Injury Severity Classification for Insurance Premium Pricing Analysis

MSIS 5633 Predictive Analytics – Term Project

Team 5

# Table of Contents

[Table of Contents 1](#_Toc39481861)

[Executive Summary 2](#_Toc39481862)

[1. Business Understanding 3](#_Toc39481863)

[1.1 Project Team: 4](#_Toc39481864)

[1.2 Evaluation Criteria: 4](#_Toc39481865)

[1.3 Project Timeline 6](#_Toc39481866)

[2. Data Understanding 6](#_Toc39481867)

[2.1 Data Selection and Variable Description 7](#_Toc39481868)

[3. Data Preparation 8](#_Toc39481869)

[3.1 Data Filtering 9](#_Toc39481870)

[3.2 Rule Engine (Transformation) 10](#_Toc39481871)

[3.3 Rule Engine (Data Binning) 11](#_Toc39481872)

[3.4 Model Preparation 13](#_Toc39481873)

[4. Modeling 15](#_Toc39481874)

[4. 1 Decision Tree 16](#_Toc39481875)

[4. 2 Random Forest 18](#_Toc39481876)

[4. 3 Support Vector Machine 20](#_Toc39481877)

[4. 4 Artificial Neural Network 23](#_Toc39481878)

[5. Evaluation 26](#_Toc39481879)

[5.1 Variable Dependence 27](#_Toc39481880)

[6. Deployment 29](#_Toc39481881)

[References 30](#_Toc39481882)

# Executive Summary

**Business Understanding:** The Stark Insurance company is constantly looking for new way to use analytics to improve the company’s performance. A new proposed project using predictive analytics combined with National Highway Traffic Safety Administration data promises to be able to predict injury severity among the current pool of insurance customer and future application for car insurance. The Goal of predicting high vs low injury severity will allow the company to better assess risk and price medical portion of our customer’s car insurance premiums. The project will follow the CRISP-DM methodology, and the project team is comprised of a cross section of the company executives and data science teams.

**Data Understanding:** The data used for the project is Crash Report Sampling System data from the National Highway Traffic Safety Commission. The data was divided into 4 tables: Accident, Person Vehicle. The fourth table, Distract, was not relevant to the study. Selecting variables from these 3 tables was accomplished by only selecting variables that are currently collected by car insurance companies about their customers, vehicles, or historical crash data from their customer pool. This selection methodology allowed the team to limit variables for this project to 21 variables that met the selection criteria for the project.

**Data Preparation:** Data preparation removed/reduce noise from the data set. Table Joining, Data Filtering, Rule Engine (transformations, binning) were used for initial data preparation in KNIME. Final prep for modeling required the following KINIME transformations depending on model type: Number to String, String to Number, Domain Calculator and Normalizer nodes.

**Modeling:** Decision Tree, Random Forest, Neural Network and Support Vector Machine classification predictive models predicted INJ\_SEV of vehicle occupants. The model that performed the best and met or exceeded all evaluation criteria was Random Forest model with Accuracy: 83.29%, Sensitivity: 98.206%, Specificity: 17.933% with high explanatory value.

**Evaluation:** The Random forest model was the only model of the four that met or exceeded all four of the selection criteria outlined in business understanding for the project to move forward. The model has a low false negative rate and high accuracy which will help the insurance company reduce risk in pricing medical portion of its current and future customers medical premium portion of their car insurance premium.

**Deployment:** Since the Random Forest model met all criteria for the project, Tony and the Stark Board of Directors chose to move the project into deployment. Peter Parker and his application development team will take over the project on 5.7.20 and work in three team simultaneously integrating the model into the Stark Insurance CRM, Premium Pricing System and JARVIS Data Warehouse.

# Business Understanding

Driving in America is a dangerous affair. In 2019 over 38,800 people lost, their lives in car crashes and another 4.4 million were severely injury due to car crashes according to the national safety council ([On](https://www.nsc.org/road-safety/safety-topics/fatality-estimates) The Road). The Stark Insurance company insures over 15 million Americans across 47 states. The company is interested in the ability to use Crash Report Sampling System (CRSS) data obtained from the National Highway Traffic Safety Administration (NHTSA) to predict the severity of injury of its insured members. Then using this data to create new pricing models and coverage processes, in regard to the medical coverage of its insurance members. This will help the company provide better value to its shareholders, customers, and medical community.

