

Introduction to AI and ML

CSL236

Project Report



Faculty name: Akanksha Kaushik

Student name: Riya Gupta

Roll No.:20csu269

Semester:5TH

Group: AIML-B

Department of Computer Science and Engineering
The NorthCap University, Gurugram- 122001, India
Session 2022-23

Table of Contents

S.No		Page No.
1.	Project Description	3
2.	Problem Statement	3
3.	Analysis 3.1 Hardware Requirements 3.2 Software Requirements	4
4.	Design 4.1 Data/Input Output Description: 4.2 Algorithmic Approach / Algorithm / DFD / ER diagram/Program Steps	5
5.	Implementation and Testing (stage/module wise)	7
6.	Output (Screenshots)	9
7.	Conclusion and Future Scope	11

Ionosphere Classification

1. Project Description

The ionosphere, an ionized part of the earth's atmosphere that starts from an altitude of about sixty kilometers from above the sea level and extends up to about the distance of thousand kilometers above the sea level, is one of the most important layers of the atmosphere. The ionosphere is very important in radio communication and its study should be of great importance. Any variations in the radiations in the ionosphere can affect entire communication systems. For this reason, the radar returns from the ionosphere should be studied thoroughly. There are mixed radar returns from the ionosphere. Good radar returns can be used for further study which helps to know more about the ionosphere.

Thus, some machine learning methods are needed to delineate useful and non-useful radar returns. The various machine learning algorithms (classical learning algorithms) is tested in this study. Algorithm used in this are K-nearest neighbor, Decision Tree, Random Forest classification and SVM. Highest Accuracy of 93% received in Random Forest with error rate of 6%.

2. Problem Statement

Study of the ionosphere is important for research in various domains. Especially in communication systems, this study holds a great importance. In ionospheric research, there is a need to delineate useful and non-useful radar returns from the ionosphere. The useful radar returns can be used for further analysis and non-useful radar returns can be discarded. When the usefulness of radar returns is analyzed by humans, it is simply time-consuming and is prone to more human errors. In this project, the famous dataset by Johns Hopkins University is used. In the dataset, there are two labels viz. "Good" and "Bad". The radar returns which showed evidence of some type of structure in the region of ionosphere were labeled as "Good" returns. "Bad" returns are the returns which didn't pass signals through the ionosphere

3. Analysis

3.1 Hardware Requirement

- Intel(R) Core(TM) i5-1035G1 CPU @ 1.00GHz 1.19 GHz processor
- 8.00 GB RAM
- 64-bit operating system
- x64-based processor
- Edition-Windows 10 Home Single Language
- Version-22H2
- OS build-19045.2251
- Experience-Windows Feature Experience Pack 120.2212.4180.0

3.2 Software Requirement

- Project is performed on Knime Analysis platform Version-2022
- Size of software- 893 bytes (893 bytes)
- Other great apps like KNIME are RStudio, Orange, IBM SPSS Statistics and RapidMiner

