PART 1: Data Cleansing (30 points)

Import the manager_performance_v2.csv dataset into Power BI, click the data tab, and open the Query Editor window. If you receive any errors during this process, that's okay. Be sure to address these errors and their assumptions during your data cleansing process and list them with your data transformations and assumptions.

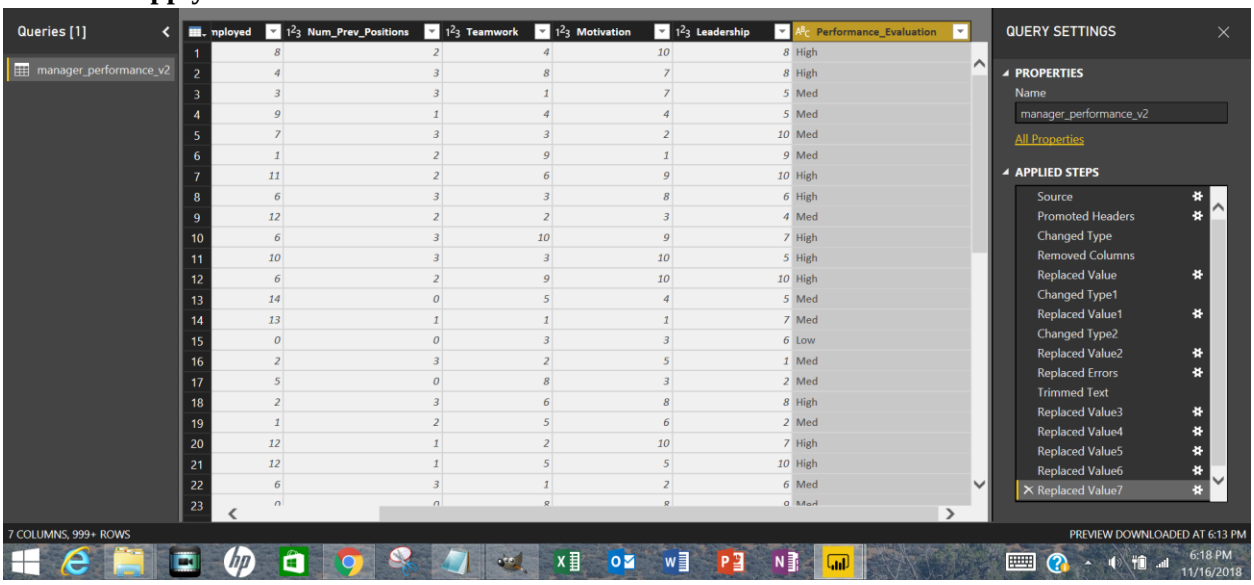
Go through each attribute column and perform various data cleansing actions necessary to cleanse the dataset. For each attribute, record each data cleansing step performed and the underlying assumption as to why the data cleansing action was performed.

Do not simply state that "all columns were trimmed" or restate the cleansing action itself. State the assumption (e.g., "M" was changed to "Male" because it was assumed that "M" indicated "Male" in this dataset.). Also, if no data transformations were made, state your assumption here as well (all data were assumed to be correct/clean).

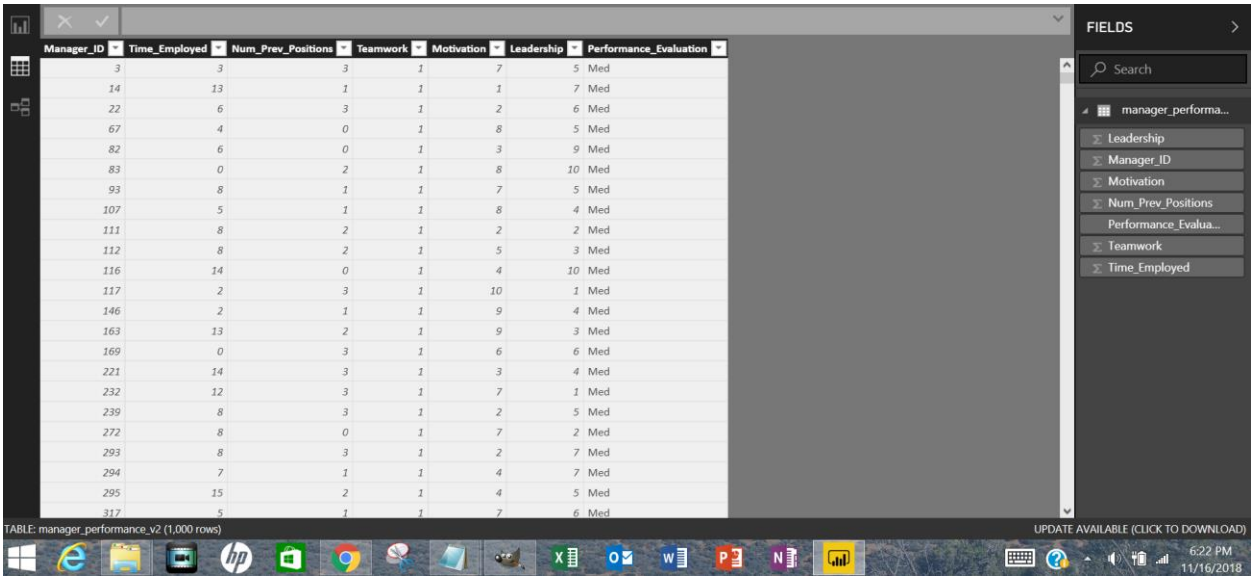
Think about any ethical concerns regarding this dataset. Remove any columns that personally identify employees or could be used to discriminate against employees (sex, marital status, age, sexual orientation, etc.).

1. When you're finished performing data transformations…

a. Take a screenshot of the Query Editor Window (this is a back up), then click close and apply.



b. Take another screenshot of the data tab. The screenshot does not need to illustrate all rows, but do include the total number of rows shown at the bottom of the data tab table and all remaining attribute columns (and the date/time stamp).



Manager_ID	Time_Employed	Num_Prev_Positions	Teamwork	Motivation	Leadership	Performance_Evaluation
3	3	3	1	7	5	Med
14	13	1	1	1	7	Med
22	6	3	1	2	6	Med
67	4	0	1	8	5	Med
82	6	0	1	3	9	Med
83	0	2	1	8	10	Med
93	8	1	1	7	5	Med
107	5	1	1	8	4	Med
111	8	2	1	2	2	Med
112	8	2	1	5	3	Med
116	14	0	1	4	10	Med
117	2	3	1	10	1	Med
146	2	1	1	9	4	Med
163	13	2	1	9	3	Med
169	0	3	1	6	6	Med
221	14	3	1	3	4	Med
232	12	3	1	7	1	Med
239	8	3	1	2	5	Med
272	8	0	1	7	2	Med
293	8	3	1	2	7	Med
294	7	1	1	4	7	Med
295	15	2	1	4	5	Med
317	5	1	1	7	6	Med

Example of .pbix screenshot (data tab circled to the left; number of rows circled at the bottom; and all attributes showing in the table – please also include your date/time stamp):

2. Save your work as a .pbix file in case you need it later.

a. For this portion of the assignment, add the list of assumptions (in Word)

Preface: *Upon importing the dataset, Power BI automatically "promoted headers" and performed a "change type" between "text" and "numeric" on roughly half of the attributes.*

Manager_ID: No modifications were made to this beyond the "change type" Power BI" implemented itself, it should not be necessary to further concern ourselves with this attribute as its function will be identification and every instance is unique**.**

First_Name: I have deleted this entire column because it is not a quantifiable value variable and is consequently insignificant to our business question/intent. Again, this not germane and would not improve or inhibit managerial performance in any meaningful way.

Last_Name: I have deleted this entire column because it is not a quantifiable value variable and is consequently insignificant to our business question/intent. Again, this not germane and would not improve or inhibit managerial performance in any meaningful way.

Age: I have deleted the "age" attribute from our dataset for ethical and liability concerns. If I were to discover that "age" was a significant influence on our "target variable", there would be no way to defend our conclusion as though it hadn't been a factor. This would not only be unethical, but would also put the company in violation of "Age Discrimination in Employment Act".

Time_Employed: I replaced the value "O" with "0" as I assumed all values in this column were intended to be integers and would otherwise be useless to our models as independent variables. I reformatted this column from "text" to "numeric" as I assumed it more accurately reflects their function for our purposes.

Num_Prev_Positions: I replaced the value "zero" with "0" as I assumed all values in this column were intended to be integers and would otherwise be useless to our models in their respective contexts. I reformatted this column from "text" to "numeric" as I assumed it more accurately reflects their function for our purposes.

Teamwork: I replaced the value "100" with "10" as I assumed all values in this column were intended to be integers (between 1-10) and would otherwise be useless to our models in their respective contexts. I

replaced error “ten” with the value “10” for the aforementioned reasons. (Attribute was already numeric)
Motivation: Upon inspection, I assume this attribute to be free of errors (attribute was already numeric).
Leadership: Upon inspection, I assume this attribute to be free of errors (attribute was already numeric).
Performance_Evaluation (target): I trimmed any leading or trailing spaces from “Med” because I assumed there were not meant to be 3 variations of the variable**.** I replaced the value “Med.” and “med” with “Med” because I assume there are not meant to be any additional variations of that variable. I replaced the value “Loww” with “Low” as I assume there should only be one variation of that variable. I replaced the value “High” with “Hi” then back to “High” in order to normalize some invisible character defect; I did this for the same rationale as “Low” and “Med”.

PART 2: Unsupervised DM (32 points)

Return to Canvas and download the manager_performance_v2_clean.csv. This file is provided so that any errors potentially made during data cleansing do not result in subsequent errors/deductions for the remaining portions of the assignment. Look through this data set and determine the types of data variables therein (nominal, ordinal, ratio, or interval). In RapidMiner, import the clean dataset and conduct an unsupervised data mining technique appropriate for this dataset. Think about the data variable types in this dataset then choose from association analysis or cluster analysis.