Value to Shareholders will come by increased accuracy in pricing the medical premium portion of car insurance to an individual vehicle and driver risk profile. This more accurate pricing will ensure the company is receiving an adequate return for covering its customers medical risk, thus increasing profit for the company. Value to medical community again comes from having adequate premiums as the insurance company will have more funds to pay for medical services of its members, so it likely will be able to reduce the amount of haggling with medical providers on rates. Finally, some customers will benefit from this project due to decreases in their premiums if their risk profile based off the completed project is rated low.

## 1.1 Project Team:

This project is important to Stark Insurance so the CEO Tony Stark will be the main project owner. The chain of command for the data-mining portion of the project will be as outlined in the diagram below:

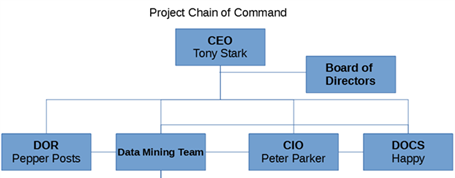
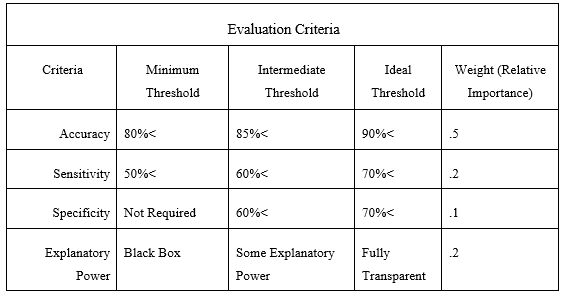


Figure .1.1

Pepper Potts is Director of risk for the company and will be the main contact for any questions regarding risk profiling of customers. Peter Parker is CIO and will be the one primarily responsible for taking the project from data mining project into a production system. The data mining team will not need to liaison with Peter until they are ready to start deployment of their model(s). Happy is the director of customer service and is an expert on Car insurance and the processes that go into analyzing current and potential customers and turning that analysis into an insurance premium. Happy has been in the car insurance industry for over 40 years. He will be the primary subject matter expert (SME) on the car insurance industry after Tony. The data mining team includes 4 people. They will be the team in charge of the data-mining project until the deployment stage. At deployment, the project will be handed off to Peter and his software development team and the data mining team will remain to function as an advisory team to the software development team and augment the software development team with data science skillsets.

## 1.2 Evaluation Criteria:

A project committee will decide if the project is a success. The project committee is made up of these members: Pepper, Peter, Happy, Tony and the Stark Board of Directors. As Tony has a majority stake in the company, the final decision-making authority is up to him regardless of the outcome from the committee. Success of the project will be in determined by meeting the criteria outlined in the chart below:



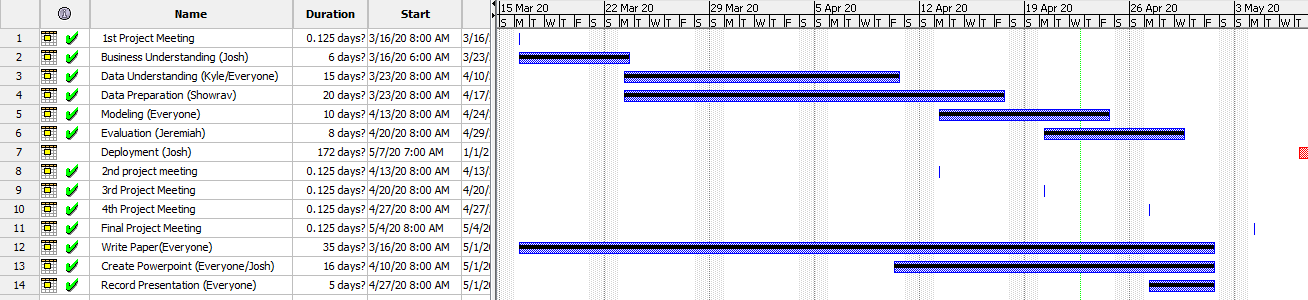
Failure to meet minimum criteria standards on any of the metrics will require further analysis from the project committee on proceeding with the project. If there are multiple models that meet the needs of the organization, the committee may choose at the end of the project to proceed with multiple models for deployment, one for prediction and another for evaluation of business practices (explanatory power).