4. Design

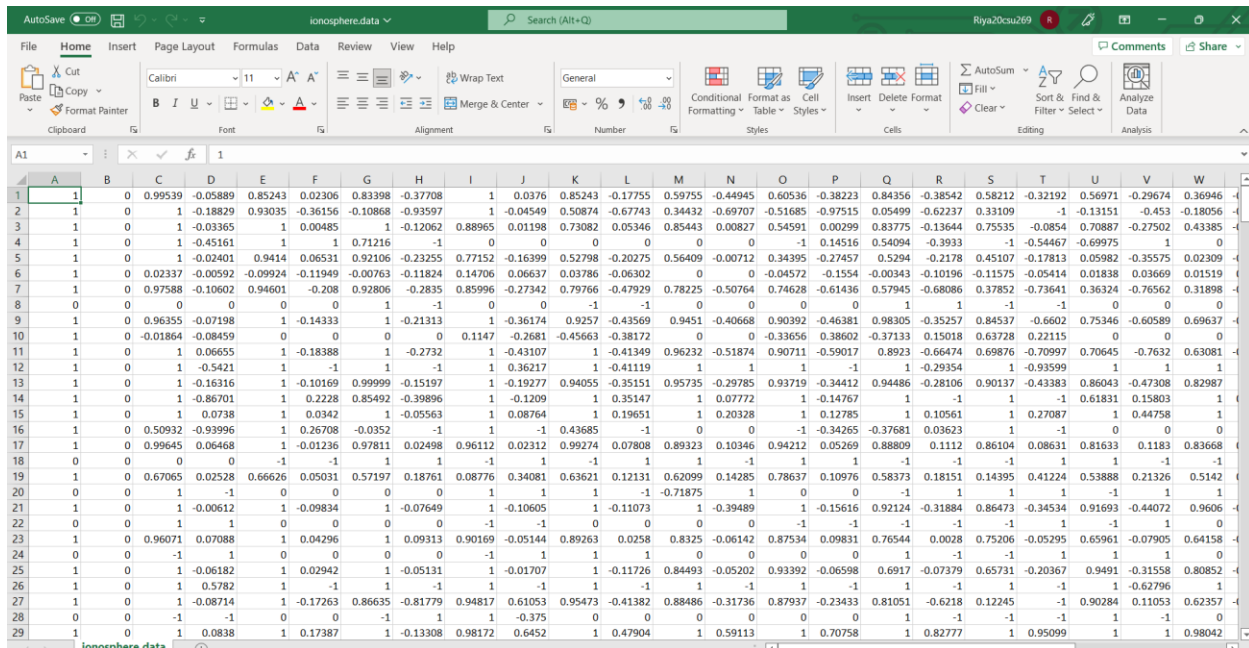
4.1 Data Description

This radar data was collected by a system in Goose Bay, Labrador. This system consists of a phased array of 16 high-frequency antennas with a total transmitted power on the order of 6.4 kilowatts. See the paper for more details. The targets were free electrons in the ionosphere. "Good" radar returns are those showing evidence of some type of structure in the ionosphere. "Bad" returns are those that do not; their signals pass through the ionosphere. Received signals were processed using an autocorrelation function whose arguments are the time of a pulse and the pulse number. There were 17 pulse numbers for the Goose Bay system. Instances in this database are described by 2 attributes per pulse number, corresponding to the complex values returned by the function resulting from the complex electromagnetic signal.

Attribute Information:

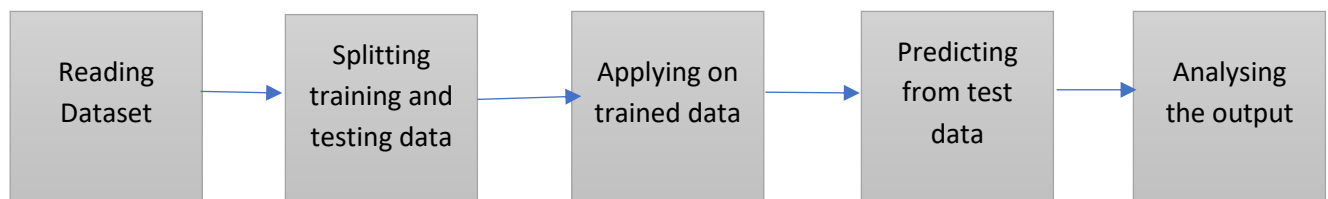
- All 34 are continuous
- The 35th attribute is either "good" or "bad" according to the definition summarized above. This is a binary classification task.

Dataset:https://www.kaggle.com/code/mfarahmand98/classification-of-the-ionosphere-dataset/data?select=ionosphere_data.csv



