Class Rules

Ganapathy Seshadri C Aiyer
gsa277@nyu.edu
Priya Singh Khokher
pnk230@nyu.edu
Sanjitha Udipi
su472@nyu.edu
Shi Fan
sf2632@nyu.edu

Project page: https://github.com/NYU-CS6313-SPRING2016/Group-9-INET-Class-Rules
Video: https://vimeo.com/167367790
Working demo: https://nyu-cs6313-spring2016.github.io/Group-9-INET-Class-Rules/

What is the problem you want to solve and who has this problem?

Classification rules as an algorithm has become very popular due to rise in the field of Machine Learning. But users consuming this analysis view it more as a black box with relatively opaque ways to see the results. For instance, the process of generating rules for a dataset often results in hundreds of rules made up of numerous attributes, which are not human readable always. We want to visualize these rules so that they can be better understood by the technical audience, so that they can make more sense out of the rules in a cohesive manner.

What are the driving analytical questions you want to be able to answer with your visualization?

For our classification, we obtained 79 rules, with 20 independent attributes and 1 independent attribute on German Credit History data.

1. **What each rule constitutes?**
   We want to see which attributes are incorporated into each classification rule, thus also obtaining the length of each rule, as in the number of attributes in the said rule.

2. **If there is a certain rule which has more data points and is there a rule with really less data points?**
   The inherent attributes in classification rules are support and coverage. Support means the ratio of the number of data points in the rule correctly classified to the number of data points in the rule. While coverage means the ratio of the number of data points in the rule to the total number of data points.

This is what needs to be known for rules to know which rules take into account more data points (meaning they are more generalizable) while which ones take few data points (meaning they are specific to a few combination of attributes and data points)

3. **What are the values of attributes that are in each rule?**
   A dataset consist of categorical and numerical attributes. It is important to be able to read all these values of the attributes that are occurring in the rules. Whether it is their absence/presence/more than one value in the same attribute that determine what the rule is all about.

4. **How attributes are moving between the rules ?**
   If there is some correlation between the rules with respect to attributes, as in A attribute occurs in 4 or 5 different rules, which would also mean that is a similar behaviour in the rules, if that is.

What does your data look like? Where does it come from? What real-world phenomena does it capture?

1. **What does your data look like?**
   Our data is the decision results for German loan applicants based on their personal information, containing 1,000 data points, 20 attributes, and 1 label (the classification result) [1]. In other words, there are 1,000 applicants, and our data shows how the credit risk classification results look for each of them based on 20 attributes, which are the personal information they provide along with the application.

   In our vis design, we attempt to visualize the attributes that each rule entails. For example, our first rule is that if the applicant has no other payment plans and no checking account but has a critical/other existing credit, then they will have a good credit risk, meaning that they are likely to repay the loan. The 3 attributes involved in this particular rule, other payment plans, checking status, and credit history, will be highlighted in the vis design. It works similarly for all the other rules.

2. **Where does it come from?**
   We acquired this dataset directly from the sample datasets provided by Weka software. Weka is an open-source machine learning software that is able to implement rule-based classification with an easy-to-use user interface. The dataset had already been preprocessed as we received it from the software.

3. **What real-world phenomena does it capture?**
   Almost all the features were derived, except accuracy, coverage, and length.

Attribute Name	Attribute Type	Real-world Entity	Value Description	Derived
checking_status	categorical	How much amount of checking the loan applicant has on the bank account	<0, 0<=X<200, >=200, no checking	no
duration	numerical	How many months the loan applicant have had the checking account	4-72	no
credit_history	categorical	How the credit history looks for the loan applicant	critical/other existing credit, existing paid, delayed previously, no credits/all paid, all paid	no
purpose	categorical	What the loan is used for	radio/tv, education, furniture/equipment, new car, used car, business, domestic appliance, repairs, other, retraining	no
credit_amount	numerical	How much credit the loan applicant has on the account	250-18424	no
savings_status	categorical	How much amount of savings the loan applicant has	<100, 100<=X<500, 500<X=<1000, >=1000, no known savings	no
employment	categorical	How many years the loan applicant has been working	<1, 1<=X<4, 4<=X<7, >=7, unemployed	no
installment_commitment	numerical	Installment rate in percentage of disposable income	1, 2, 3, 4	no
personal_status	categorical	Gender and marriage situation	male single, female div/dep/mar, male	no

		for each loan applicant	div/sep, male mar/wid	
other_parties	categorical	What other parties are involved in the application	co applicant, guarantor, none	no
residence_since	numerical	Present residence since	1, 2, 3, 4	no
property_magnitude	categorical	The biggest type of property (in value) the loan applicant has	real estate, life insurance, car, no known property	no
age	numerical	The age of the applicant	19-75	no
other_payment_plans	categorical	What other types of payment plans the loan applicant has	none, bank, stores	no
housing	categorical	What type of housing the loan applicant has	own, rent, for free	no
existing_credits	numerical	Number of existing credits at this bank	1, 2, 3, 4	no
job	categorical	What type of job the loan applicant has	skilled, high qualif/self emp/mgmt, unskilled resident, unemp/unskilled non res	no
num_dependents	numerical	Number of people liable to provide maintenance for	1, 2	no
own_telephone	bool	Whether the loan applicant owns telephone or not	yes, none	no
foreign_worker	bool	Whether the loan applicant is a foreign worker or not	yes, no	no

accuracy	numerical	Percentage of people that satisfy both the condition and the result of a rule	69-100	yes
coverage	numerical	Percentage of people that satisfy the condition of a rule	0.3-13.7	yes
length	numerical	The number of attributes involved in a rule	0-7	yes
***class**	bool	The classification result for the loan applicant	good, bad	

* This is the dependent variable. Technically, it is not an attribute in our data.

What have others done to solve this or related problems?

There has been a fair amount of research, both academic and non-academic, in the field of classification rules visualization. All the research we did was to increase the consideration space for our vis design, so that we could find the best suited vis in this particular context.

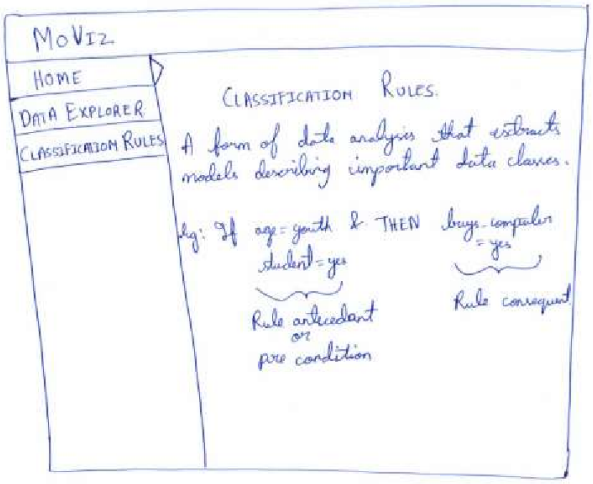
Some of the work we referred to apart from the suggestions of out professor are:

1. Shiny Apps
   This is a UI giving a clear understanding of rule interaction, rule listing and scatterplot for support and confidence.