HINT: Revisit the ‘summary slide’ for each of these techniques to remember what type(s) of data variables can be input into these models.

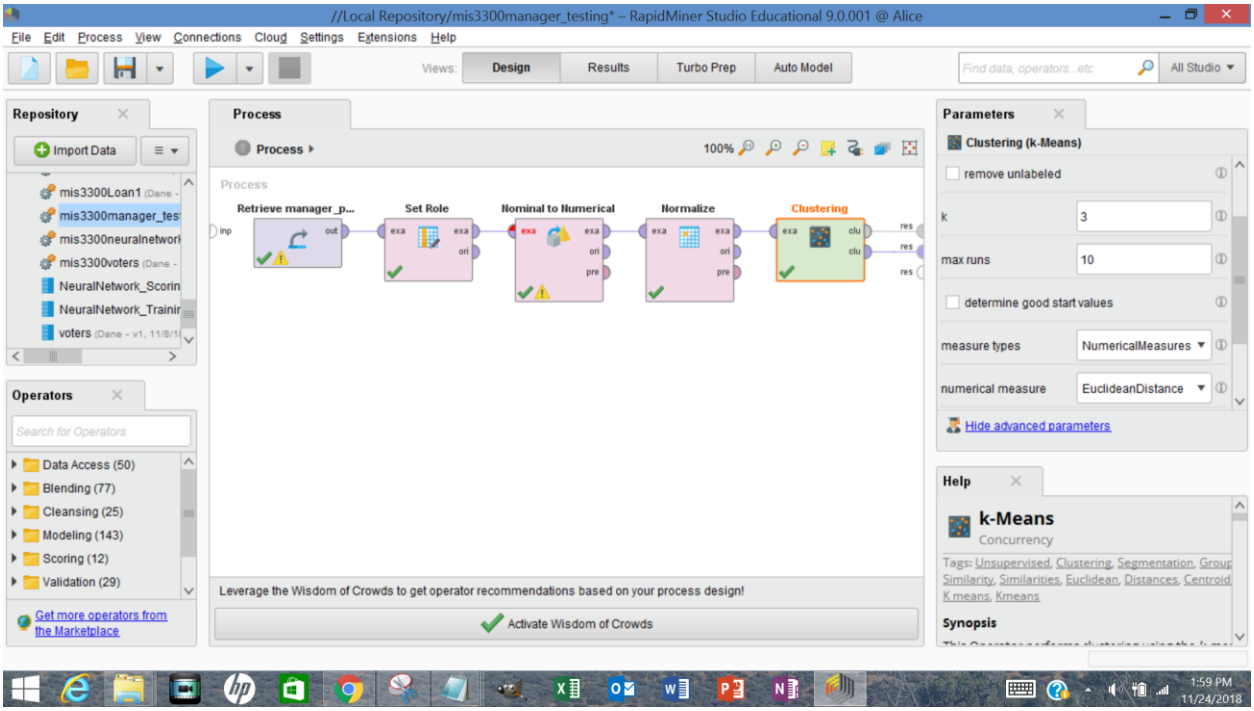
In addition to your chosen model operator, you will need to use the following operators:

- Retrieve dataset
- Set role: set ‘Manager_ID’ as ID and ‘Performance_Evaluation’ as cluster
- Nominal to Numerical: Choose the single attribute ‘Performance_Evaluation‘
- Normalize: Choose a subset of attributes – all except ‘Manager_ID’
- Choose an appropriate and informative model operator (we’ve used it before in class). Leave all of its parameters as the default setting except one. Think about what we’re interested in figuring out with this manager performance dataset (performance evaluation) and how many classes we have for this attribute. Change one parameter based on this.

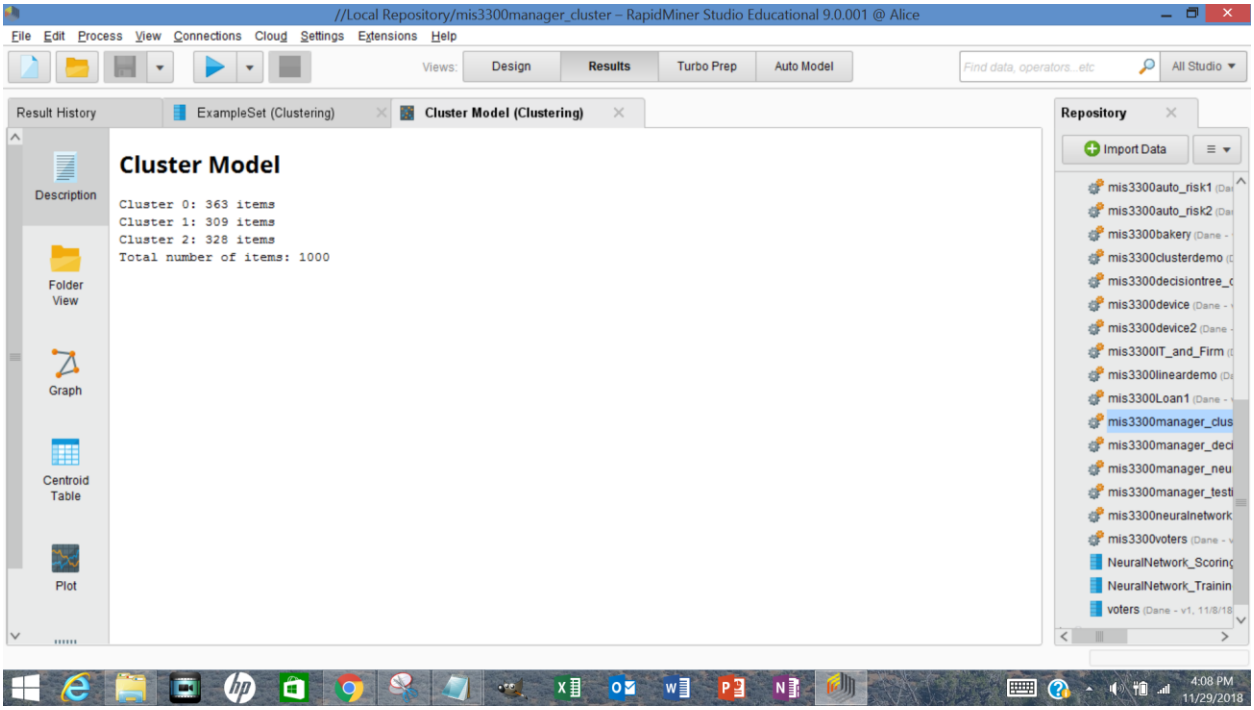
A. Identify which model operator was selected and why. The why should focus on the types of data variables in the manager performance data set. Also discuss any parameters changed and why.

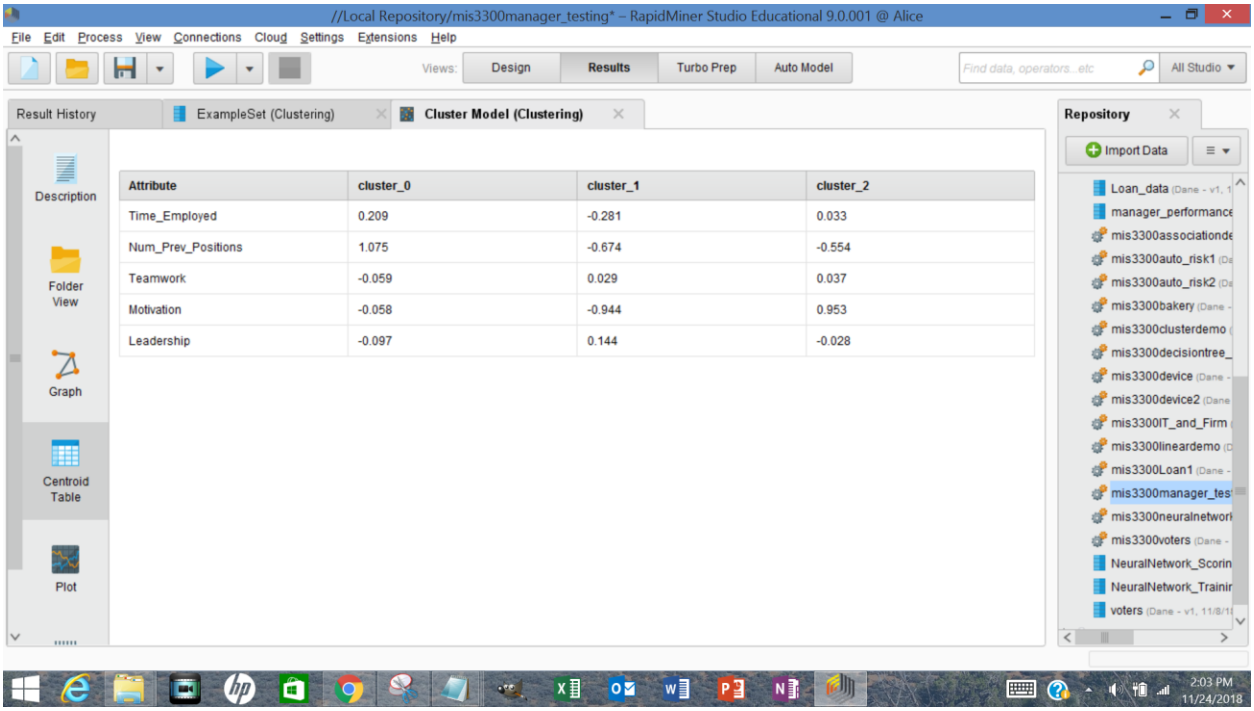
The model operator selected was the “K-Means” cluster in order to sort the data by relevant groupings according to the attributes that most inform each cluster and the business question that we are asking. Because the data was a mix of text and integer, the “performance_evaluation” attribute had to be made numeric, and because the data were ordinal and ratio, it had to be normalized by z-transformation. After our variables have been standardized by number, it was necessary to employ an aggregate distance measure, for these reasons, I select “numeric measures” and “euclidean distance” in the “K-means” cluster operator. Similarly, because I would like to observe three specific variables (the high/med/low) of our “performance_evaluation” attribute, I selected k = 3 clusters.