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W
1	1	0	0.99539	-0.05889	0.85243	0.02306	0.83398	-0.37708	1	0.0376	0.85243	-0.17755	0.59755	-0.44945	0.60536	-0.38223	0.84356	-0.38542	0.58212	-0.32192	0.56971	-0.29674	0.36946
2	1	0	1	-0.18829	0.93035	-0.36156	-0.10868	-0.93597	1	-0.04549	0.50874	-0.67743	0.34432	-0.69707	-0.51685	-0.97515	0.05499	-0.62237	0.33109	-1	-0.13151	-0.453	-0.18056
3	1	0	1	-0.03365	1	0.00485	1	-0.12062	0.88965	0.01198	0.73082	0.05346	0.85443	0.00827	0.54591	0.00299	0.83775	-0.13644	0.75535	-0.0854	0.70887	-0.27502	0.43385
4	1	0	1	-0.45161	1	1	0.71216	-1	0	0	0	0	0	0	0	-1	0.14516	0.54094	-0.3933	-1	-0.54467	-0.69975	1
5	1	0	1	-0.02401	0.9414	0.06531	0.92106	-0.23255	0.77152	-0.16399	0.52798	-0.20275	0.56409	-0.00712	0.34395	-0.27457	0.5294	-0.2178	0.45107	-0.17813	0.05982	-0.35575	0.02309
6	1	0	0.02337	-0.00592	-0.09924	-0.11949	-0.00763	-0.11824	0.14706	0.06637	0.03786	-0.06302	0	0	-0.04572	-0.1554	-0.00343	-0.10196	-0.11575	-0.05414	0.01838	0.03669	0.01519
7	1	0	0.97588	-0.10602	0.94601	-0.208	0.92806	-0.2835	0.85996	-0.27342	0.79766	-0.47929	0.78225	-0.50764	0.74628	-0.61436	0.57945	-0.68086	0.37852	-0.73641	0.36324	-0.76562	0.31898
8	0	0	0	0	0	0	0	1	-1	0	0	-1	-1	0	0	0	0	1	1	-1	0	0	0
9	1	0	0.96355	-0.07198	1	-0.14333	1	-0.21313	1	-0.36174	0.9257	-0.43569	0.9451	-0.40668	0.90392	-0.46381	0.98305	-0.35257	0.84537	-0.6602	0.75346	-0.60589	0.69637
10	1	0	-0.01864	-0.08459	0	0	0	0	0.1147	-0.2681	-0.45663	-0.38172	0	0	-0.33656	0.38602	-0.37133	0.15018	0.63728	0.22115	0	0	0
11	1	0	1	0.06655	1	-0.18388	1	-0.2732	1	-0.43107	1	-0.41349	0.96232	-0.51874	0.90711	-0.59017	0.8923	-0.66474	0.69876	-0.70997	0.70645	-0.7632	0.63081
12	1	0	1	-0.5421	1	-1	1	-1	1	0.36217	1	-0.41119	1	1	1	-1	1	-1	1	1	1	1	1
13	1	0	1	-0.16316	1	-0.10169	0.99999	-0.15197	1	-0.19277	0.94055	-0.35151	0.95735	-0.29785	0.93719	-0.34412	0.94486	-0.28106	0.90137	-0.43383	0.86043	-0.47308	0.82987
14	1	0	1	-0.86701	1	0.2228	0.85492	-0.30896	1	-0.1209	1	0.35147	1	0.07772	1	-0.14767	1	-1	1	-1	0.61831	0.15803	1
15	1	0	1	0.0738	1	0.0342	1	-0.05563	1	0.08764	1	0.19651	1	0.20328	1	0.12785	1	0.10561	1	0.27087	1	0.44758	1