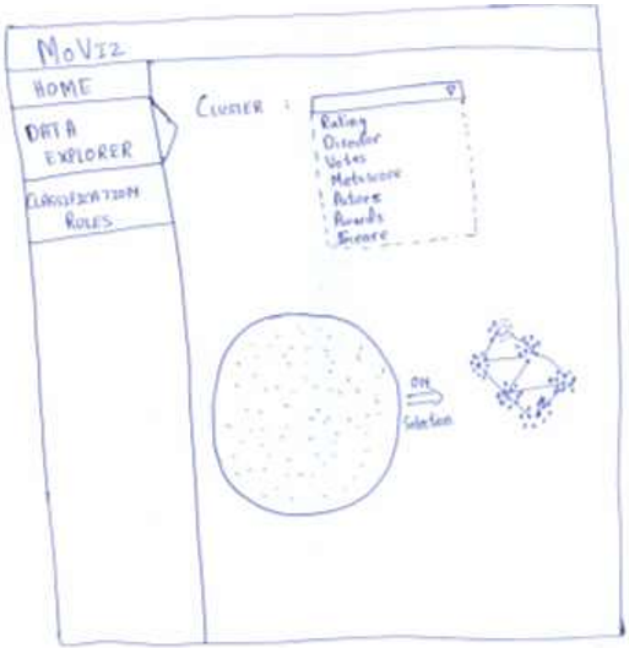
2. Association rules in R
   This is a step further into classification and particular to association rules (not very useful for our visualization) but it exposed to different set of options, like matrix and graph based visualization
3. Rule Based Classifiers Interaction
   This was not a very minimalist way of visualizing rules. They used Chord Diagram to visualize the rules. It helped us in knowing which approach should not be taken for our purpose.

Initial Mockup

This is the first page of the web application, which has description about classification rules. This page will also include the details of the data source.

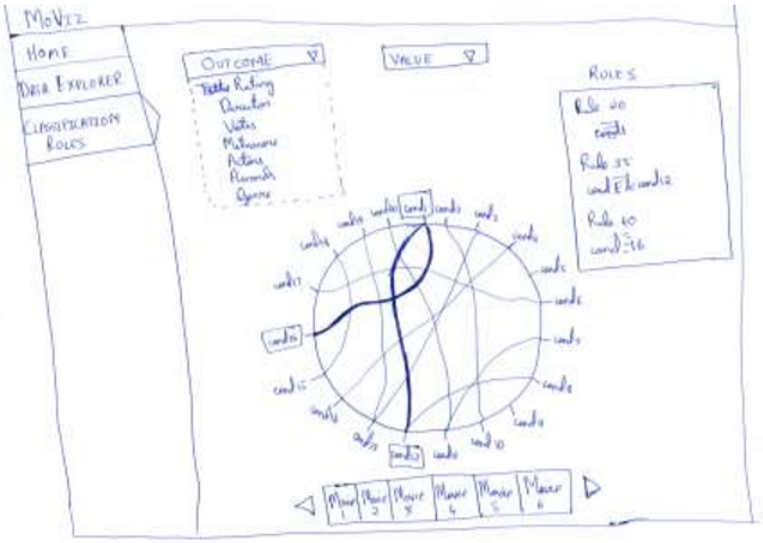


This page will help the user to explore the dataset being used. The user will be able cluster the data points by choosing an attribute. This page has no relation with the visualisation of class rules, it just helps the user understand the data in a better way.



This page allows user to choose outcome attribute for which the rules were generated. The user is also allowed to choose values for that particular outcome attribute. This page has 3 views.

The first one is the circle with all the conditions in a rule, the second view is the list of rules and the third view is the slider present at the bottom which shows the poster of movies. All the views



are linked and the objects displayed in these views solely depends on the outcome attribute and the value chosen. While hovering over the connection between two conditions in the circle, the specific rules will get displayed in view 2 and the posters of movies related to the link will be shown in view 3.When a rule is chosen in the list the corresponding connections will get highlighted in the circle and the movie posters will be shown in view 3. The attributes of rules generated will be shown along with the rules in view 2. There is one rule network for each possible outcome. The width and color of connections between two conditions in the circle will also depend according to the following relation [2]:
This particular visualization was chosen over others because it helps us to represent the class rules in an efficient way using networks as the encoding channel. This helps the user to view all the class rules generated in efficient manner. A visual graphical representation of each outcome would help us analyze the relative importance of the conditions and the corresponding rules in that outcome. Also the user is able to view the data sets that generated that particular rule in the form of movie posters. The views designed in the UI are arranged in such a way that the user can see the effect of interaction on different views simultaneously.

Mistakes in the approach:

1. More centered on visualizing the data than rules
2. Not very minimalist but over complicated in reading the rules
3. Not question-centered but design centered. We were focussing more on which visualization to use than the questions needed to be answered with it. This was partly because of the selective reading of research papers

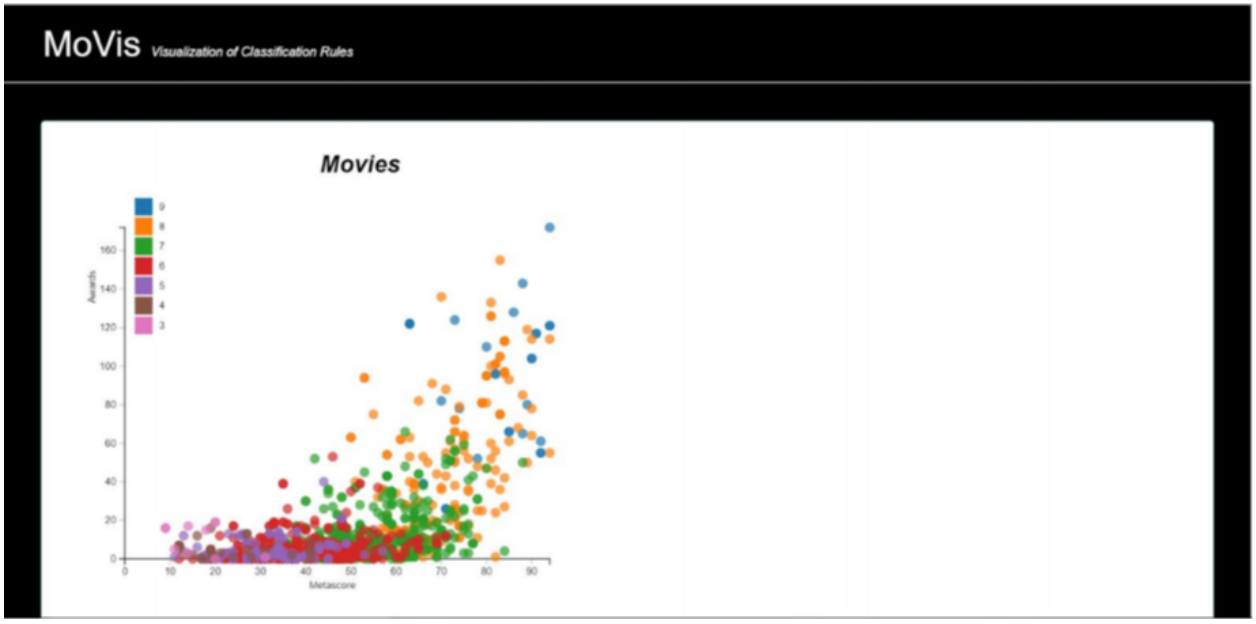
Ways we improved:

1. Developed a solid framework of questions to be answered.
2. Understood the data and the algorithm better and what kind of attributes it would provide us with
3. Tried different set of classification rules to better understand the problems of many rules being produced

Project Update

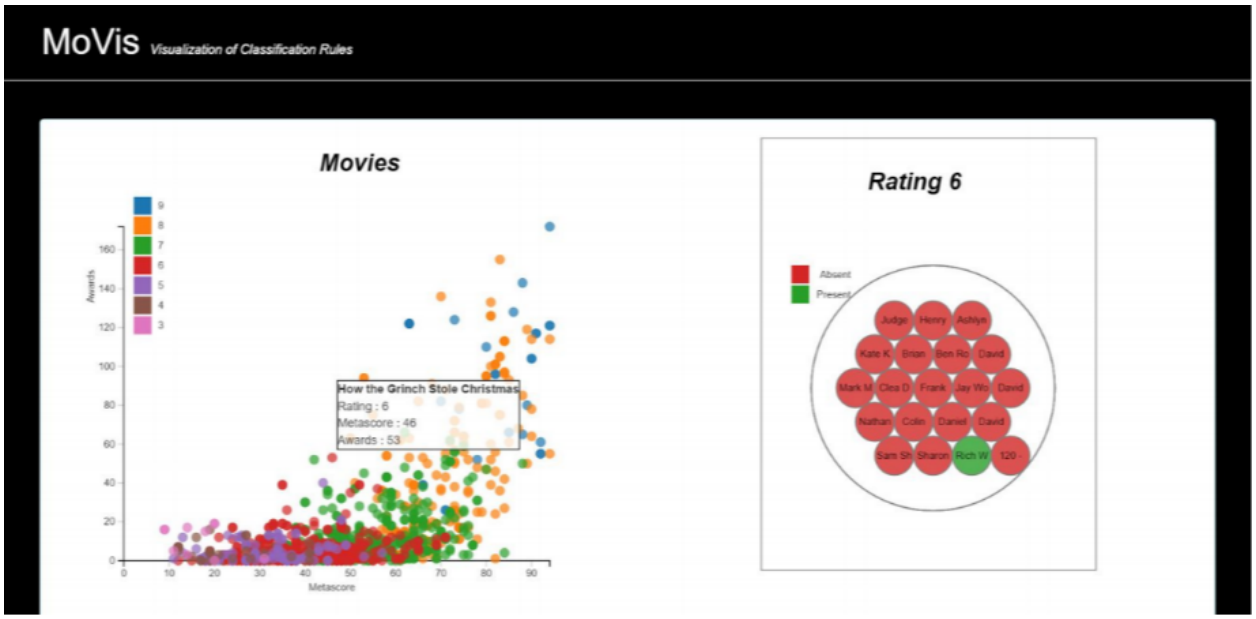
Initially we were using Movie Dataset, that we had collected by the OMDb API, cleaned and analyzed on our own. It took us a significant amount of time to analyze the data than to visualize it. The classification produced rules that were used for the visualization in the project update.

The following was the Project Update with its due intention:



The primary motivation of PCA is to visualize high dimensional data into two or three dimensions. Based on features which are highly variant and hence are prominent for the data. Those two features were Awards and Metascore for our data. The scatterplot shows them on the X and Y axis respectively and these points are color encoded according to the Rating, which is our ‘Y’, the variable for which we are running the classification.

VIEW1:



This view provides a hover over the point (each data point is a movie hence its title is shown) to give the name and the values of Awards, Metascore and Rating.
And in which rule does it fall, along which attributes in the same rule is also shown.

Mistakes:

- On our Professor's suggestion we were supposed to perform PCA on the rules, but PCA on the data points were performed as the analysis required to do the former were not well known to us.

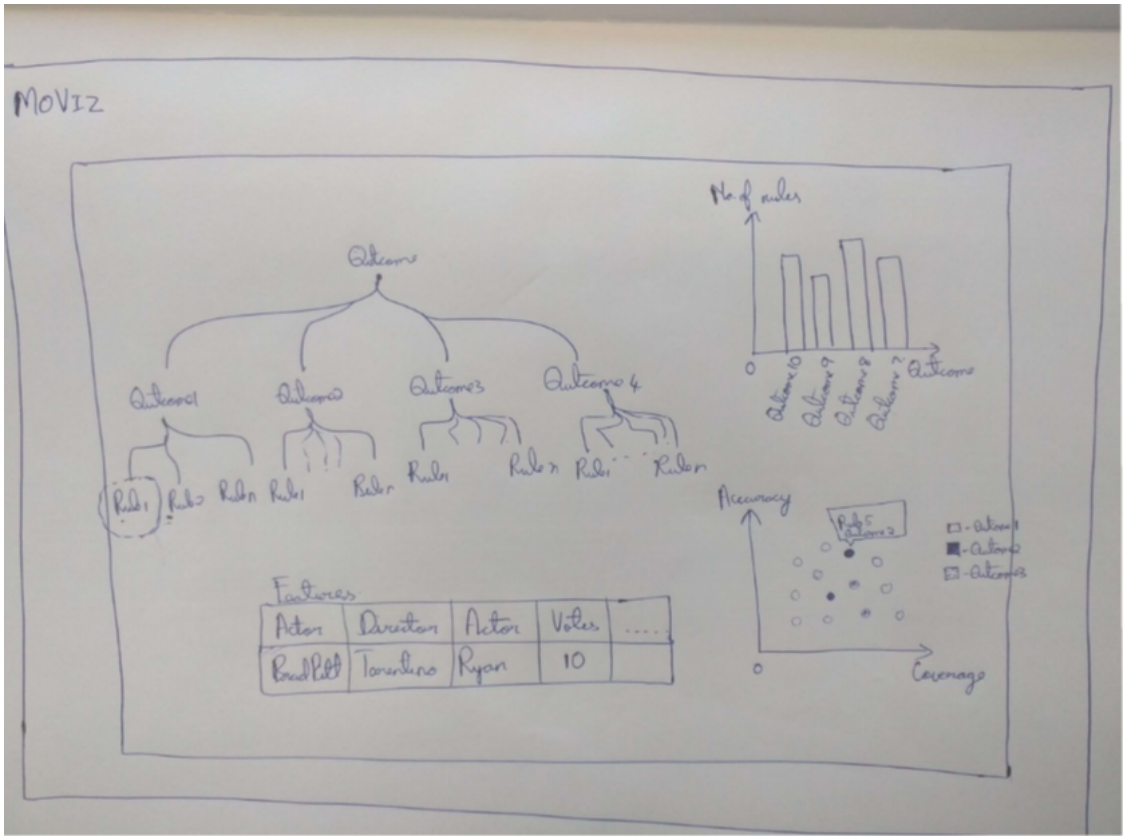
Way for improvement:

- Our Professor assigned all of us to submit individual designs for selecting the most apt design later.

The submitted designs:

Design 1

Dashboard to visualize class rules



View 1: A plot to show number of rules per outcome.
Graph: Bar graph

View 2: A plot to show rules on the basis of coverage vs accuracy
Graph: Scatterplot. Each point will be color coded according to outcome.

View 3: A tree showing various outcomes and rules belonging to that particular outcome

View 4: Display features and values of the chosen rule. If a feature is present then its font is green in color else it is in red color.

Interaction:

1. On click of a particular bar (outcome) in view 1, the bar gets highlighted and view 2 shows rules that are related to the outcome in the scatterplot and the tree will show all the rules related to that outcome.
2. On click of a particular rule in the scatterplot the location of the rule in tree will be highlighted. In the same time view 4 will be populated with attributes of the rule.
3. If we hover over a rule in view 2 we display the rule number and the outcome as a tool tip.

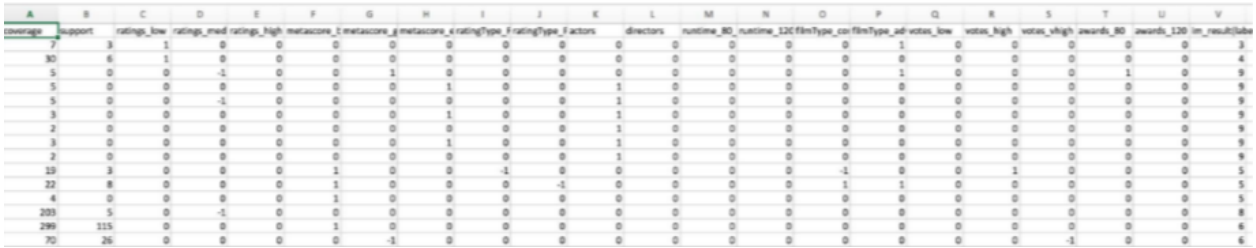
Design 2

Data

The data for classification rules can be extracted in the following way, using the JRIP rule as an example.

Containing features including coverage, support, ratings, metascores, actors, directors, runtime, film type, votes, and awards, the data label is the result of IMDb rating in this particular case. Except coverage and support, all the other features are dummy-encoded (i.e. ratings ⇒ ratings_low, ratings_high, ratings_vhigh).

1 stands for such feature exists in the rule and it's marked present; -1 stands for such feature exists in the rule and it's marked absent; 0 stands for such feature doesn't exist in the rule. It should be noted that as long as there exist conditions related to actors/directors in the rule, the actors/directors columns are marked as 1/-1. The reason is that there are way too many actors/directors, and dummy-encoding all the names will significantly increase our number of features, which is rather undesirable. Besides that, all the other dummy-encoded features are quite straightforward.

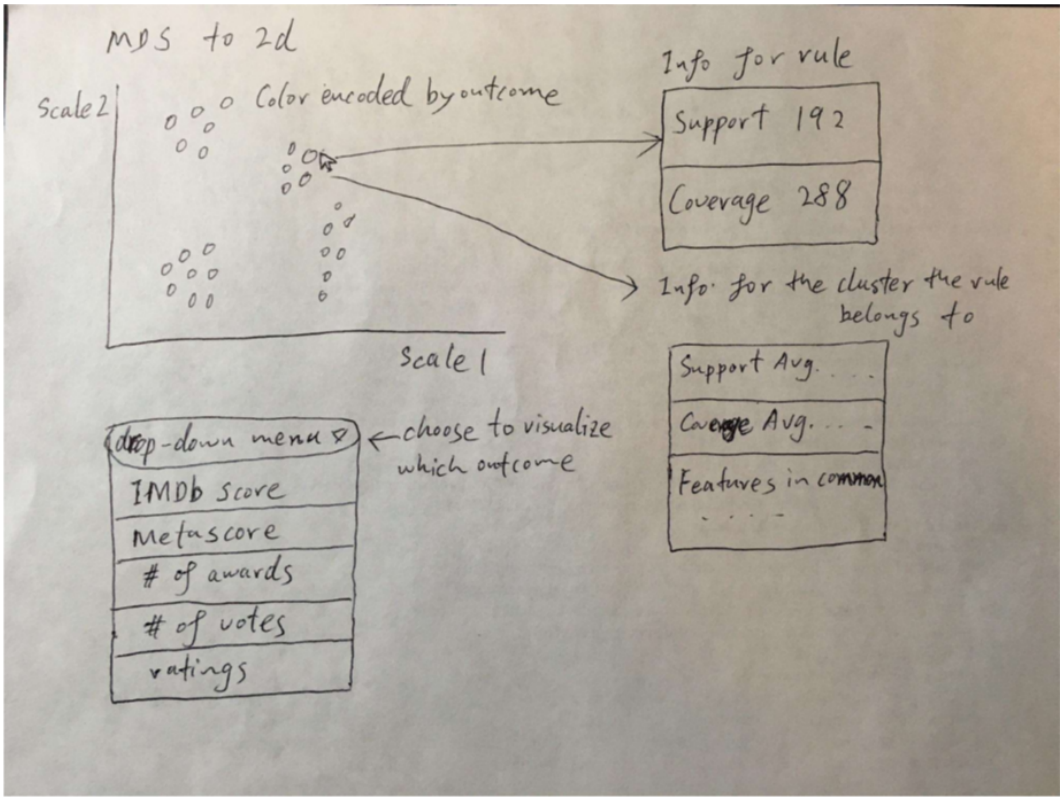


Let's take one step further. This way of encoding rules only applies to one particular outcome, and if we want to visualize classification rules for multiple outcomes, say for instance, metascore, then we'll have to either reconstruct the set of features (because metascore is currently within the feature space for each rule under outcome=imstr) or delete all the outcomes from our feature space (but then we'll lose information on the rules). It'll be a quite tricky tradeoff to make, and we need to be careful.