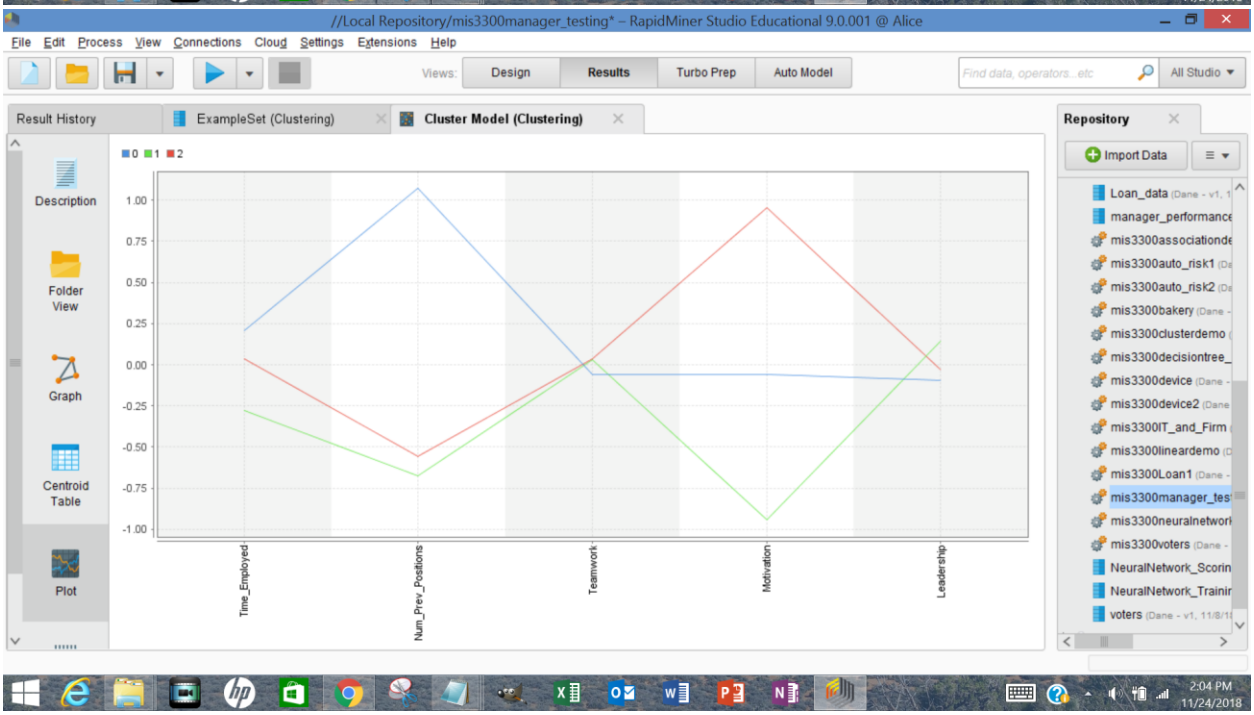
B. Include screenshots of your RapidMiner process window and relevant results screens and interpret these results. Revisit the previous homework assignment for the chosen model to remember what the relevant results screens and interpretations should focus on.



These choices are largely explained by the given instructions and the answer to question A.







Because I have set “perfomance_evaluation” to “cluster” in the set role operator, set k=3 in the k-means cluster operator, the “performance_evaluation” attribute does not appear in our centroid table nor associated chart. And because we know that the business question is seeking information about “high/med/low” variables, we can assume that the given clusters are representative of those 3 variables. Looking at the chart, I see the most variation (intra-cluster distance) belongs to the “motivation” attribute and the “num_prev_positions” with less variation between 2 of the clusters. Examining deviation from the mean in our centroid table, I believe that “motivation” above the mean should be indicative of a “high” performance score, closer to the mean a “med” performance score, and below the mean a “low”

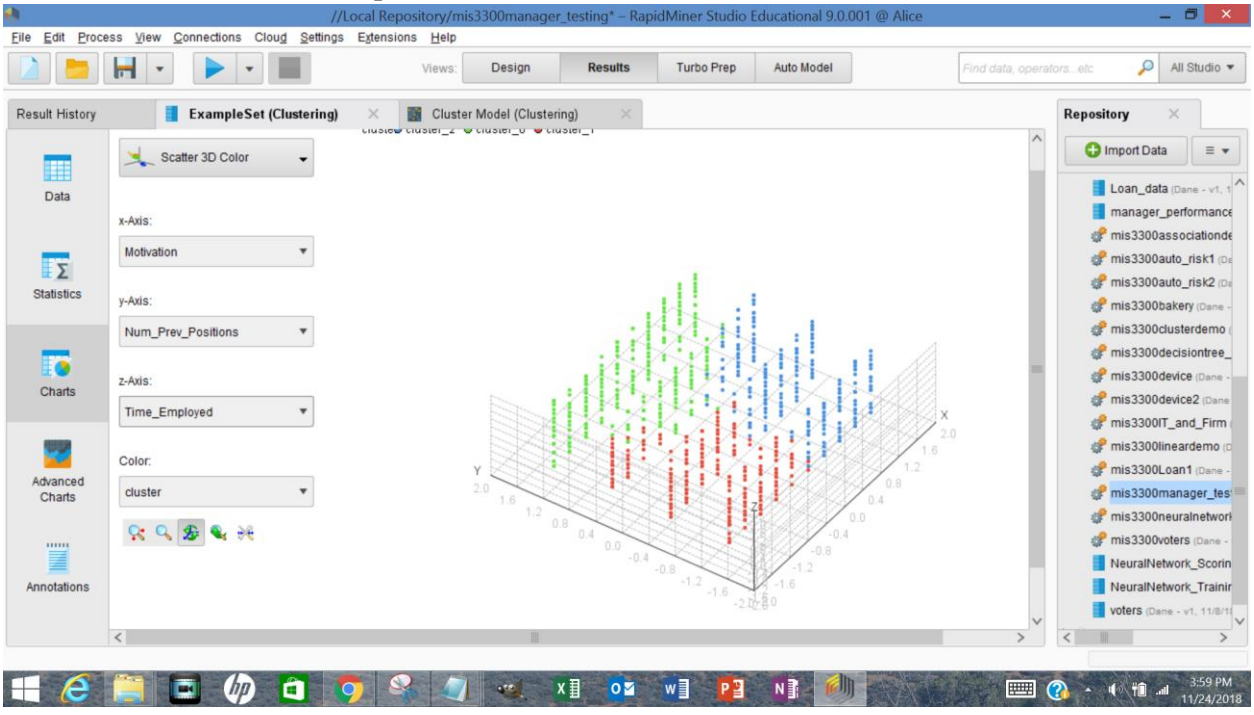
performance score, which then separates our clusters into 0=med, 1=low, and 2=high. Additionally, it stands to reason that motivation is the most significant factor for the existing "performance_evaluations" in our data. Alternately, we can describe the clusters as follows:

Cluster 0: The "med" performance cluster of our data has been employed for slightly greater duration than "med" or "high" performance, but has held a much greater amount of previous positions and its teamwork, motivation, and leadership scores are all just marginally below the mean. An assumption might be made that this is our most experienced group, but also our most cynical and only do what is necessary to get by.

Cluster 1: The "Low" performance cluster of our data has been employed the least amount of time and held the least previous positions and its teamwork and leadership scores are marginally above the mean, while its motivation score is by far the worst. An assumption might be made that this is our least experienced group, and while eager to lead and collaborate, either unsure how or unwilling to self-start and make things happen.

Cluster 2: The "high" performance cluster of our data has been employed slightly longer duration than "Cluster 1", held a few more previous positions, and its teamwork and leadership scores are just about the mean; however, its motivation score is by far the best of the set. An assumption might be made that this is an adequately experienced segment with the motivation, and likely the know how, to get work done.

C. Create at least one visualization (in RapidMiner or Power BI) with a caption or description about how this visualization contributes towards the meaningful interpretation of the manager performance data. You cannot use a results screen automatically generated by RapidMiner. You must draw from the visualization portion of this course and create your own relevant visualization, label it, and include a brief caption.



The largest intra-cluster distances were created by the motivate, previous positions, and time employed attributes (respectively); consequently, I have enlisted these attributes in hopes of illustrating meaningful

cluster separation and some semblance of inter-cluster homogeneity.

