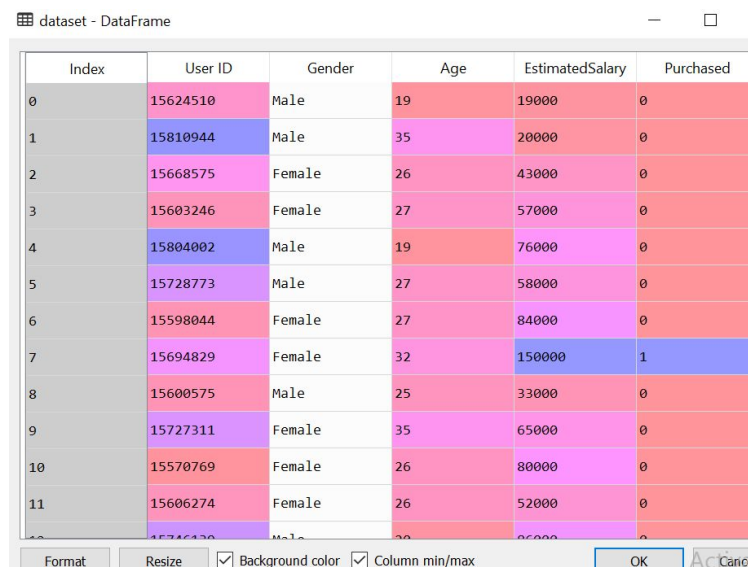
The code has been trained using python and the knowledge gained from course is implemented here.

1. Pandas
2. Matplotlib
3. Sickit learn

implemented Knowledge from course work

1. Scaling data
2. Reducing overfitting, underfitting
3. Data preprocessing
4. Ensemble Algorithm
5. Model Evaluation, Confusion Matrixes
6. Plotting predicted result
7. Dimensionality reduction

DATASET



Index	User ID	Gender	Age	EstimatedSalary	Purchased
0	15624510	Male	19	19000	0
1	15810944	Male	35	20000	0
2	15668575	Female	26	43000	0
3	15603246	Female	27	57000	0
4	15804002	Male	19	76000	0
5	15728773	Male	27	58000	0
6	15598044	Female	27	84000	0
7	15694829	Female	32	150000	1
8	15600575	Male	25	33000	0
9	15727311	Female	35	65000	0
10	15570769	Female	26	80000	0
11	15606274	Female	26	52000	0

The dataset contains user purchase history of a certain product along with user information. From the dataset I have trained a model that can detect potential customer.

The dataset contains userId, Gender, Age, Estimated Salary, Car Purchase column, We can see that Age and estimated salary will have influence on users purchasing decision. To reduce dimensionality, I'm excluding other variables except for Age and Estimated salary, so these variables are independent variables and purchase column is dependent variable and that what we want to predict.

I have tried different classification algorithms to predict and compare them to find the best model for this problem.

The algorithm I have trained using this dataset are

1. Logistic regression
2. KNN classifier
3. SVM classifier
4. Decision tree classifier
5. Random Forest classifier

I have included Confusion matrix and plot of both training and test result, so that the performance of the model could be understood properly.

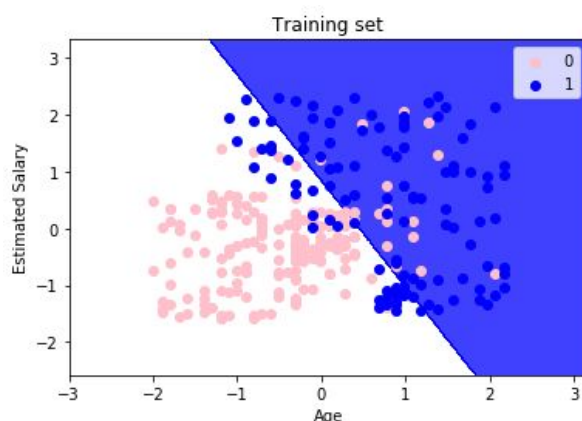
Logistic regression Result:

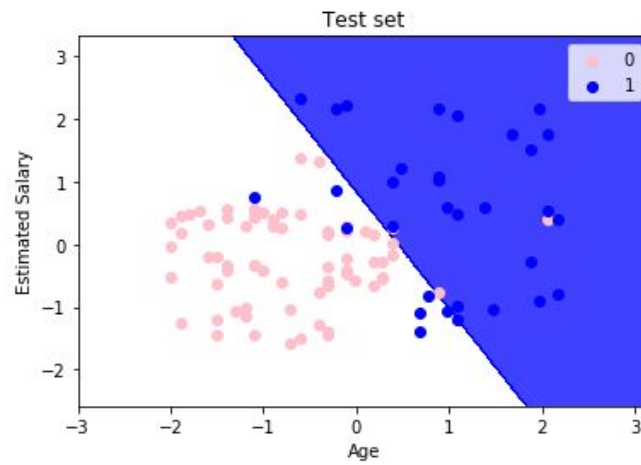
cm - NumPy array

	0	1
0	65	3
1	8	24

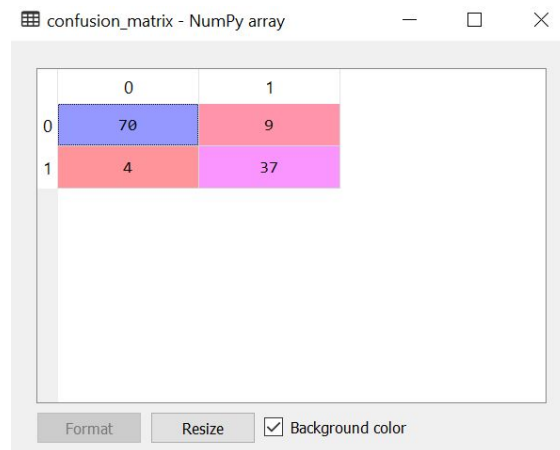
Confusion Matrix

In the confusion Matrix for logistic regression shows us that the model failed to classify 11 data points.





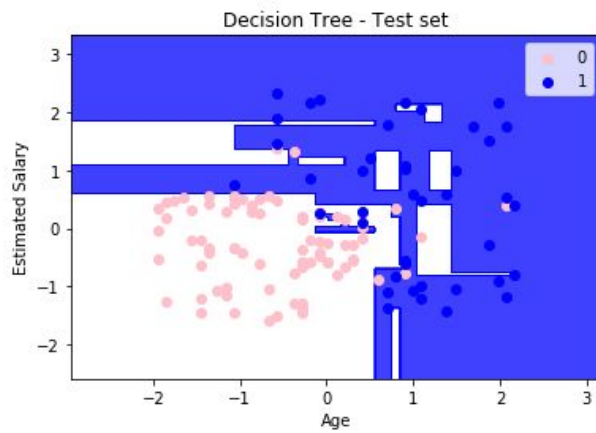
Decision Tree:



Confusion Matrix

In the confusion Matrix for Decision tree shows Total 13 wrong classification





From the plot we can see that it's overfitting the data a little

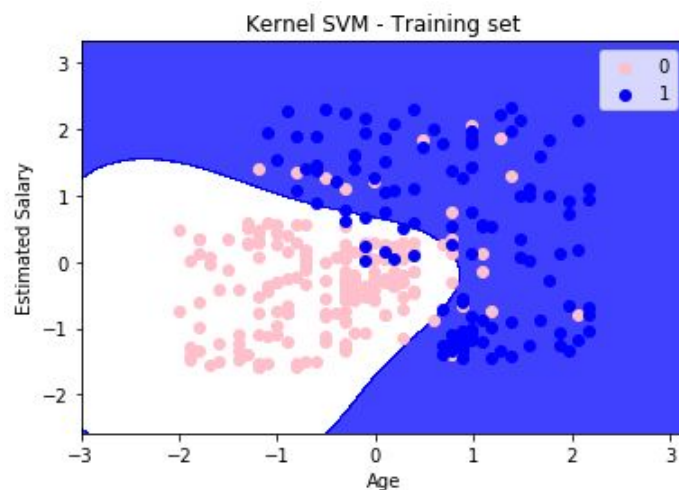
Kernel-SVM

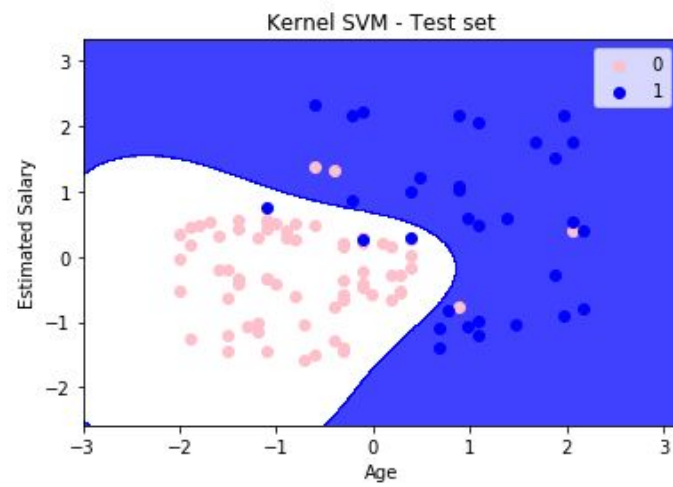
cm - NumPy array

	0	1
0	64	4
1	3	29

Confusion Matrix

The model failed to identify total 7 data points



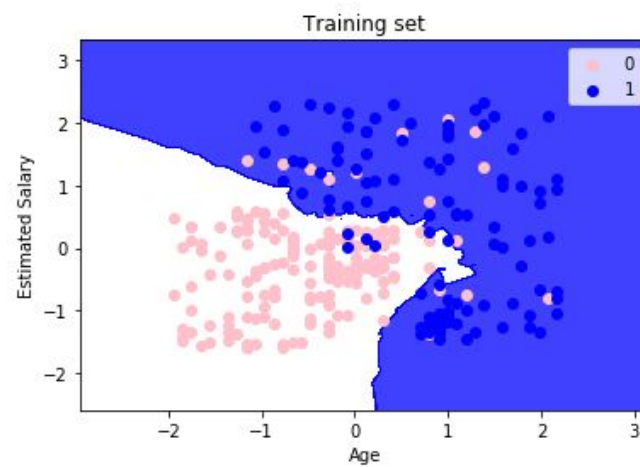


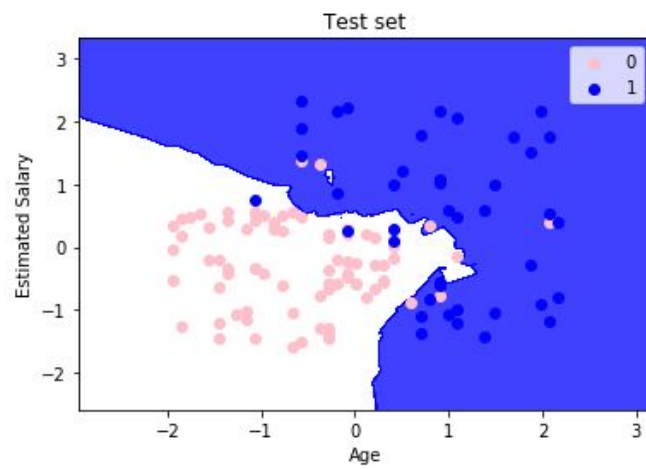
KNN-Classifier

cm - NumPy array

	0	1
0	73	6
1	4	37

Confusion Matrix

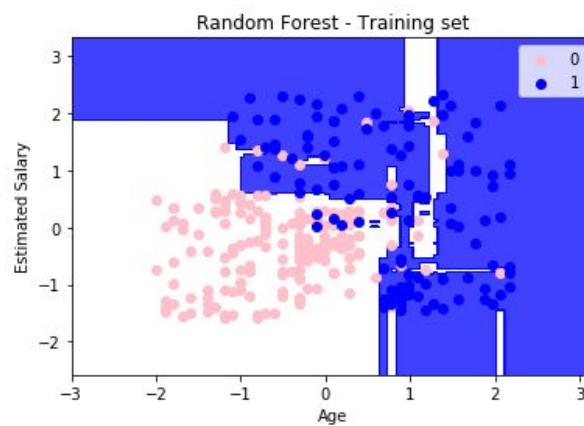


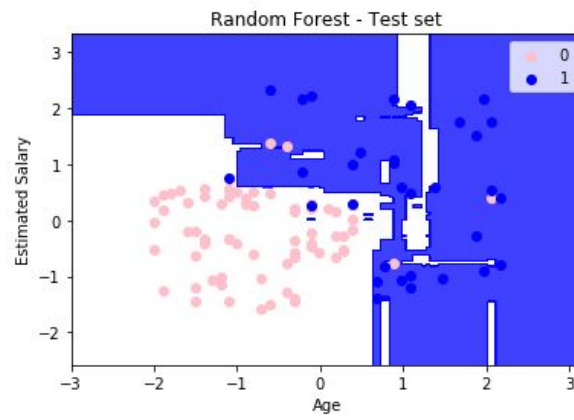


Random Forest:

cm - NumPy array

	0	1
0	63	5
1	3	29





Conclusion:

After training all the models Kernel SVM and KNN classifier is performing better for the dataset, random forest classifier is learning data too well and somewhat failing to generalize for new data.