Viz

We can visualize our 20-dimensional data using MDS, which will reduce our number of dimensions to 2. We expect to see certain clustering patterns for the rules we have. In the UI, the main viz is shown on the left hand side, each scatter point stands for each rule in the 2-d MDS representation. At the bottom, user can select which outcome to visualize using the dropdown menu, and the MDS plot will get updated automatically. As user hovers over each rule in the scatterplot, on the right hand side detailed information of each rule will be displayed (i.e. support, coverage). In addition to the detailed information of each rule, the detailed information of each cluster that the rule belongs to will also be displayed at bottom right (i.e. support average, coverage average, features in common...).

There may be some additional features to add, such as relation of each rule to the data points that we have.



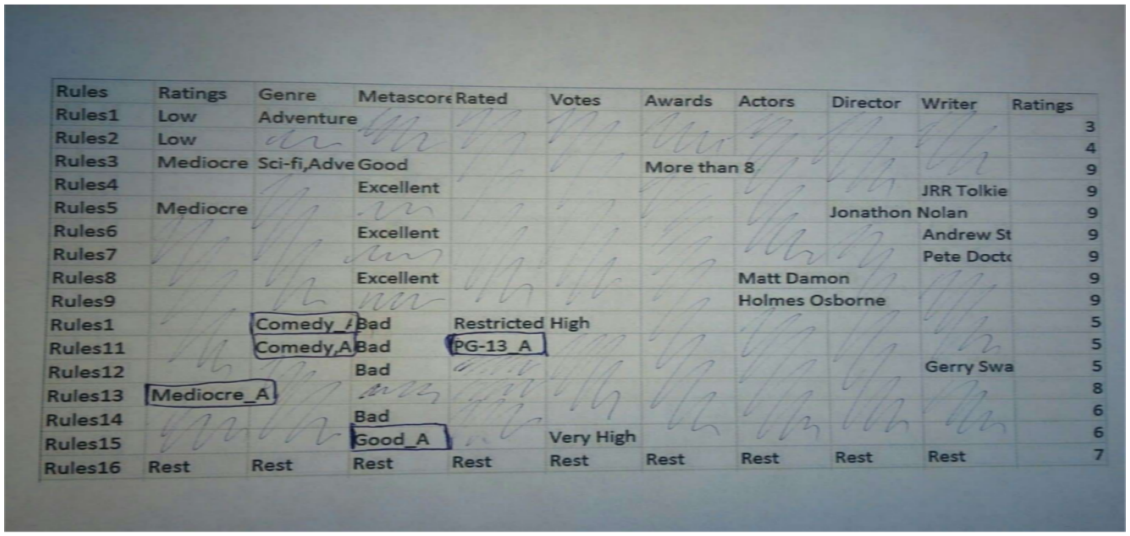
Design 3

Matrix for Info Vis:

Rule has that value, its attribute is mentioned and for the rest it's blank. On hover the coverage and accuracy would be shown.

Design 4

This visualization is more space saving in my opinion as shown below



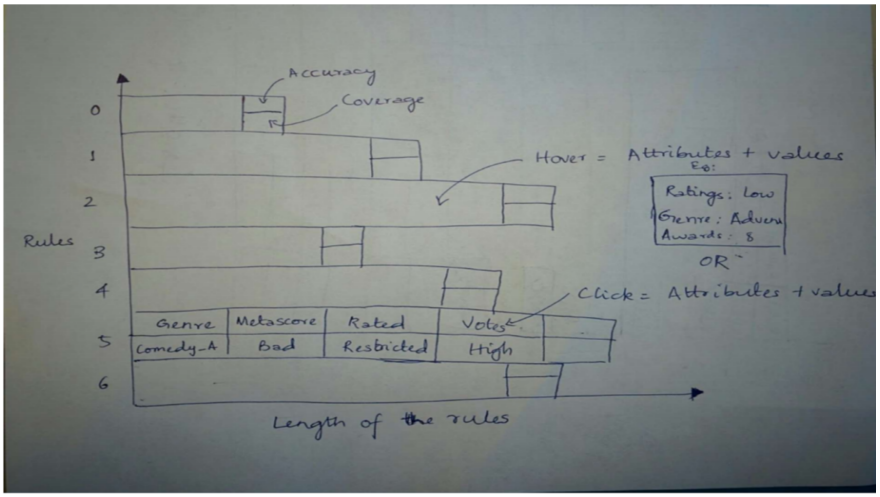
Rules	Ratings	Genre	Metascore	Rated	Votes	Awards	Actors	Director	Writer	Ratings
Rules1	Low	Adventure								3
Rules2	Low									4
Rules3	Mediocre	Sci-fi,Adve	Good			More than 8				9
Rules4			Excellent						JRR Tolkie	9
Rules5	Mediocre							Jonathon	Nolan	9
Rules6			Excellent						Andrew St	9
Rules7									Pete Doct	9
Rules8			Excellent				Matt Damon			9
Rules9							Holmes	Osborne		9
Rules1		Comedy_A	Bad	Restricted	High					5
Rules11		Comedy,A	Bad	PG-13_A						5
Rules12			Bad						Gerry Swa	5
Rules13	Mediocre_A									8
Rules14			Bad							6
Rules15			Good_A		Very High					6
Rules16	Rest	Rest	Rest	Rest	Rest	Rest	Rest	Rest	Rest	7

The X-axis is for the length of the rule, and the Y axis is the rule number - so that we know how many rules are there.