Evaluation criteria overview: Overall accuracy is the most important factor in deciding the success of the project as the project needs to predict the severity of injury better than the company’s current methods, which result in an accuracy of roughly 75%. If the mew model cannot at least improve on the model by 5% than the expense of implementing the new model will not compensate for the savings/increased revenue generated by using the new model. Sensitivity is as important as Explanatory power because the company cannot have an algorithm that predicts too many false negatives. Files negatives represents a layer of financial risk to the company in the form of higher than predicted incidence of severe injury crashes, which would hurt the company’s bottom line due to higher than predicted medical payout to customers. False positives are not overly important to the company as this simply means we will end up charging customers more for their premium, even though they will likely not end up in a severe injury crash. However, if we continually charge customers too high a premium and another company ends up charging less, the company may start losing customers by being uncompetitive in the car insurance market. Thus, having a low false positive rate would be good, but not essential. Explanatory power is also important to Stark Insurance, as the company would like to use this data-mining project to improve some of its insurance processes surrounding what data to collect about its customers and customer’s vehicles in the future. Having a model that can show relative importance of variables in its decision-making will help the company shape current and future data and customer service practices.

## 1.3 Project Timeline

The project will follow the CRISP-DM data-mining methodology, which will split the project into six sections: business understanding, data understanding, data preparation, modeling, evaluation, and deployment.

The project started after the first meeting of the data-mining group, held on 3/16/20. The first five phases of the project, except deployment, are expected to be completed by 5/2/20. Deployments is expected to take at minimum 6 months after phase 5 is completed. The final timeline for phase six will be determined after at the final meeting for phase 5. The task breakdown is shown in the Gantt chart below:



# 2. Data Understanding

To analyze injury severity, we used data obtained from the National Highway Traffic Safety Administration (NHTSA). The data was divided into 4 tables: Accident, Distract, Person, and Vehicle.

* The Accident table provided data based on the accident. The primary key for this table is CASENUM and is the primary identifier for accidents.
* The vehicle table provides data based on the individual vehicle involved in the accident. The CASENUM variable links this record to the accident table. VEH\_NO identifies which vehicle in the accident this record identifies.
* The Person table identifies each individual in the accident. Each record has a CASENUM to link it to an accident, a VEH\_NO to link to the vehicle involved, and PER\_NO uniquely identifies the person in the vehicle.
* The Distract table identifies if the driver of the vehicle was distracted. The CASENUM, VEH\_NO, and MDRDSTRD are the unique Identifiers for this table. Our team did not find any value in using this table.

## 2.1 Data Selection and Variable Description

The goal of our team was to select variables that were known to the insurance underwriters at the point of determining premium pricing. This would only include pre-collision data. As well as data collected from previous accidents to help the underwriters to determine which vehicles have a history of accidents involving high severity injuries.

* REGION – This variable describes the region of the United States that the accident was involved in. 77 percent of all vehicle accidents occur within 15 miles or less of the person’s home. If a region has more accidents, then other regions the model can use this to predict higher injury severity
* URBANICITY – If the population is less than 250,000 it is considered urban, greater than 250,000 is city.
* MAKE – The manufacture of the vehicle the customer is looking to ensure.
* MODEL – The model of the vehicle.
* MOD\_YEAR – the year the vehicle was made.
* GVWR – Categorical variable that identifies what weight class the vehicle belongs to.
* ROLLOVER – Categorical variable that describes if the vehicle experienced 90 degrees or more of rotation. We included this variable because some vehicles like SUVs are more susceptible to rollovers than others.
* IMPACT1 – This variable describes the area of the vehicle that resulted in the first instance of injury.
* DEFORMED – This is a categorical variable that describes how badly damaged the vehicle is after the accident. It rates from no damage to disabling damage.
* FIRE\_EXP – This variable indicates if there was a fire from the result of the accident. We chose this variable to try and identify vehicles with a high fire risk like the Ford Pinto.
* SPEEDREL – This is a categorical variable that explains if the accident was the result of speeding or not.
* P\_CRASH2 – This variable describes the event that lead to the crash. This variable identifies what the vehicle was doing just prior to the accident and what made the vehicle’s situation critical.
* ACC\_TYPE – This describes the type of accident involved.
* SEX – Identifies the gender of the person
* AGE – Identifies the age of the person
* SEAT\_POS – Identifies where in the vehicle the person was located. We used this information to identify areas in the vehicle that are the most dangerous.
* REST\_USE – Binary variable that describes if a restraining system like a seat belt was used.
* EJECTION – Used to identify if the person involved was ejected from the vehicle due to the accident.
* AIR\_BAG – Identifies if an airbag was deployed and what type was used. We chose this variable to determine if vehicles with multiple airbags are safer than just one.
* ALCH\_TRANS – Derived variable that indicates if the driver was under the influence of alcohol at the time of the accident.
* DRUGS\_TRANS – Derived variable that indicates if the driver was under the influence of drugs at the time of the accident.