PART 3: Supervised DM (32 points)

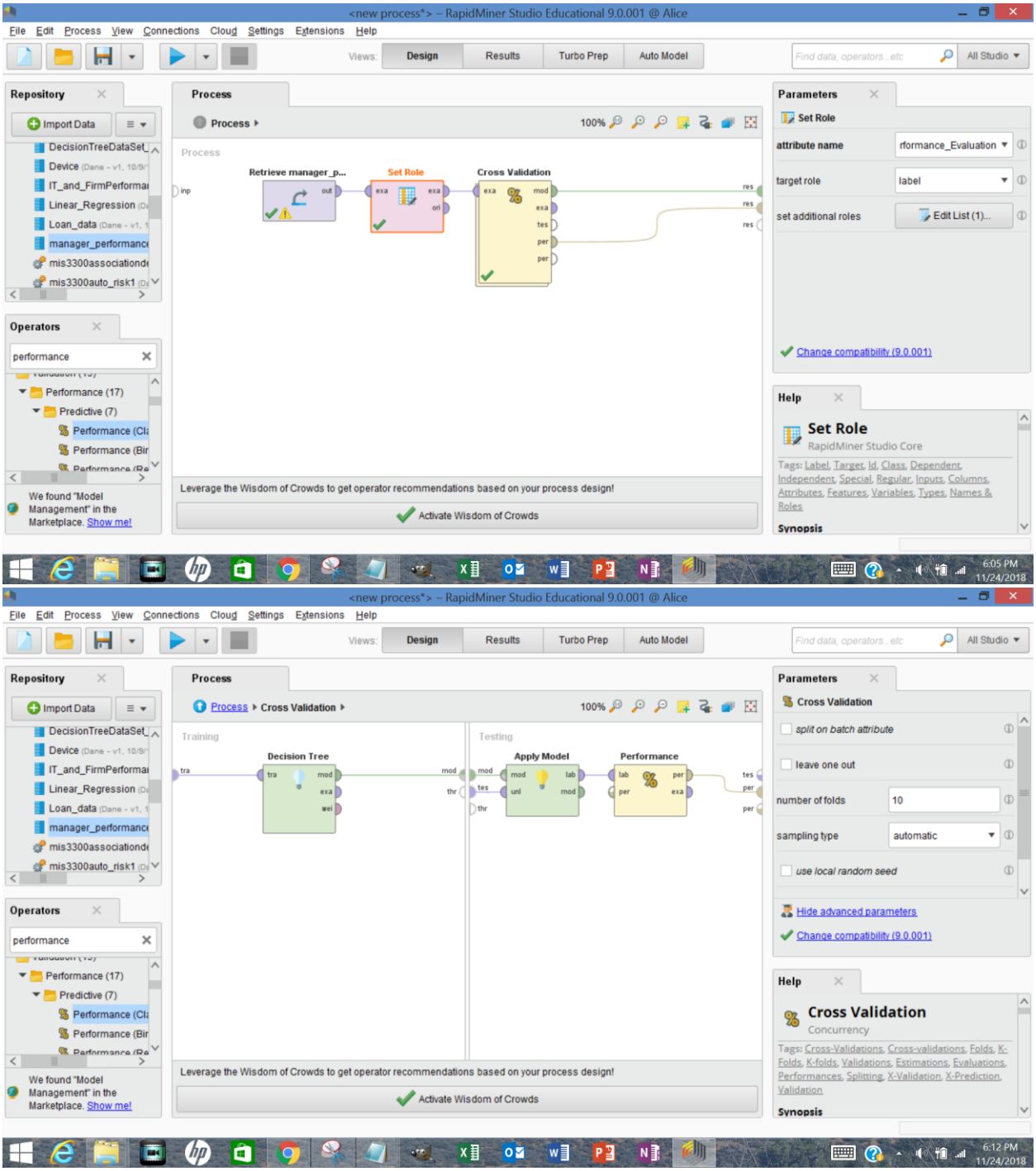
Using the manager_performance_clean_v2.csv file from Canvas, conduct a supervised data mining technique other than neural network analysis (which will be done below) in RapidMiner. You will compare this chosen supervised data mining technique to your neural network analysis later. Think about comparing these models and the data variables in this dataset, then choose from linear regression or decision tree analysis. Be sure to choose a model that is appropriate for the data variables used in the analysis and will compare well to a neural network model!

In addition to your chosen model operator, you will need to use the following operators:

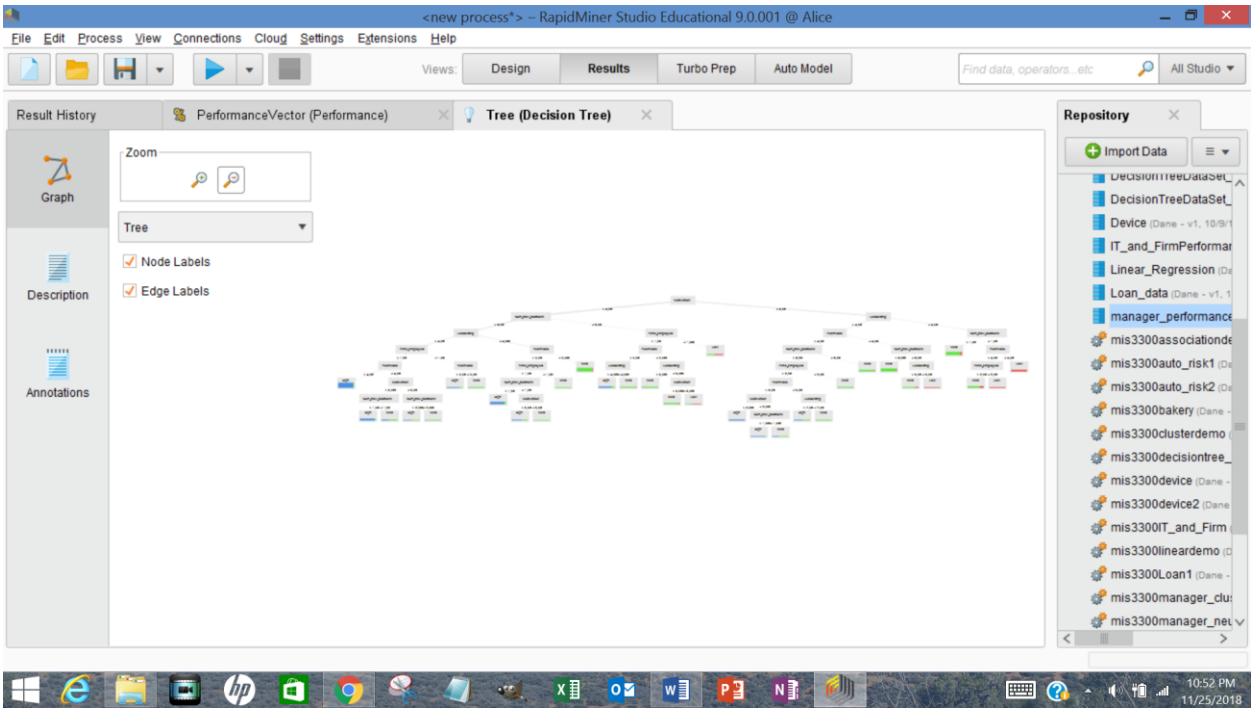
- Retrieve dataset
- Set role: set 'Manager_ID' as ID and 'Performance_Evaluation' as label
- Cross-Validation
    - Choose an appropriate and informative model operator (we've used it before in class). Leave the default parameters except set minimal leaf size to 10 and ensure maximal depth is set to 20.
    - Apply Model
    - Performance: Choose the correct type of performance operator depending on the chosen model operator (classification, binominal classification, or regression). Select the following performance metrics: accuracy, classification error, and kappa.

A. Identify which model operator was chosen and why. Focus on the data variable types in the manager performance data set AND a model type that compares well to neural network analysis (HINT: Think about different types of supervised models we've learned about – regression versus classifiers.**).** Because our data set is categorical and our target attribute is multinomial, the "decision tree" operator is ideal to predict the likelihood of our 3 performance outcomes (dependent variables) based on the associated attributes (independent variables) in our data set. Additionally, the decision tree process of calculating entropy to determine information gain is not terribly dissimilar from the way neural networks use the Bernoulli error function to weight node signals and improve accuracy, which may contribute to the comparable model interpretability evidenced in their confusion matrices and kappa values.

B. Include screenshots of your processes and relevant results screens and interpret these results. Do not just restate results. Tell me what these results mean. Revisit the previous homework assignment for the chosen model to remember what the relevant results screens and interpretations should focus on.



The process is partially explained by the instructions and the answer to question A; however, the set role operator was used to identify our target variable (performance_evaluation) and isolate the manager id as identification (pointless to split node by ID). Decision tree operator was selected for the aforementioned reasons and to act as the "training model", the apply and performance "classification" operator were selected as "testing model" with the latter also selected to help predict our target variables given the presence of our independent variable attributes. The accuracy, classification error, and kappa values were checked in the performance operator to compliment the model's confusion matrix and improve interpretability of the model and its fitness.



a. (10 pts) The first three nodes of the decision tree. (Starting at the root node of the tree, explain the classification process in terms of each of the rules represented by the first three nodes in the tree.)

The root node for this model is "Motivation" signifying the highest information gain above the minimal gain threshold required to split a node. The "Motivation" then splits between greater or less than/equal to a 4.500 average evaluation score for motivation, where every observation above 4.500 directs to the interior node of "Num_Prev_Positions". Every observation below/equal to 4.500 splits to the interior node "Leadership", both of which are the next highest information gain for the model in respective order. "Num_Prev_Positions" is then split between observations with greater or less than/equal to 0.500 and all cases greater than 0.500 continue to "Leadership". (In the interest of brevity and a desire to reach a terminal leaf node, we will follow only "greater than" branches to the nearest leaf node) From the aforementioned "Leadership" node, we follow greater than 4.500 to the "Time_Employed" node. From the "Time_Employed" node, we follow greater than 1.500 to the "Teamwork" node. From the "Teamwork" node, we follow greater than 4.500 to the terminal leaf node "High", which has the largest concentration of "High" performance observations at 127 out of 132 for the leaf.