On hover the attributes will show - that will be color encoded in the sense that if the rule is (Low_ratings = p) and (Adventure = p) => imstr=three (7.0/3.0)

Then the attributes that are present will be green color and in the case of another rule (Good_Metascore = a) and (Very_high_votes = a) => imstr=six (70.0/26.0)

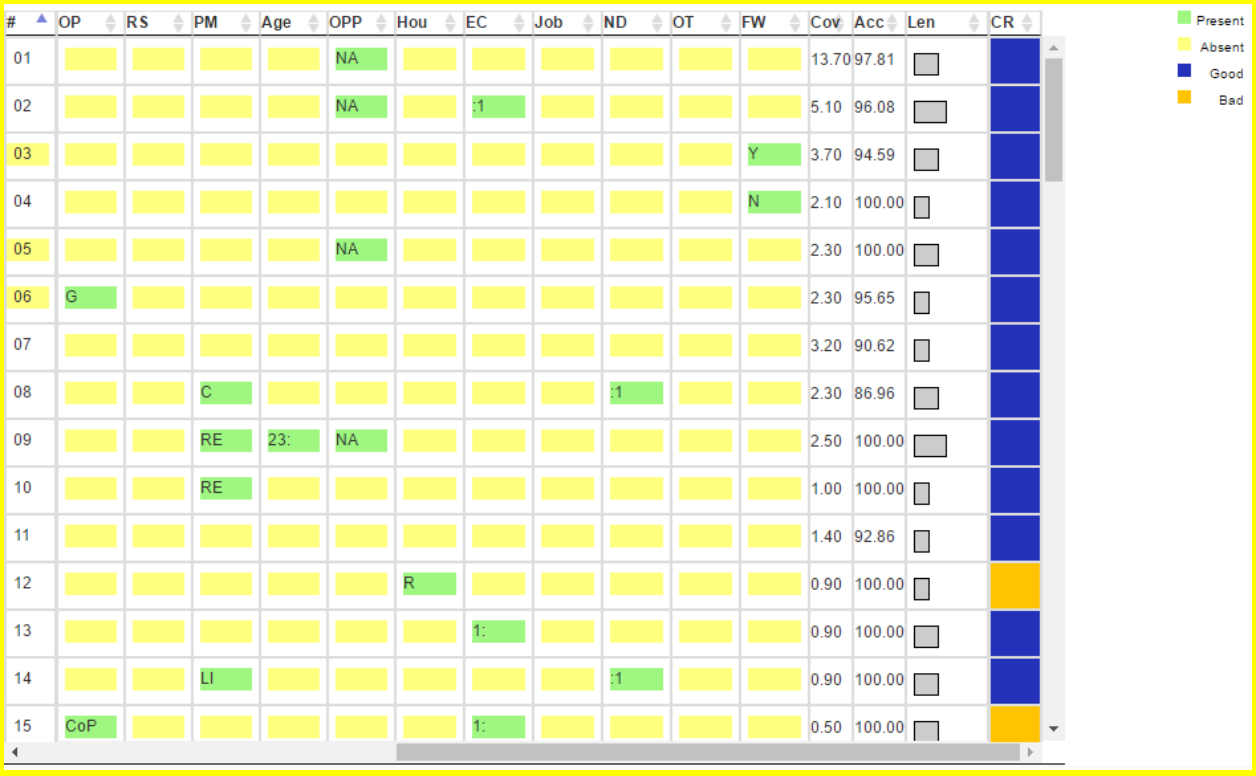
The color in the bar on hover for that rule will be red.



A merger of design 3 and 4 was suggested on the same dataset on Professor's suggestion.

Final Visualization

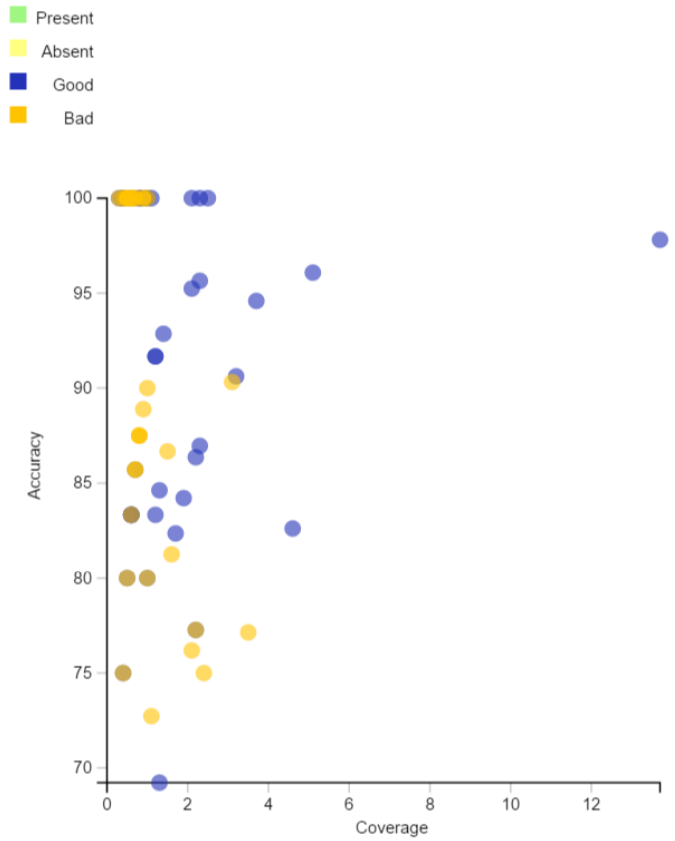
VIEW 1:



#	OP	RS	PM	Age	OPP	Hou	EC	Job	ND	OT	FW	Cov	Acc	Len	CR
01					NA							13.70	97.81		
02					NA		:1					5.10	96.08		
03											Y	3.70	94.59		
04											N	2.10	100.00		
05					NA							2.30	100.00		
06	G											2.30	95.65		
07												3.20	90.62		
08			C						:1			2.30	86.96		
09			RE	23:	NA							2.50	100.00		
10			RE									1.00	100.00		
11												1.40	92.86		
12						R						0.90	100.00		
13							1:					0.90	100.00		
14			LI						:1			0.90	100.00		
15	CoP						1:					0.50	100.00		