# 3. Data Preparation

The next step in the CRISP-DM process is the Data Preparation step. The key goal of this step is to remove any noisy data and prepare it for different modeling techniques. This section will be broken out into four parts: Data Filtering, Rule Engine (Transformation) Rule Engine (Data Binning), and Model Preparation. Figure 3.1 below provides a summary flow of the continued steps from the data selection and data understanding portion. An in-depth explanation will be provided along with a further understanding of this data flow. As mentioned in the prior sections our work was done with the open source tool KNIME and Microsoft Excel.

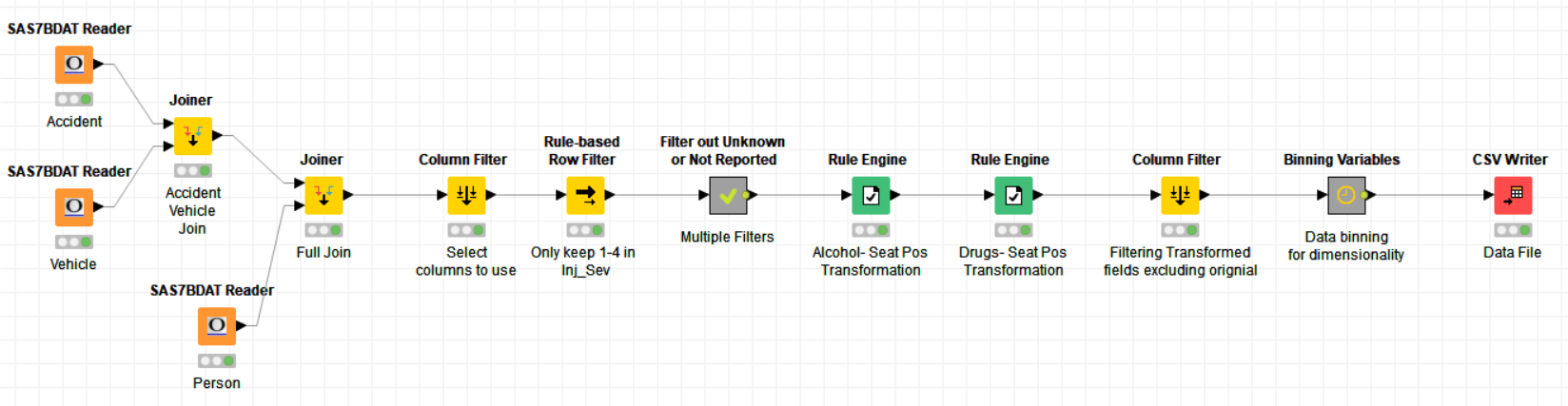


Figure 3.1

## 3.1 Data Filtering

The data-filtering step will apply a series of row filters that will help remove noisy data from our data set. The workflow can be seen in Figure 3.1.1 and can be seen in the workflow there is a meta-node called multiple filters. Figure 3.2.1 gives an extended view of this meta-node name multiple filters.