16	1	0	0.50932	-0.93996	1	0.26708	-0.0352	-1	1	-1	0.43685	-1	0	0	-1	-0.34265	-0.37681	0.03623	1	-1	0	0	0
17	1	0	0.99645	0.06468	1	-0.01236	0.97811	0.02498	0.96112	0.02312	0.99274	0.07808	0.89323	0.10346	0.94212	0.05269	0.88809	0.1112	0.86104	0.08631	0.81633	0.1183	0.83668
18	0	0	0	0	-1	-1	1	1	-1	-1	1	-1	-1	-1	1	-1	-1	-1	-1	-1	-1	-1	-1
19	1	0	0.67065	0.02528	0.66626	0.05031	0.57197	0.18761	0.08776	0.34081	0.63621	0.12131	0.62099	0.14285	0.78637	0.10976	0.58373	0.18151	0.14395	0.41224	0.53888	0.21326	0.5142
20	0	0	1	-1	0	0	0	0	0	1	1	1	-1	-0.71875	1	0	0	-1	1	1	-1	1	1
21	1	0	1	-0.00612	1	-0.09834	1	-0.07649	1	-0.10605	1	-0.11073	1	-0.39489	1	-0.15616	0.92124	-0.31884	0.86473	-0.34534	0.91693	-0.44072	0.9606
22	0	0	1	1	0	0	0	0	-1	-1	-1	0	0	0	0	-1	-1	-1	-1	-1	-1	-1	-1
23	1	0	0.96071	0.07088	1	0.04296	1	0.09313	0.90169	-0.05144	0.89263	0.0258	0.8325	-0.06142	0.87534	0.09831	0.76544	0.0028	0.75206	-0.05295	0.65961	-0.07905	0.64158
24	0	0	-1	1	0	0	0	0	-1	1	1	1	0	0	0	0	0	-1	-1	1	1	1	0
25	1	0	1	-0.06182	1	0.02942	1	-0.05131	1	-0.01707	1	-0.11726	0.84493	-0.05202	0.93392	-0.06598	0.6917	-0.07379	0.65731	-0.20367	0.9491	-0.31558	0.80852
26	1	0	1	0.5782	1	-1	1	-1	-1	1	1	-1	1	-1	1	-1	-1	-1	-1	-1	1	-0.62796	1
27	1	0	1	-0.08714	1	-0.17263	0.86635	-0.81779	0.94817	0.61053	0.95473	-0.41382	0.88486	-0.31736	0.87937	-0.23433	0.81051	-0.6218	0.12245	-1	0.90284	0.11053	0.62357
28	0	0	-1	-1	0	0	-1	1	1	-0.375	0	0	0	0	0	0	1	-1	-1	-1	1	-1	0
29	1	0	1	0.0838	1	0.17387	1	-0.13308	0.98172	0.6452	1	0.47904	1	0.59113	1	0.70758	1	0.82777	1	0.95099	1	1	0.98042

4.2 Algorithmic Approach / Algorithm / DFD / ER diagram/Program Steps



Algorithm of KNN-

- Step-1: Select the number K of the neighbors
- Step-2: Calculate the Euclidean distance of K number of neighbors
- Step-3: Take the K nearest neighbors as per the calculated Euclidean distance.
- Step-4: Among these k neighbors, count the number of the data points in each category.
- Step-5: Assign the new data points to that category for which the number of the neighbor is maximum.
- Step-6: Our model is ready.