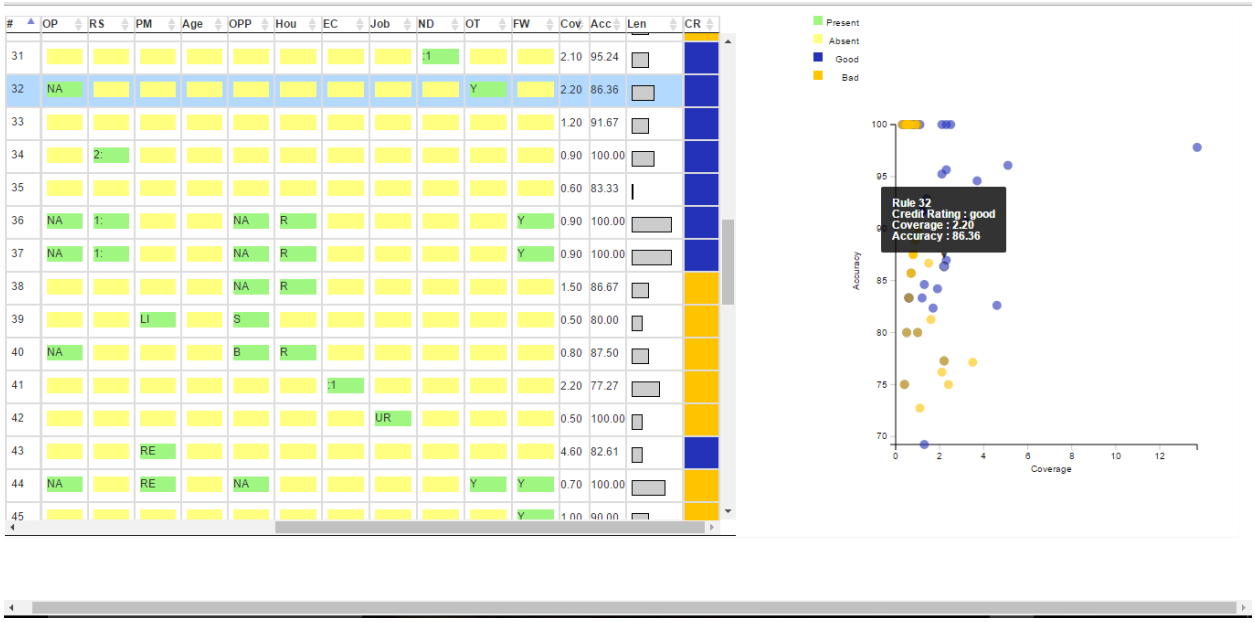
- The view is a matrix table for the rules.All the rules are enlisted, showing the value for each attribute that is present in the rule.
- It is left-right scrollable as our feature space is huge (21 attributes in total including the outcome variable + support+ coverage + length channel (horizontal bar) used to encode the length of the rule).
- It is also up-down scrollable as there are 79 rules. So users can view all values for all rules if they wish to.
- The outcome variable (Credit Rating is color encoded in two color - orange and blue showing bad and good credit rating respectively).

VIEW 2:



- The figure depicts a scatterplot where each point is a rule. The X and Y are the coverage and accuracy for each rule.
- This is plot to show which rules have similar values of each of them.
- In effect, it shows 3 attributes, as the credit rating for each rule is color encoded as orange and blue.

INTERACTION:



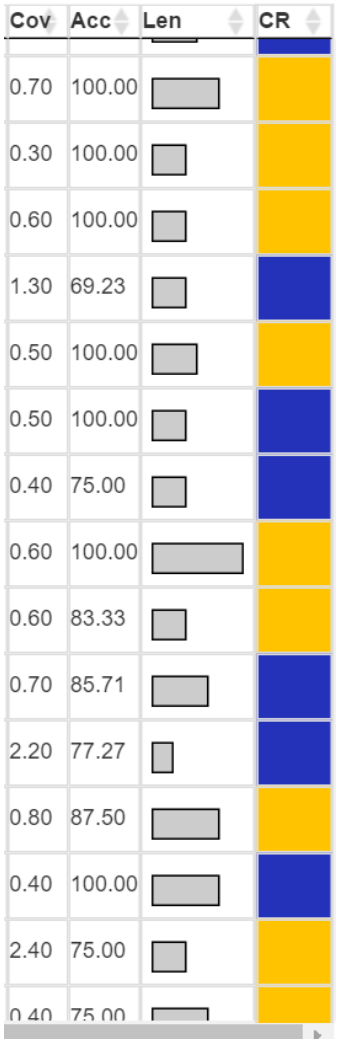
These two views are in turn interlinked to each other in both directions.
If a rule is selected, corresponding point on the scatterplot is highlighted by means of a blob of a larger radius than the point to enhance highlighting.

On the other hand, if a point is selected on the scatterplot, corresponding rule is highlighted in the matrix table.
Hence, when a point in the scatterplot is selected, it also shows by means of a label popup the Rule number, the outcome variable (in our case Credit Rating) the X and Y values - here the coverage and accuracy respectively.

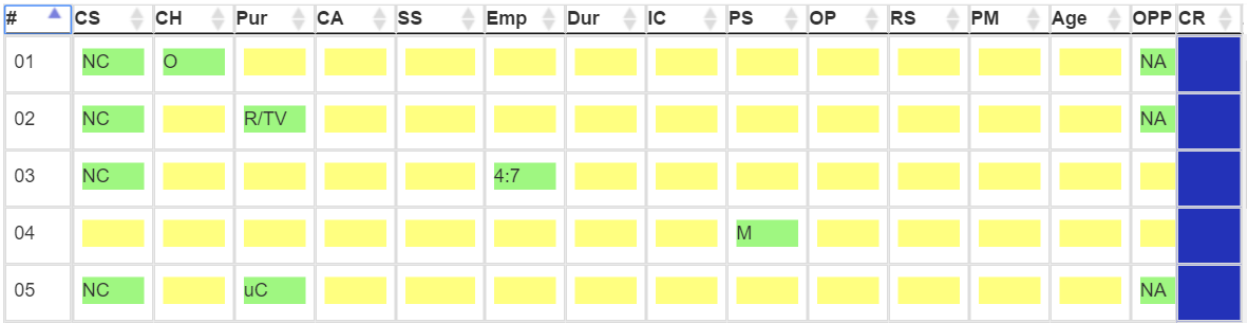
Data Analysis

Question:
Overall the visualization provides better readability of the rules, as anyone can view it better in a matrix than in a form of output text given by a computer program.

1. **Which rule contains more or less attributes? Can this be shown easily without a user having to go through the rules,and counting the attributes in the rule, in a fast and efficient way?**
   Yes, we accomplished that for each rule by adding ‘Length attribute’ in the matrix table, and instead of adding the the number itself, i.e. the length as it is, we encoded it in the length channel by a horizontal bar. The longer the bar, the larger the length. This is useful as writing the length, is not the best way to go through the list, to answer the question fast, as there are different range of values (can be 1, 2, 3… Many values) that are taken care of horizontal bar.