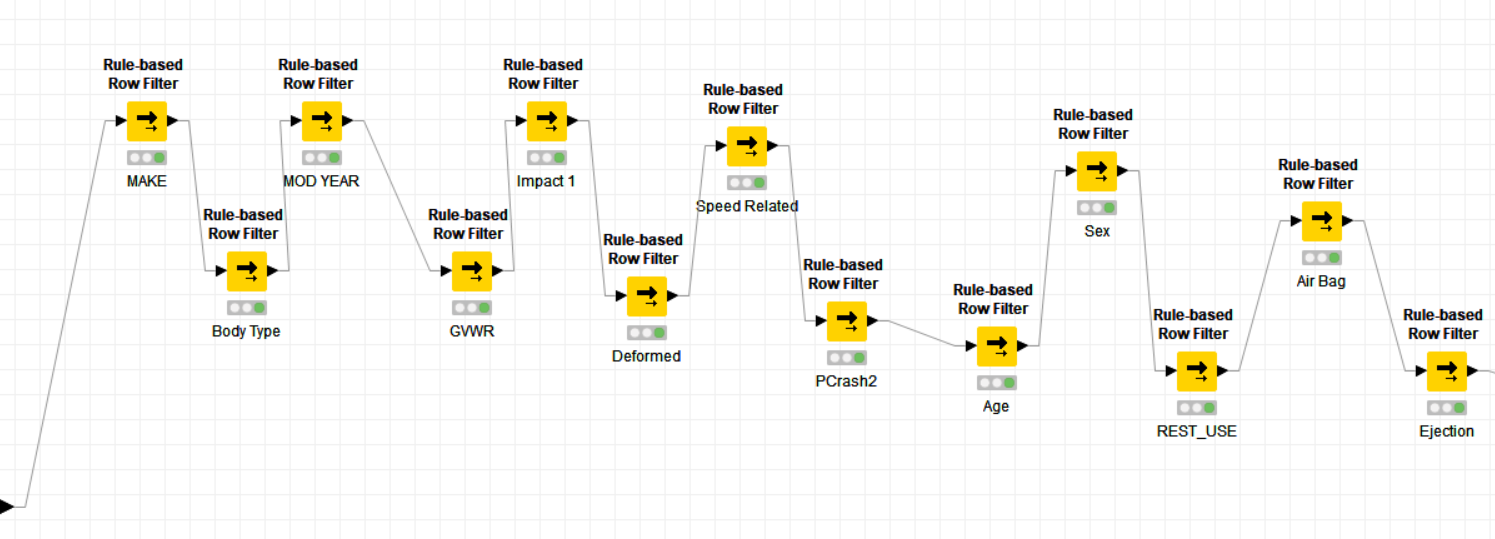


Figure 3.1.1

Starting back at Figure 3.1 we can see that there was an early row filter that were applied to the variables Inj\_Sev (injury severity). The injury severity filter applied was to keep only the injury severity classified from one through four. This decision was made as the other classes (0, 5, 6, 9) were classified as no injury, unknown, or death prior to injury.

Next Figure 3.1.1 goes through a series of row filters to remove unknown or not applicable classes in variables. For the variable Make the classes 97 (not reported) and 99 (unknown) were filtered out. For the variable Body\_typ (body type) the classes 98 (not reported) and 99 (unknown) were filtered out. For the variable Mod\_Year (model year) the classes 9998 (not reported) and 9999 (unknown) were filtered out. For the variable GVWR (gross vehicle weight rating) the classes 8 (not reported) and 9 (unknown) were filtered out. For the variable Rollover the class 9 (unknown) was filtered out. For the variable Impact1 (area of impact – initial contact point) the classes 98 (not reported) and 99 (unknown) were filtered out. For the variable Deformed the classes 8 (not reported) and 9 (unknown) were filtered out. For the variable Speedrel (speed related), the classes 8 (no driver present/unknown) and 9 (unknown) were filtered out. For the variable P\_Crash2 (critical event – pre crash) the class 99 (unknown) was filtered out. For the variable Age, the classes 998 (not reported) and 999 (unknown) were filtered out. For the variable Sex, the classes 8 (not reported) and 9 (unknown) were filtered out. For the variable Rest\_use (restraint system/helmet use) the classes 98 (not reported) and 99 (unknown) were filtered out. For the variable Air\_bag (air bag deployed) the classes 98 (not reported) and 99 (unknown) were filtered out. For the variable Ejection the classes 8 (not applicable) and 9 (unknown) were filtered out.

These filters will allow to be the initial step in removing noisy data. This will be one of the key steps to improving predictive models to help classify injury severity.

## 3.2 Rule Engine (Transformation)

The rule engine step will apply if-then rules to certain variables. As can be seen in Figure 3.1 there are two rule engine nodes applied in our data preparation step and after the first two rule engine nodes there is a column filter. Our first two rule engines are created for a specific seat position.