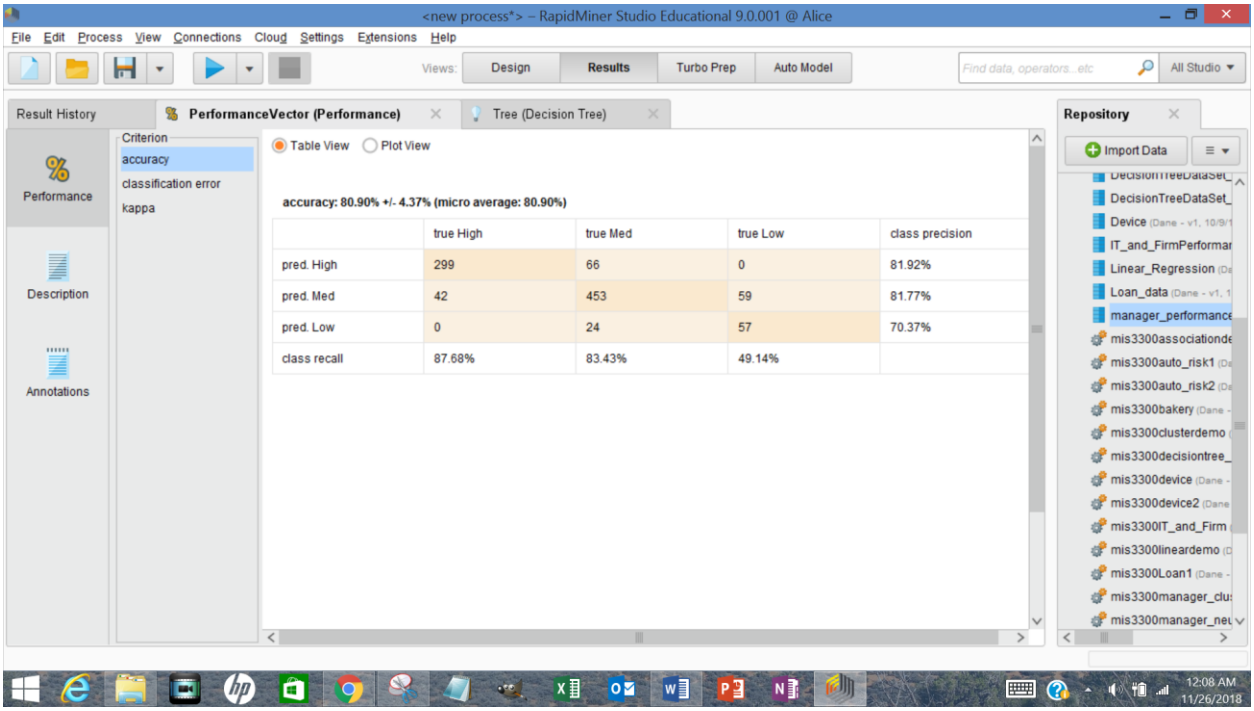
Note: "Information Gain" is determined by RapidMiner's algorithm systematically reducing entropy to promote "purity" (homogeneity of observed DV values within a node).

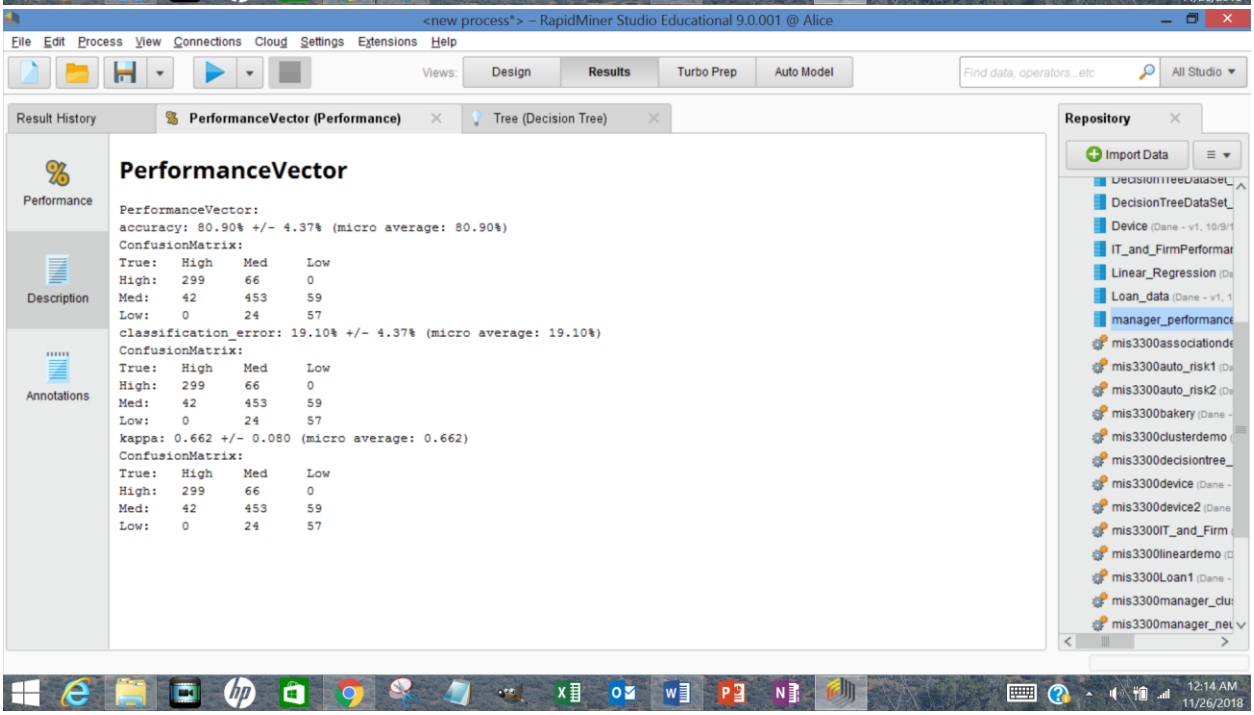
b. (10 pts) Which leaf node in the tree has the most entropy? Provide a calculation of the entropy in this node (show your work).

The "Low" terminal leaf node less than/equal to 1.500 resulting from "Time_Employed" (one node removed from the root) has the most entropy with 0 "High", 10 "Med", and 11 "Low".

$$-\left(\frac{10}{21}\cdot\log_2\left(\frac{10}{21}\right)\right)-\left(\frac{11}{21}\cdot\log_2\left(\frac{11}{21}\right)\right)=.9984$$

Note: I omitted the "High" variable "-(0/21*Log$_2$(0/21))" as it is "zero". Also, the full result of this entropy calculation was ".9983636726".



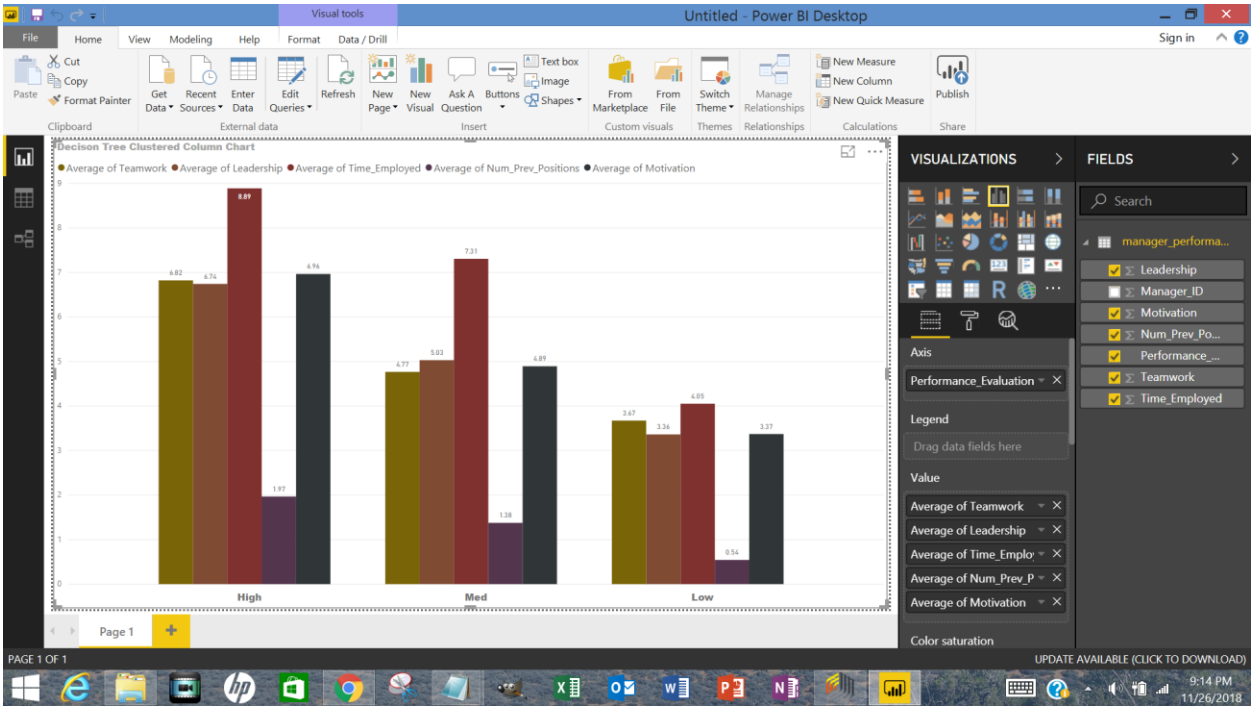


The model's accuracy is a decent 80.90%, meaning its total high/med/low predictions were accurate 809 times out of the 1000 observations in our model. Although this is reassuring for our model, it is less significant than class precision or class recall because the class distributions are uneven and the false positives outweigh the false negatives for our business purpose. For instance, even though "true high"s class recall is 87.68%, the model is not predicting the other 12.32%, or 42 false negative observations in that segment who are likely qualified for the position in question but not included. However, I believe the false positive of 66 observations actually being "true med" when the model predicted "high" to be more detrimental to our business purpose because employing medium performers we thought were high

performers would likely be more costly than just overlooking a segment of qualified applicants; consequently, the 81.92% class precision should be our most significant determinant of fitness (which is still pretty decent). The model's prediction of "true low" is the worst with a class recall of 49.14%, meaning that it is miscategorizing 50.86% of its "low" predictions. Though, for our purposes, it does not appear that the errors in our model's "low" prediction result in any false negatives or false positives regarding our "high" performance variable. Cohen's Kappa (0.662 for our model), is a measure of how well our model is performing above "chance" (approx .4347 for our model), on a scale of 0 to 1. Ergo, it should be reasonable to assume out model is performing moderately well by measure of "kappa". Consequently, I believe that a business could assume this model would predict high managerial performance with roughly 81.92% accuracy as per the class precision for our "high" prediction segment.