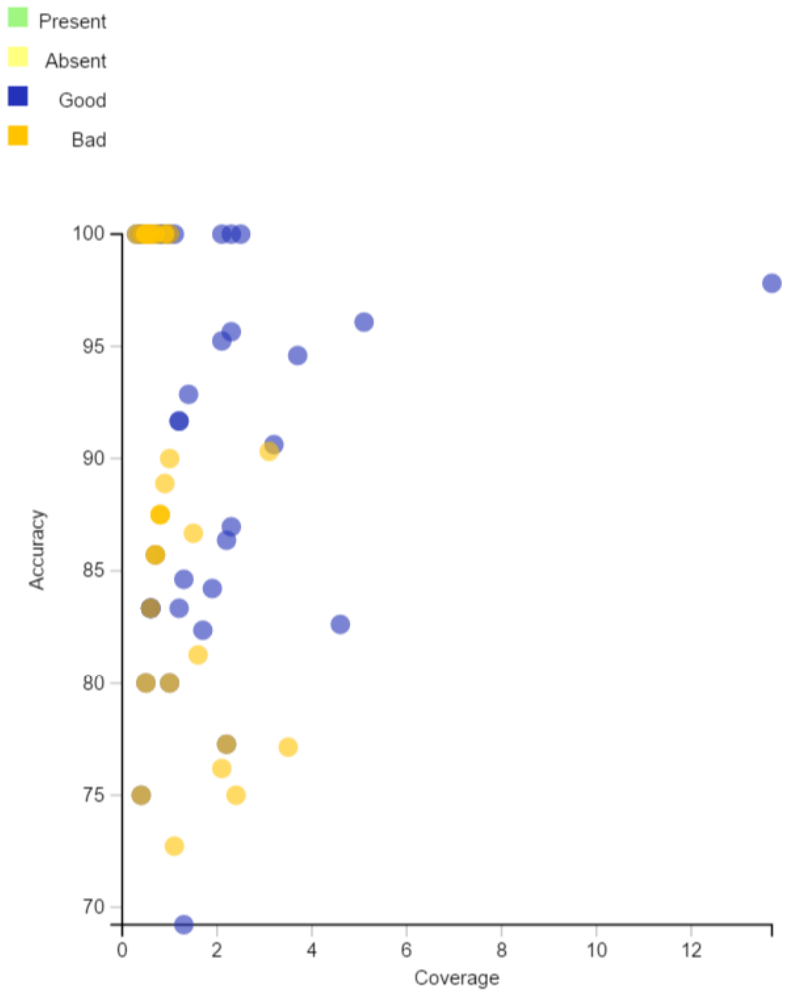
2. **Can we see which rule has which attribute in it, and which are absent? Both at once? So that we know how much of the feature space was covered by that rule?**
   We tried to accomplish that by making the matrix table shown below(in the Final Analysis Part). Here the features are the columns of the table, and rows are indexed by the rule number. A rule having that attribute in it, is shown with its abbreviations and encoded as green color, while the one that is absent is encoded with yellow color. This is done for all the rules.



#	CS	CH	Pur	CA	SS	Emp	Dur	IC	PS	OP	RS	PM	Age	OPP	CR
01	NC	O												NA	
02	NC		R/TV											NA	
03	NC					4:7									
04									M						
05	NC		uC											NA	

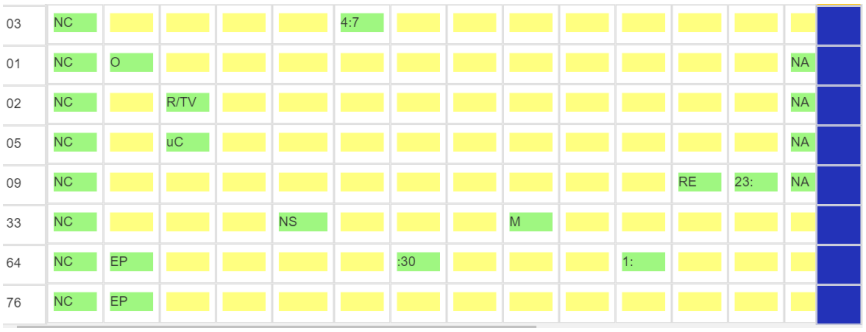
   E.g.: Here Rule 1 has Checking Status,Credit history values that determine the rule (If Checking Status = no checking and Credit history =critical/other existing credit then its Rule 1), but no condition on Purpose or Credit Amount, it is not shown having any value, i.e. is empty.

3. **Can rules be visualized according to support and coverage to easily start going into the analysis rules? What other attributes can be shown for each rule to distinguish them?**
   This is the purpose of scatter plot in our vis. While considering a machine learning problem, (we have run a classification algorithm called PART), it is important to take the output variables we work with and the ones given by the algorithm in return to us after running it. Coverage and Accuracy are the latter while credit rating is the former. We have brought the rules, together based on these features also, (apart from input features given in the table), by plotting each point as a rule, plotting the coverage and accuracy for each of them on the X and Y axis. We added another variable, via color encoding - that is credit rating which is good (blue) or bad (orange). Thus covering all output variables.



4. **For data analysis, can a user, go back between the two views, i.e a visualization based mapping from the input variables space to output variable space?**
   Using our vis, even that is possible, as the two views of matrix table (comprising mostly of input variable space) and scatterplot (output variable space) are interlinked. It is understandable that a user, will want to see the rule in one view, and want to go back and see how it looks like in the other view. For instance, the matrix table primarily consists of input variables, while the scatterplot shows mainly the output variables.The user can click on the rule (point in the scatterplot) and see what that rule is, what input attributes constitute it, and vice versa, the user can hover on the rule, and check where it falls in the scatterplot **relative to all the other rules**. The table consists of values for support and coverage, and the scatterplot will map all the rules into the same space. Thereby letting the user explore other rules, with similar support and coverage values (near by points to the point they selected). Thus the interlinking provides easy back and forth process for the process.

5. **Can a user know for every attribute, in which rule it occurs?**
   This has been accomplished by means of making the attribute column sortable. As in by clicking on the column header - here on checking status - you move from first picture to the other, where the second picture shows the rules with checking status values present in the beginning.
   Hence an analysis on a per-attribute basis is accomplished.



Limitations and Future Works

1. **Linking data with the rules**
   Here just the rules, that are generated by the classification algorithm - PART but the data underlying those rules are not connected. More so there can be same data set of data that could be interlinked between two rules.

2. **Selecting multiple rules/attributes**
   Here we could select the attribute and see which rules have that attribute present, but we can not do that for a combination of attributes, in the sense—select checking status and credit history both and see which rule has the combination of the two present.

3. **Checking the values for each attribute**
   Each attribute has different values in it. For eg: checking status has values of : *no checking, greater that 200, less than 0, between 0 and 200.* Data analysis using the vis were not explored here. We can include this in our future work, to show stacked bar charts showing that statistics of these values per attribute depending on their presence on all the rules. Meaning if there are 80 rules, and checking status has those values mentioned above, where *no checking* occurs 10 times, *greater than 200* occurs 5 times, *less than 0* occurs 3 times, and *between 0 and 200* occurs 5 times again, then clicking on checking status will also show a stacked bar plot with these values.

4. **How two rules, link to each other, if there is any linking that exists between the attributes**
   Also selection of two rules to see which ones have common attributes between them was not seen. This could be an interesting study where we could see if rules are

mutually exclusive or there is one attribute that is driving the dataset and occurs in every rule. This could also be interesting to see attributes move around between different rules on selection.

References

1. Analysis of German Credit Risk Data. Penn State Eberly College of Science, STAT 897D Applied Data Mining and Statistical Learning. Web. Accessed on May 19, 2016. URL: <https://onlinecourses.science.psu.edu/stat857/node/215>