The first two rule engines were applied so that the fields alcohol and drugs were only applied to the seat position class 11, which is the driver per the dataset. To do this for alcohol the following steps were applied:

1. If seat position equal 11 and alcohol equals 1 then return 1
2. If seat position greater than 11 then return 0
3. If seat position less than 11 then return 0
4. If seat position equal 11 and alcohol does not equal 1 then return 0

A similar rule function as above is applied for the field drug. These two rule engine nodes created two new variables: Alch\_Trans and Drugs\_Trans. This step was completed to make sure that no vehicle had duplicate counts for alcohol or drug use related accidents. The concluding step for these two was to add a column filter to remove the original fields and only leave the newly transformed fields.

The rule engine steps allow for the creation of transformed variables that will improve model understanding and allow for better classification.

## 3.3 Rule Engine (Data Binning)

Much like the previous step the Binning Variables node is a series of rule engines. However, this step differs in the fact that it is not trying to reduce the count of records for a specific column but looks to address the issue of dimensionality. The issue of dimensionality being that when string type variables have to many classes’ underneath it. The diagram flow of this metanode can be seen below in figure 3.3.1.

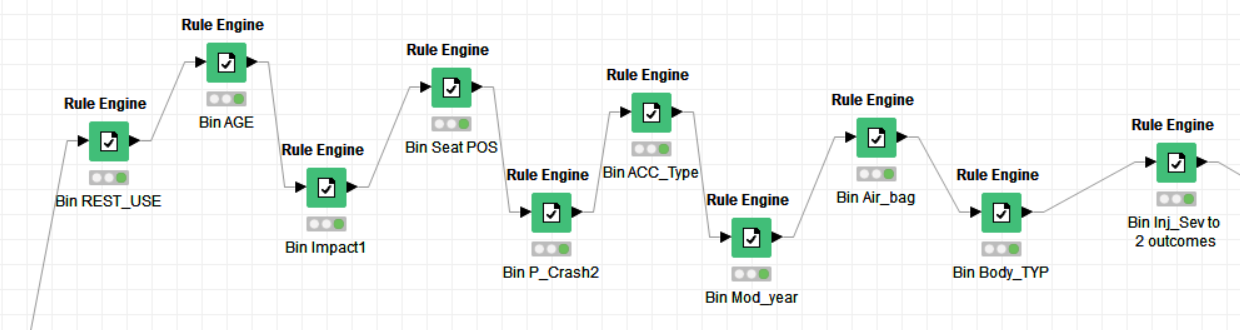


Figure 3.3.1

The first nine rule engine nodes are focused on the input variables and create new variables with a \_BIN label. As the above section the rules will be explained for each.

Rest\_Use (Restraint System/Helmet Use) bins so that the classes that are related to no helmet use are 0 and 1 otherwise, the rule applied was if variable equals 7, 16, 17, or 20 then 0 else 1. The Age bin was categorized as follows: 1 for less than 5 years, 2 for between 6 to 16 years, 3 for between 17 to 23 years, 4 if between 24 to 34 years, 5 if between 35 to 45 years, and 6 if greater than 46 years. The Impact1 (Area of Impact – Initial Contact Point) bin categorized as follows: if 0 (non-collision) then 0, greater than 0 (non-collision) and less than or equal to 12 (clock points then 1, and if greater than or equal to 13 (multiple sides of vehicle) than 3. The Seat\_pos (seat position) bin was categorized as follows: For 0 to 19 (front seat) 1, for 21 to 29 (second seat) 2, for 31 to 39 (third seat) 3, 41 to 49 (fourth seat) 4, and greater than 59 (cargo area) 5. The P\_crash2 (Critical Event-Precrash) bins was categorized as follows: less than 9 (vehicle loss of control) as 0, 10 to 21 (vehicle traveling) 1, 50 to 78 (other motor vehicle in lane) 2, 80 to 85 (non-motorist on road) 3, 87 to 92 (object or animal) 4, greater than 93 (other) 5.The Acc\_type (crash type) bin was categorized as follows: less than 16 (single driver) 0, 20 to 49 (same trafficway, same direction) 1, 50 to 67 (same trafficway, opposite direction) 2, 68 to 85 (changing trafficway, vehicle turning) 3, 86 to 91 (intersecting paths) 4, and greater than or equal to 92 (miscellaneous) 5. The Mod\_year(vehicle model year) bin is categorized as follows: greater than or equal to 2016 0, between 2008 to 2015 1, between 2001 to 2007 2, between 1990 and 2000 3, and less than or equal to 1989 4. The Air\_bag (air bag deployed) bin was categorized as follows: 20 or 28 (not deployed) 0, and all others 1. The Body\_typ (body type) bin was categorized as follows: 1 (convertible) 1, 17 (3-Door Coupe) 2, 2 to 6 (door count known) 3, 7 to 9 (door count unknown) 4, 10 to 16 (automobile derivatives) 5, 14 to 19 (utility vehicles) 6, 20 to 29 (van based light trucks) 7, 30 to 39 (light conventional trucks) 8, 40 to 49 (other light trucks) 9, 50 to 59 (buses) 10, 60 to 79 (medium/ heavy trucks) 11, 42 or 65 or 73 (motor homes) 12, 80 to 90 (motored cycles, mopeds, all-terrain vehicles, all-terrain cycles) 13, greater than or equal to 91 (other vehicles) 14.