Algorithm of Decision Tree-

1. It begins with the original set S as the root node.
2. On each iteration of the algorithm, it iterates through the very unused attribute of the set S and calculates Entropy(H) and Information gain(IG) of this attribute.
3. It then selects the attribute which has the smallest Entropy or Largest Information gain.
4. The set S is then split by the selected attribute to produce a subset of the data.
5. The algorithm continues to recur on each subset, considering only attributes never selected before.

Algorithm of Random Forest-

- Step-1: Select random K data points from the training set.
- Step-2: Build the decision trees associated with the selected data points (Subsets).
- Step-3: Choose the number N for decision trees that you want to build.
- Step-4: Repeat Step 1 & 2.

Algorithm of SVM-

Step 1: SVM algorithm predicts the classes. One of the classes is identified as 1 while the other is identified as -1.

Step 2: As all machine learning algorithms convert the business problem into a mathematical equation involving unknowns. These unknowns are then found by converting the problem into an optimization problem. As optimization problems always aim at maximizing or minimizing something while looking and tweaking for the unknowns, in the case of the SVM classifier, a loss function known as the hinge loss function is used and tweaked to find the maximum margin.

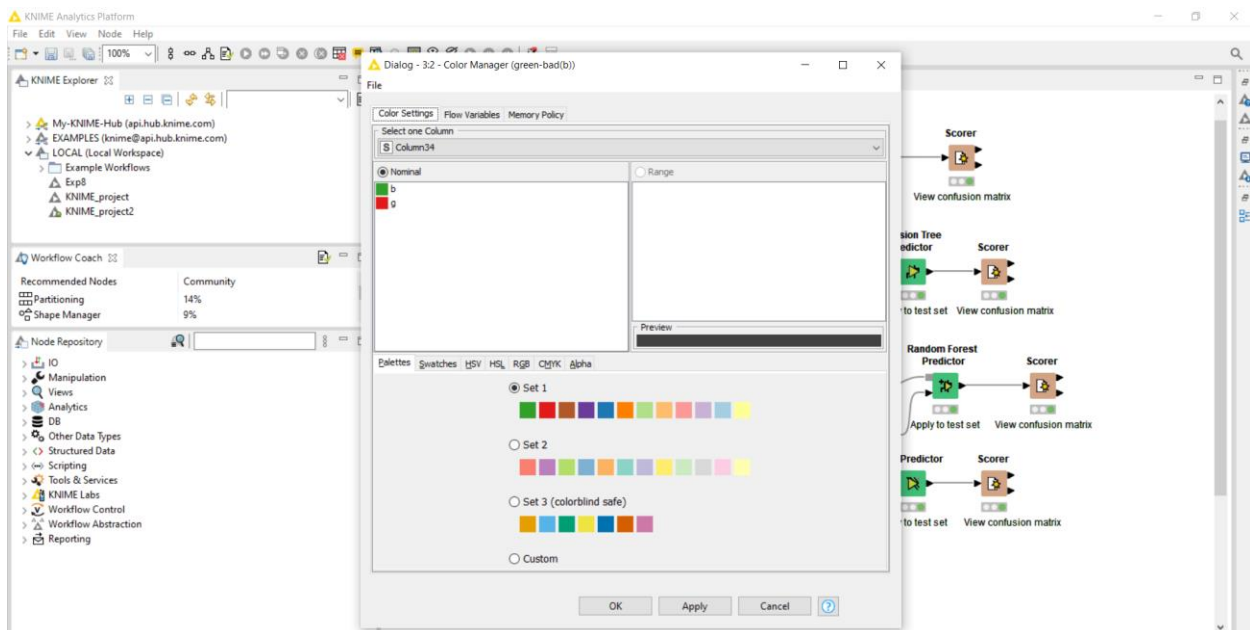
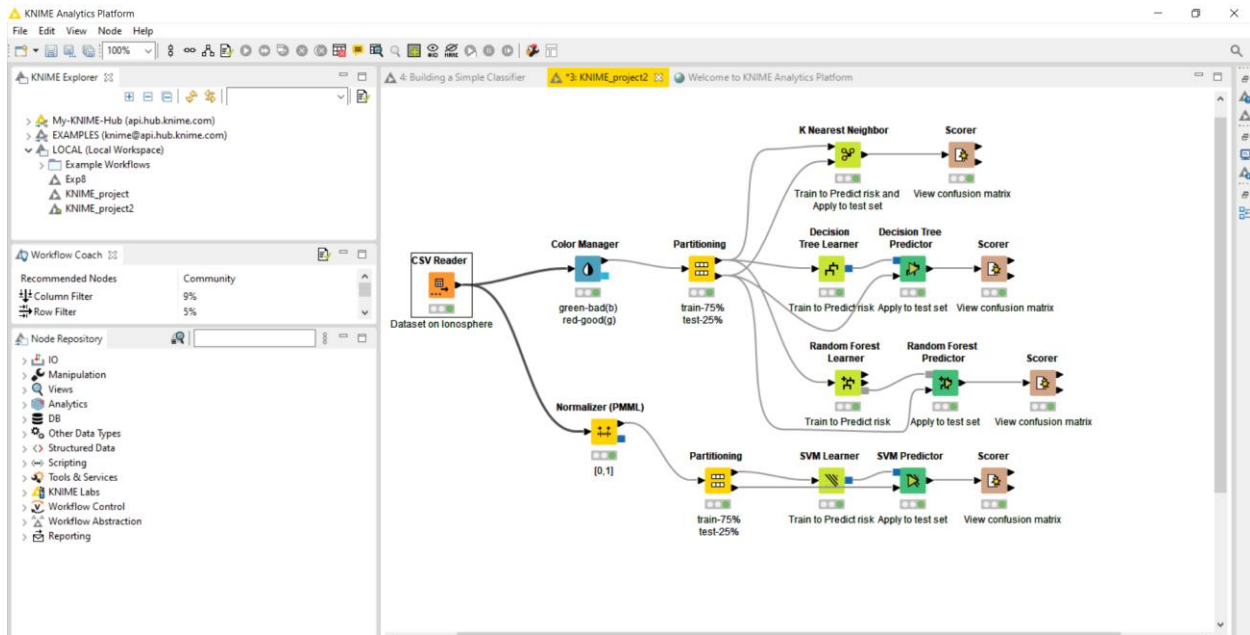
Step 3: For ease of understanding, this loss function can also be called a cost function whose cost is 0 when no class is incorrectly predicted. However, if this is not the case, then error/loss is calculated. The problem with the current scenario is that there is a trade-off between maximizing margin and the loss generated if the margin is maximized to a very large extent. To bring these concepts in theory, a regularization parameter is added.