Note: Because "class distribution" is uneven and "false positive" is more heavily weighted, "precision" should our model's unit of of measure.

C. Create at least one visualization (in RapidMiner or Power BI) with a caption or description about how this visualization contributes towards the meaningful interpretation of the manager performance data. You cannot use a results screen automatically generated by RapidMiner. You must draw from the visualization portion of this course and create your own relevant visualization, label it, and include a brief caption.



I believe this clustered column chart with the averages of the attribute scores and measures was the most representative of the underlying data that informs our decision tree. This visualization illustrates the average contribution each attribute is making toward the target variables we intend to predict, and as such, gives us a pretty good idea what to expect and a loose basis for comparison concerning model fitness.

PART 4: Supervised DM – Neural Network Analysis (47 points)

Using the manager_performance_clean_v2.csv, conduct a neural network analysis in RapidMiner. Use the following operators:

• Retrieve: Retrieve the dataset
• Set Role: Choose Performance_Evaluation in the attribute name drop down and set the target role to ‘label’. Also, set the Manager_ID attribute to the role ‘id’.
• Cross Validation: Conduct a cross validation with 10 validation runs and sampling type set To automatic. Connect the ‘mod’ and ‘per’ output ports to the result ports. Set the sub-operators of this operator as follows:

o Training:

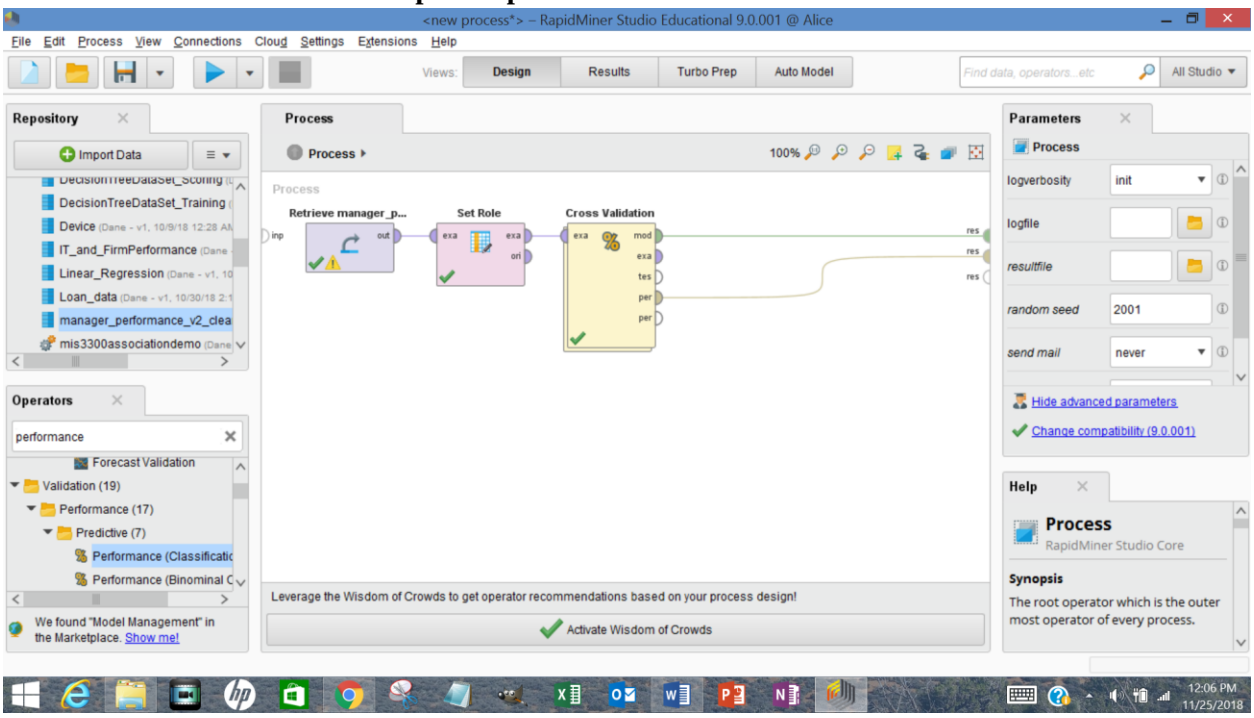
▪ Neural Net: Use Neural Net as the training operator. Leave all parameters set at their default values. Connect the ‘tra’ port on the left side of the training pane to the ‘tra’ input port. Connect the ‘mod’ output port to the ‘mod’ input port in the right side of the training pane.

o Testing

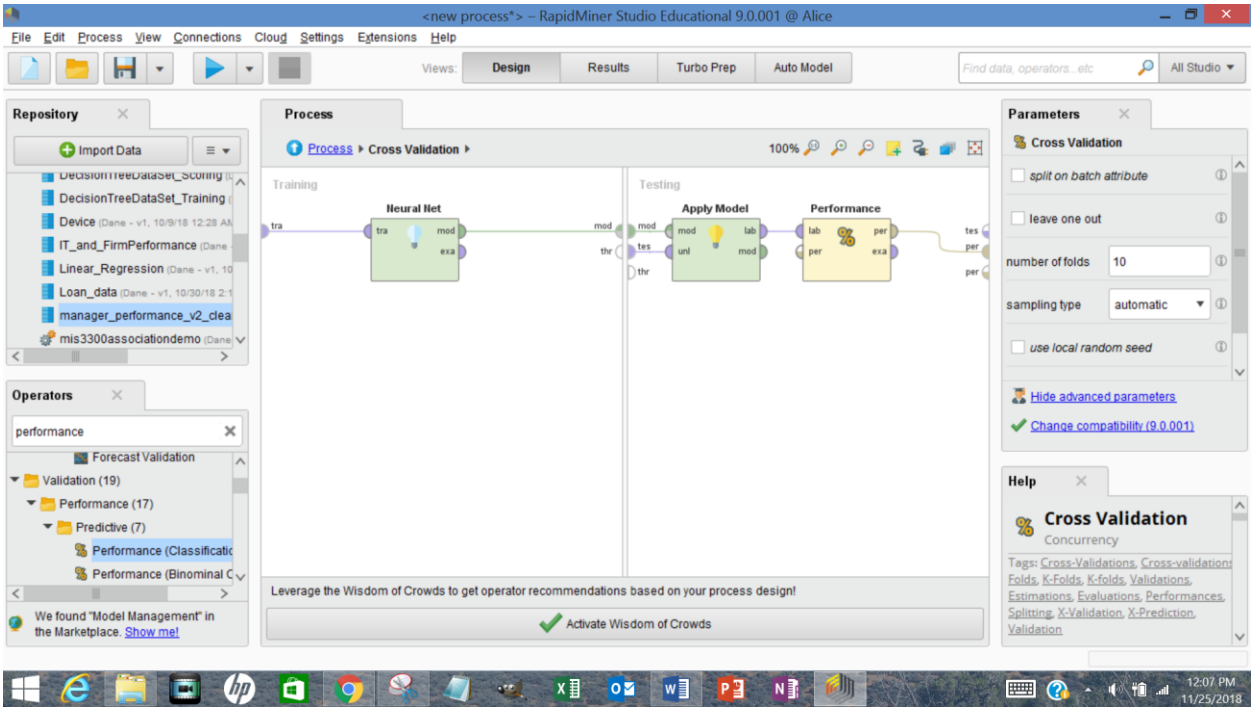
▪ Apply Model: Use the apply model to receive the model from the training phase and apply it. (Connect ‘mod’ and ‘tes’ ports on the left of the testing pane to the ‘mod’ and ‘unl’ input ports, respectively. Connect the ‘lab’ output port to…
▪ Performance (Classification): leave the ‘main criterion’ parameter set to ‘first’ and choose the following performance measures: accuracy, classification error, and kappa. Connect the ‘per’ operator port to the ‘per’ port on the right-hand side of the screen.