The last rule engine is applied to the dependent variable injury severity. As it stands there are four classes (1, 2, 3, and 4) that remain for this variable. To provide some color to these classes: 1 is possible injury, 2 is suspected minor injury, 3 is suspected serious injury, and 4 is fatal injury. As we are looking to solve a classification problem, it is pivotal to reduce the class size to a binary variable for the dependent variable type. The rule engine does this by applying the below rules:

1. If injury severity equals 1 or 2 then 0
2. If injury severity is greater than or equal to 3 then 1

The above rules have now broken down the variables to 0 (low injury severity) and 1 (high injury severity). Figure 3.3.2 below provides a bar chart to provide a count of each class for the dependent variable.

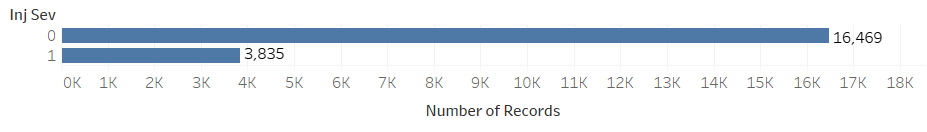


Figure 3.3.2

This rule engine step help reduce the dimensionality of our string variables into more This concludes with a csv file which will be used for model preparation and modeling.

## 3.4 Model Preparation

Our final phase in the data preparation step is making the data ready and prepared for modeling. Figure 3.4.1 below provides the walkthrough that will be discussed in this section. Figure 3.4.2 provides an idea of the number for levels in our nominal variables, if there are any missing values, and provides the top two most recurring classes. Figure 3.4.3 gives a graphical representation of the distribution of each class in each of the nominal variables.