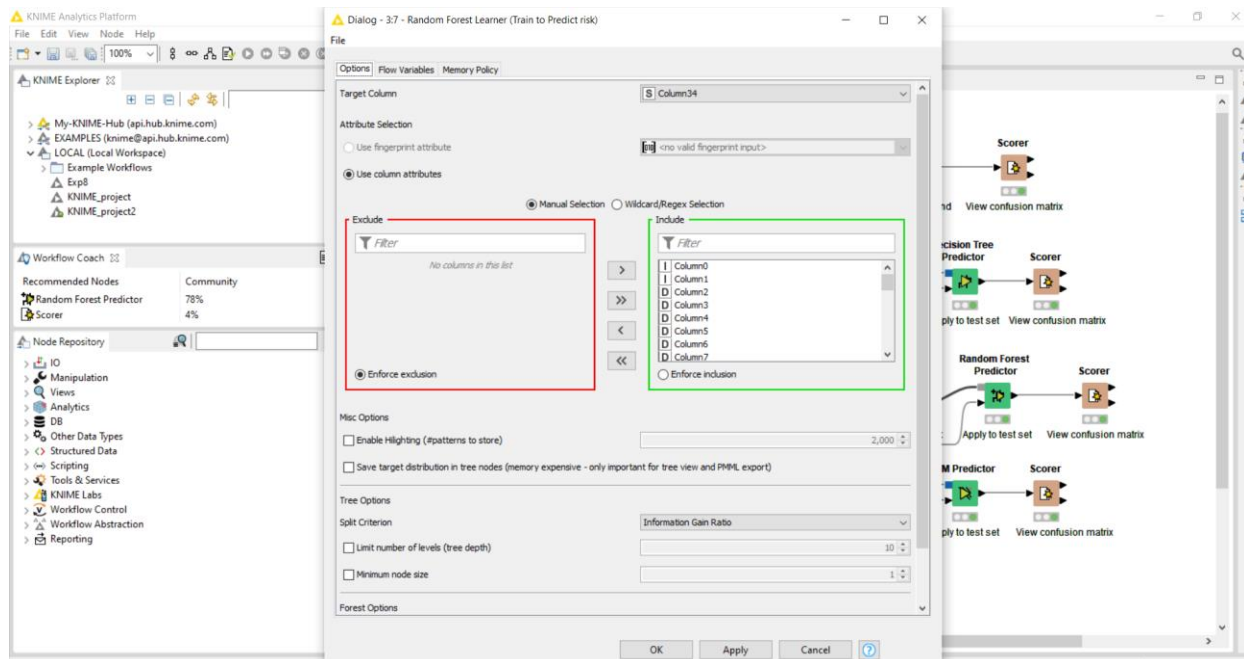
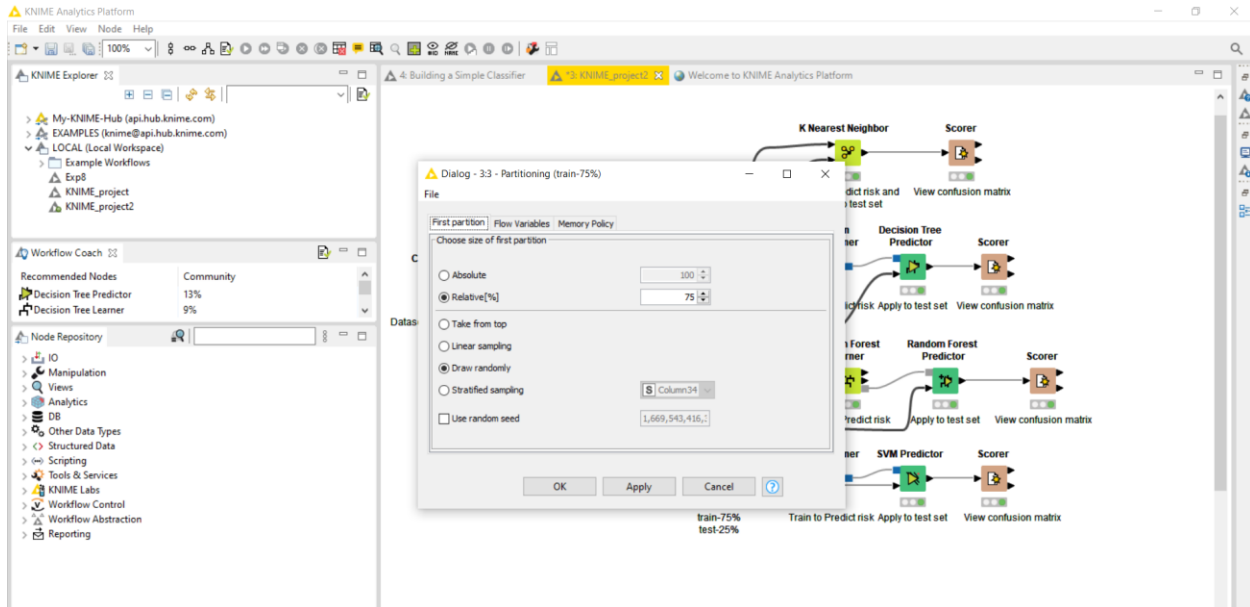
Step 4: As is the case with most optimization problems, weights are optimized by calculating the gradients using advanced mathematical concepts of calculus viz. partial derivatives.