A. Provide two screenshots of the RapidMiner process in design view as follows:

1. First screenshot: the top-level process

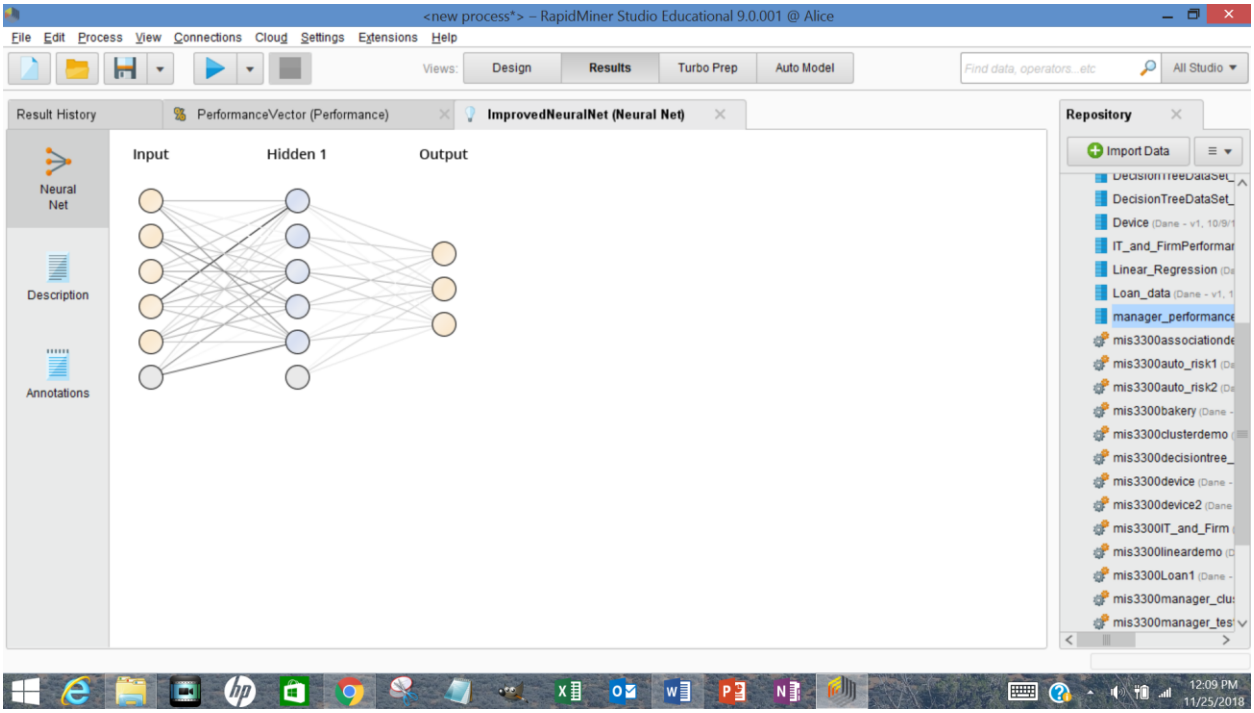


2. Second screenshot: the nested process inside the Validation operator

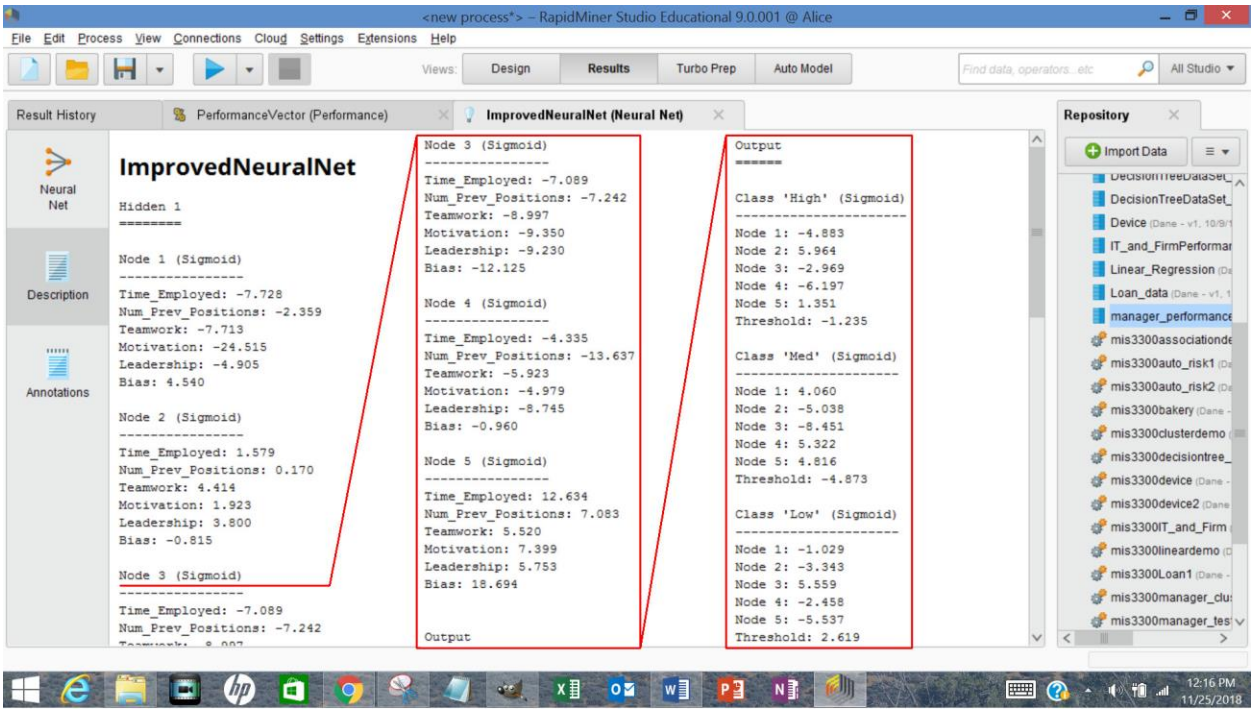


B. After running the process, go to the ImprovedNeuralNet (Neural Net) results screen and provide screenshots of the following:

1. **The graphical representation of the network shown on the 'Neural Net' tab on the left side.**



2. **The output shown on the 'Description' tab on the left side (multiple screenshots will be necessary)**



C. Referencing the output from the process, do the following:

1. **Examine the graphical output of the network. The darker lines indicate stronger connections between the nodes. Refer to your results in Part B2 above; identify the two pathways (excluding the threshold nodes) that seem to have the most impact on each of the high/med/low outcome nodes. As a hypothetical example:**

Strongest Pathways to High Outcome:

Num_Prev_Positions (-13.637) → Hidden Node 4 (-6.197) → High

Teamwork (4.414) → Hidden Node 2 (5.964) → High

Strongest Pathways to Med Outcome:

Motivation (-9.350) → Hidden Node 3 (-8.451) → Med

Num_Prev_Positions (-13.637) → Hidden Node 4 (5.322) → Med

Strongest Pathways to Low Outcome:

Motivation (-9.350) → Hidden Node 3 (5.559) → Low

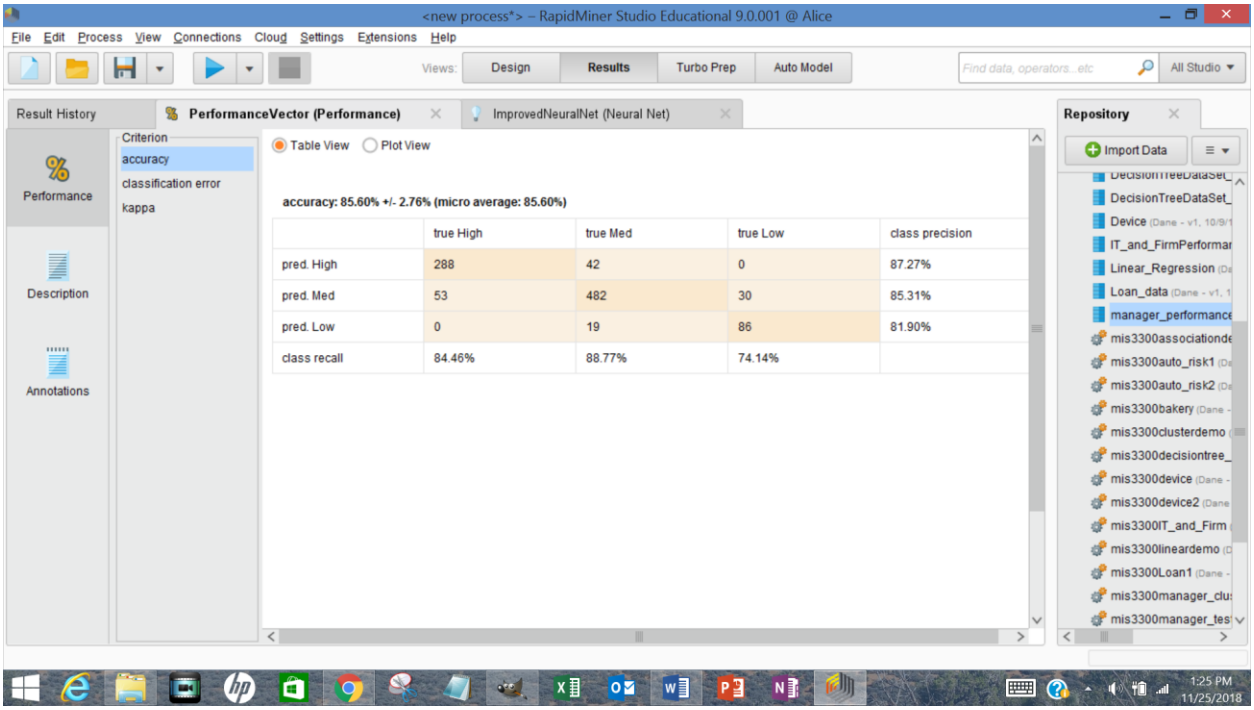
Time_Employed (12.634) → Hidden Node 5 (-5.537) → Low

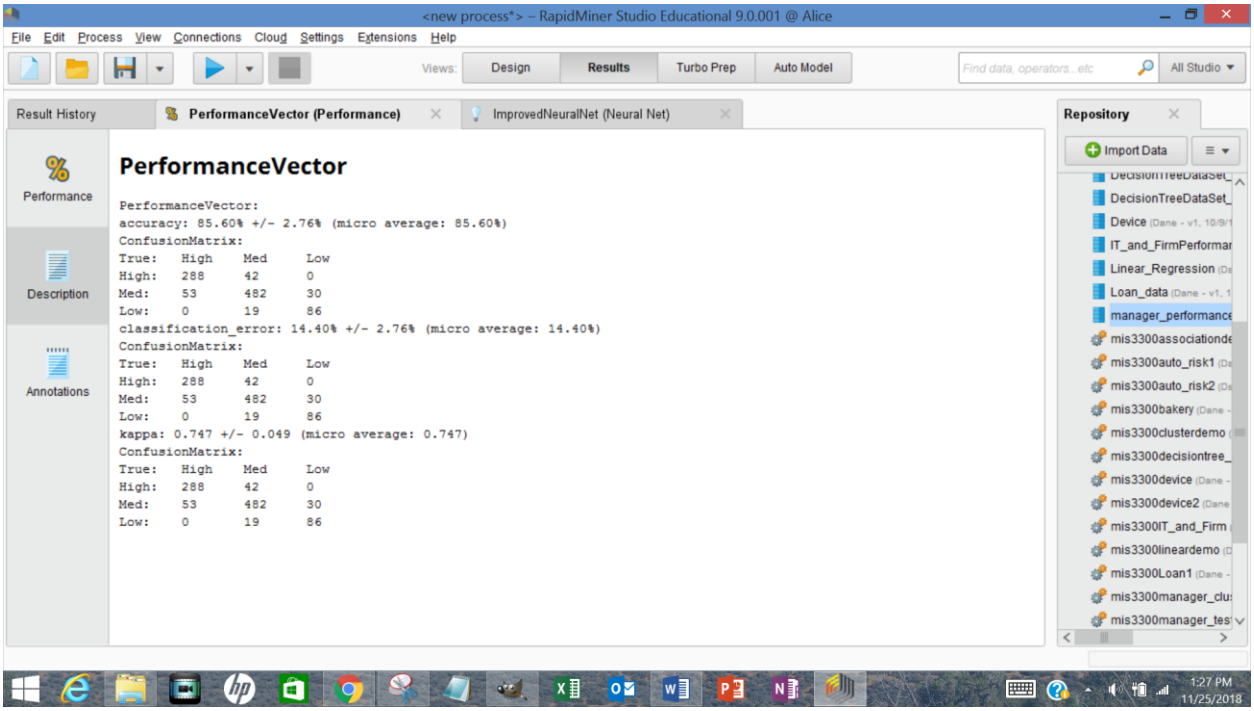
2. **Referring to your answers in C1 above, provide a few-sentence description of the pathways you identified for each outcome node (e.g. from the hypothetical example above, "From the network structure, it appears that less time employed decreases the likelihood of a 'high' outcome, while strong leadership increases this outcome."). Focus only on interpreting these results; you do not need to rationalize whether it makes sense or not here.**

From the network structure, it appears that a lesser number of previous positions decreases the likelihood of a 'high' outcome, while strong teamwork increases this outcome. From the network structure, it appears that weak motivation decreases the likelihood of a 'med' outcome, while a lesser number of previous positions actually increases the likelihood of this outcome. From the network structure, it appears that weak motivation increases the likelihood of a 'low' outcome, while greater time employed

decreases this outcome.

D. Go to the PerformanceVector results screen and provide screenshots of the confusion matrix shown on the 'Performance' tab on the left-hand side and also the results shown on the 'Description' tab. Interpret the performance of this model in terms of accuracy, precision, recall, and kappa. Which outcome class (high/med/low) does the model predict the worst? What might be the practical (business) implications of this?





The model's accuracy is a respectable 85.60%, meaning its total high/med/low predictions were accurate 856 times out of the 1000 observations in our model. Although this is reassuring for our model, it is less significant than class precision or class recall because the class distributions are uneven and the false positives outweigh the false negatives for our business purpose. For instance, even though "true high"s class recall is 84.45%, the model is not predicting the other 15.55%, or 53 false negative observations in that segment who are likely qualified for the position in question but not included. However, I believe the false positive of 42 observations actually being "true med" when the model predicted "high" to be more detrimental to our business purpose because employing medium performers we thought were high performers would likely be more costly than just overlooking a segment of qualified applicants; consequently, the 87.27% class precision should be our most significant determinant of fitness (which is still quite decent). The model's prediction of "true low" is the worst with a class recall of 74.14%, meaning that it is miscategorizing 25.86% of its "low" predictions. Though, for our purposes, it does not appear that the errors in our model's "low" prediction result in any false negatives or false positives regarding our "high" performance variable. Cohen's Kappa (0.747 for our model), is a measure of how well our model is performing above "chance" (approx .4315 for our model), on a scale of 0 to 1. Ergo, it should be reasonable to assume out model is performing pretty well by measure of "kappa". Consequently, I believe that a business could assume this model would predict high managerial performance with roughly 87.27% accuracy as per the class precision for our "high" prediction segment.

Note: Because "class distribution" is uneven and "false positive" is more heavily weighted, "precision" should our model's unit of of measure.

PART 5 – Evaluation of Models & Business Recommendations (29 points)

Compare the two supervised data mining models from Parts 3 and 4 above.

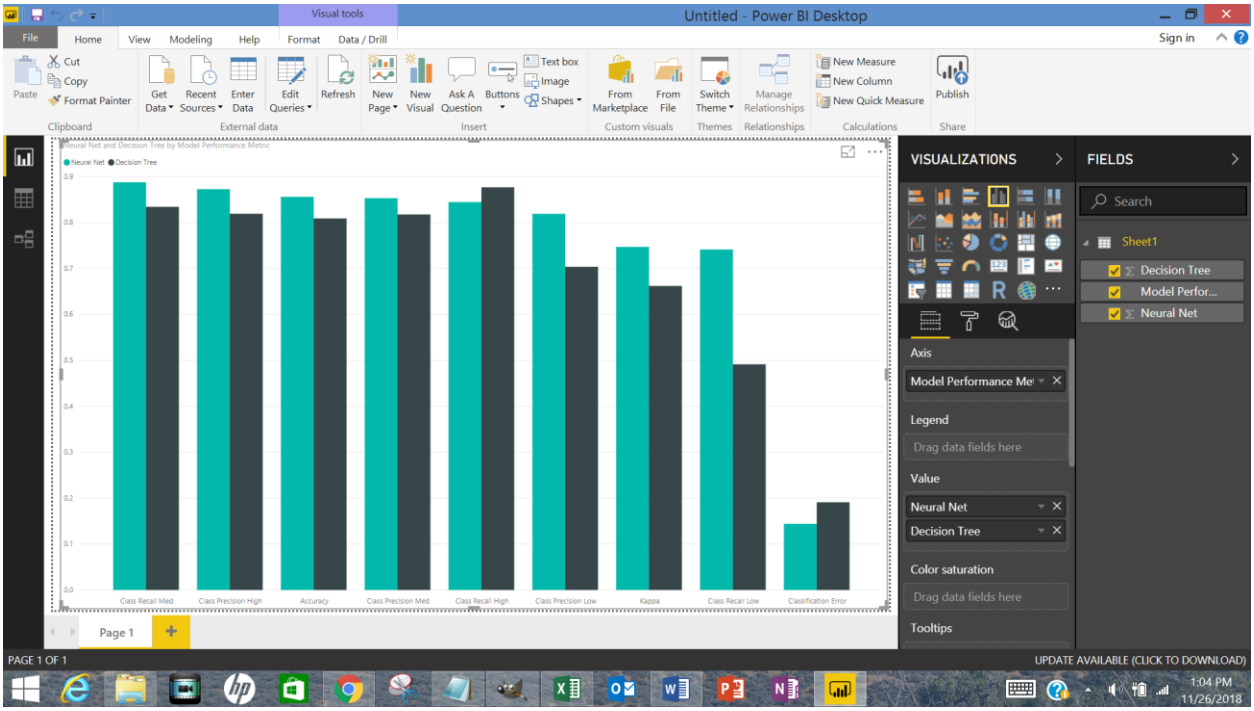
A. Recopy each Performance Vector results tab (description tab) outputs from the two supervised models above (Parts 3B & 4D).

Model Performance Metric	**Decision Tree (Part 3)** "Model 1"	**Neural Net (Part 4)** "Model 2"	**Which is Better? Or Same**
Accuracy	**80.90%**	**85.60%**	**Model 2: Better**
Classification Error	**19.10%**	**14.40%**	**Model 2: Better**
Kappa	**0.662**	**0.747**	**Model 2: Better**
Class Precision High	**81.92%**	**87.27%**	**Model 2: Better**
Class Precision Med	**81.77%**	**85.31%**	**Model 2: Better**
Class Precision Low	**70.37%**	**81.90%**	**Model 2: Better**
Class Recall High	**87.68%**	**84.46%**	**Model 1: Better**
Class Recall Med	**83.43%**	**88.77%**	**Model 2: Better**
Class Recall Low	**49.14%**	**74.14%**	**Model 2: Better**

B. Which model performed better and why? Which performance measures (list their values) were used to determine this and why?

I believe that the Neural Network model performed better than the Decision Tree model by virtually every metric, with the exception of "class recall high" for our intended business purpose. Neural Network accuracy is nearly 5 percent higher, kappa is performing roughly 32 percent above chance (compared to 23 percent for Decision Tree), and most importantly for our purposes, "class precision high" is performing over 5 percent better for this model. Again, I've chosen "precision" as the determinate metric for our models because the class distribution is uneven and the "false positive" error is more costly for our business intentions than the "false negatives". The rationale for cost weighting is taken directly from 25:18 of our 11.02 Decision Tree Analysis lecture stating that "false positives were predicted "yes" and they are actually "no", so they are predicted to be class A, but they are not. So in the table, it is the rest of this row, the 3 remaining cells. The model predicted them to be class A, but "x" amount were class B, "y" amount were class C, etc." From this we can assume that a false positive in our model is more costly because it predicted "High" performance that was actually "Medium", which might lead us to employ a medium performer in place of a high performer. Consequently, because the Neural Network was more accurate with respect to "class precision high", I can assume it is better suited to our purposes.

C. What business recommendations can be made after this analysis? Create a visualization to demonstrate which model is performing better and provide it here. Include a caption about how this visualization illustrates which model is performing better.



This visualization illustrates that the Neural Network model outperformed the Decision Tree model on every metric but the “class recall high”, which is not an integrally significant metric for our purposes. More importantly, Neural Network outperforms the Decision Tree model with respect to “class precision high”, this means it is less likely to predict an observation is a high performer when the are actually a medium performer. Additionally, it is important to note that the model did not mispredict any “low” performers as “high” and the model has a .747 Kappa value which means the the model is performing roughly 32 percent above random chance. From the data modeling done and information gleaned, I would suggest that the business should implement a Neural Network model to locate more qualified applicants for managerial purposes. This is the cumulative determination arrived at by rigorously classifying observations according to categories and levels of performance therein to ensure fitness of model and appropriateness of outcomes.