Step 5: The gradients are updated only by using the regularization parameter when there is no error in the classification while the loss function is also used when misclassification happens.

Step 6: The gradients are updated only by using the regularization parameter when there is no error in the classification, while the loss function is also used when misclassification happens.

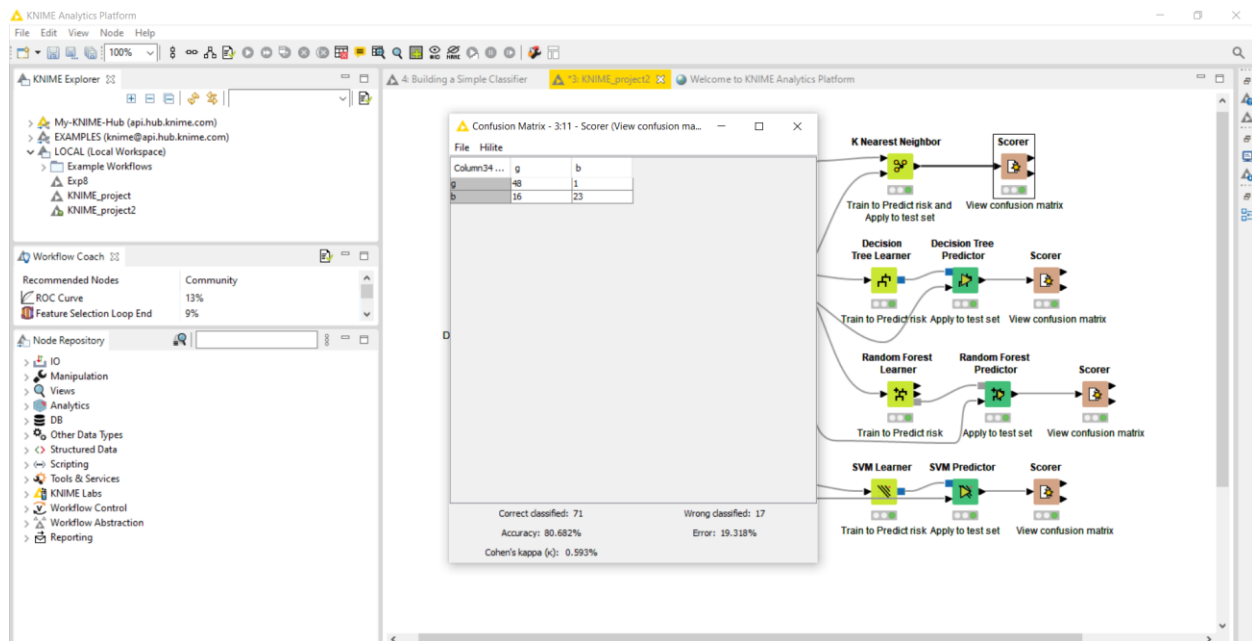
5. Implementation and Testing



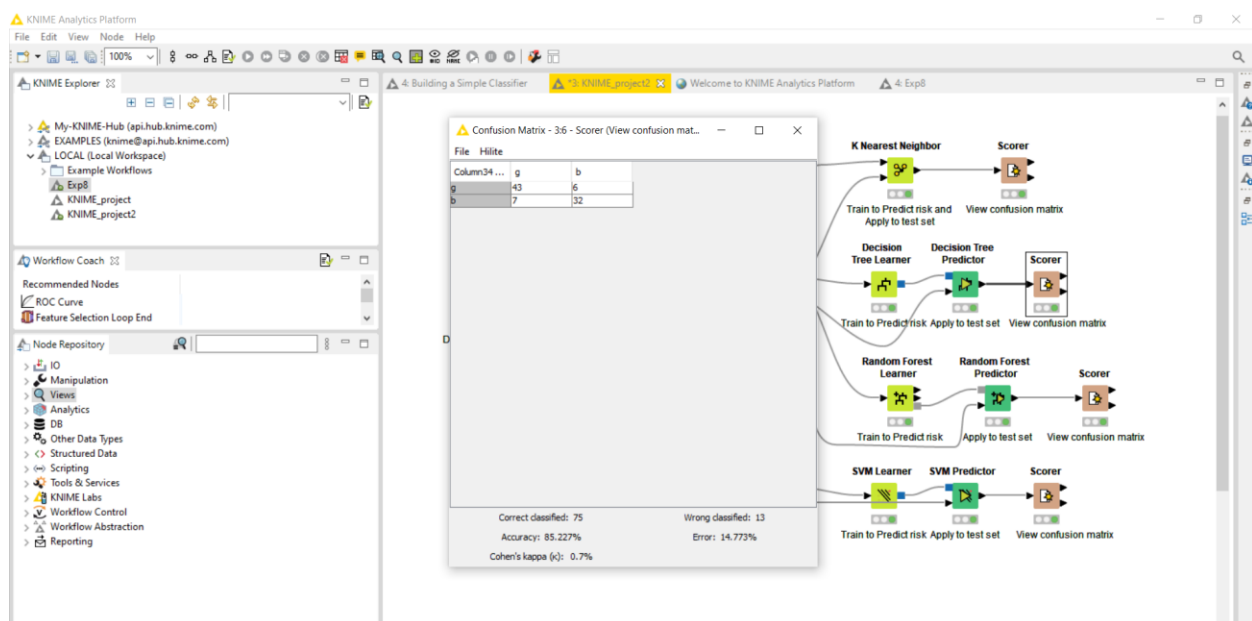


6. Outputs

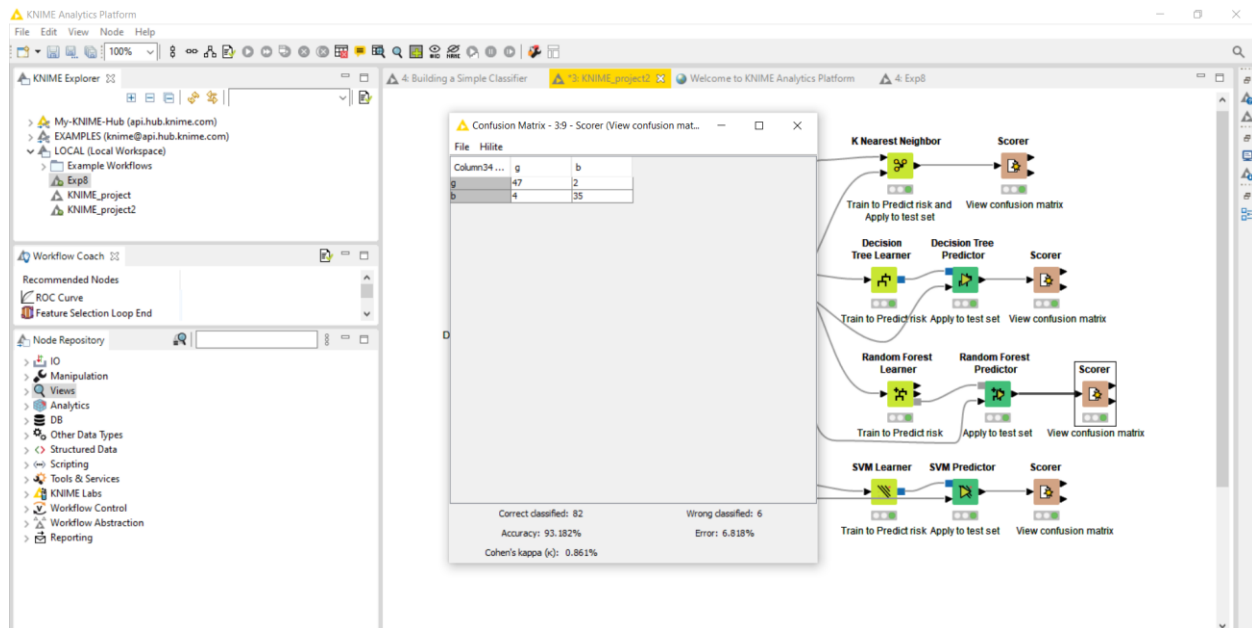
Confusion Matrix of KNN-



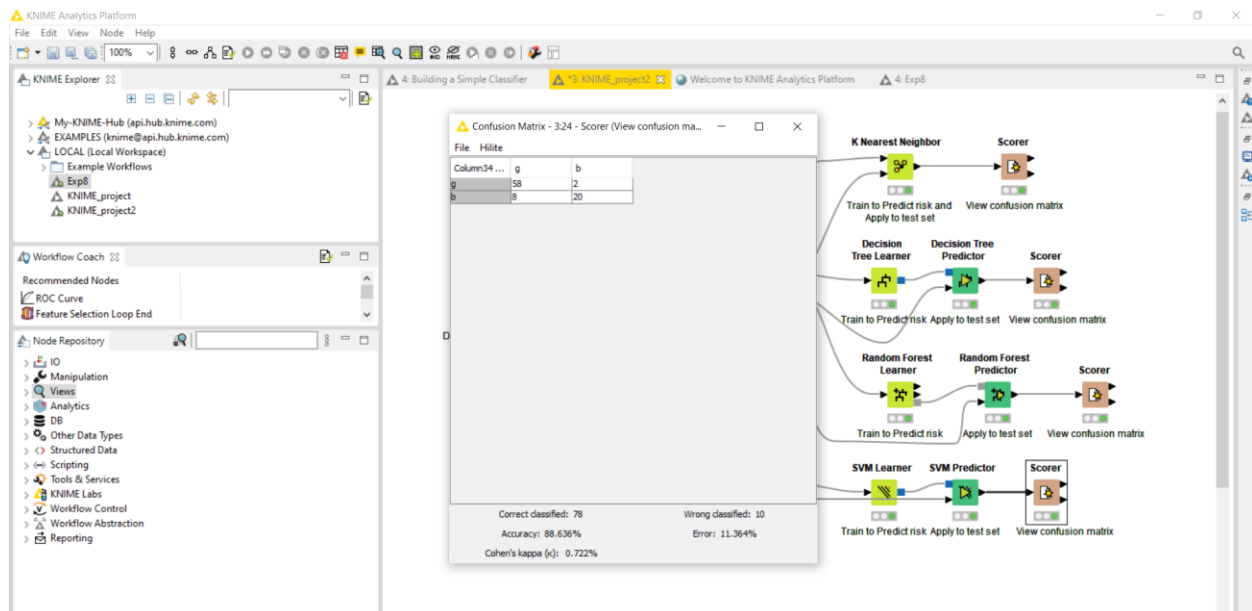
Confusion Matrix of Decision Tree-



Confusion Matrix of Random Forest-



Confusion Matrix of SVM-



7. Conclusion And Future Scope

The study done above has shown that there are multiple algorithms that can be used to delineate good radar returns from bad ones. The performance of Random Forest classification (93% accuracy) is best among all the classifiers which indicates that with advent of newer technologies and better computation power, Random Forest can be used. Since study of radar returns helps to study about ionosphere more, the research can be extended with more data and more sophisticated learning methods which would be of great help in the classification of good returns and bad returns more accurately without any human intervention. More study can be done with newer algorithms that will be developed during the course of time for successful classification of radar returns.