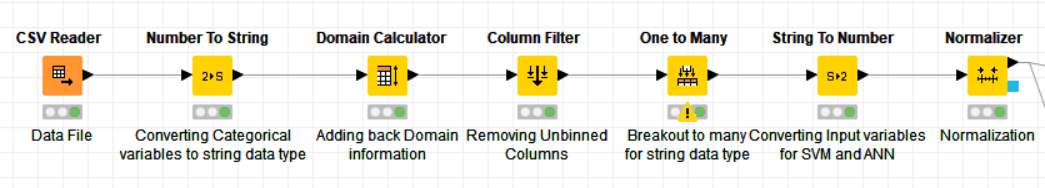
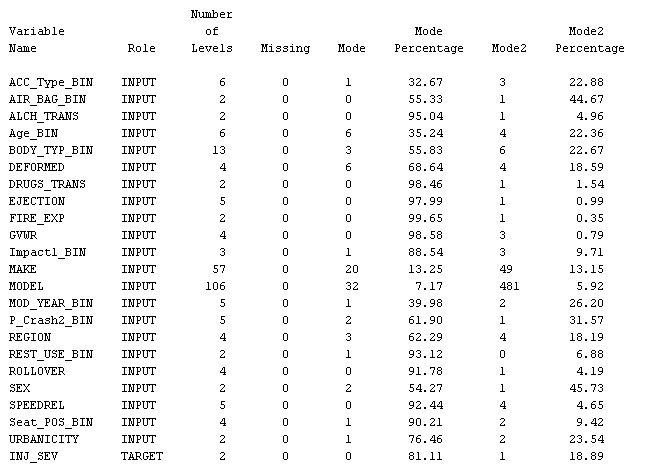
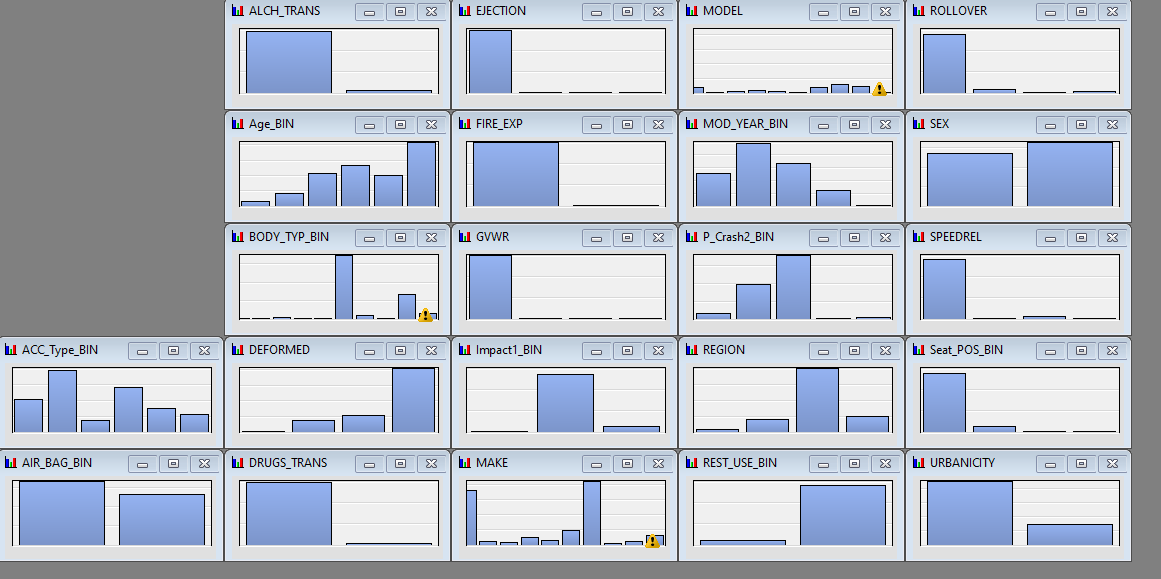


Figure 3.4.1

 Figure 3.4.2

 Figure 3.4.3

The initial step is taking all the variables and making sure that all variables are once again converted to string type variables (as they dataset has them as numbers, but as described in the data understanding step these are nominal variables). We then must use a Domain Calculator node as there is potential information loss with the amount of filtering that was applied in the previous data preparation steps. Next, we must remove the original variables that were binned in the last data preparation step and only keep the new binned variable. For a classifier model this would be the final step, however, three additional steps are needed for a number loving model. The first is the one to many steps, which takes each class of a categorical variable and puts a 1 if that column populated and 0 if not. We then take all but our dependent variable and convert it to a number variable type. The last step is the normalizer step, which takes any number variable and puts it between 0 and 1 for our input variables. With these steps complete the modeling can begin!

3.5 Variable Validation

The team used the decision tree model not only for predictive modeling of injury severity but also to validate that the independent variables were not direct copies or derivatives of the dependent variable INJ\_SEV. The way this validation was accomplished was by running an un-optimized decision tree model on the chosen data set to see if the model resulted in a shallow tree or a well formed tree with depth. A shallow tree generally means you have a variable that is a copy or derivative of the dependent variable, while a fully formed deep tree generally means none of the variables are copies or derivatives of the dependent variable.

The pre-binned data set that was not used by the team resulted in a wide but shallow tree. However, after experimentation with removing variables from the data set and binning variables with a large number of categories it was found this initial shallow tree was simply due to one age having too many categories and not age being a copy or derivative of INJ\_SEV. Once the team built a data set that binned Age into a smaller number of categories the decision tree model created a fully formed wide tree with 8 layers. It was at this time the team decided to use a binned data set to simplify decision tree and random forest modeling and to hopefully achieve a tree that was not as wide due to many variables having a large number of categories. The final binned dataset resulted in a fully formed decision tree, which was still fairly wide with 8 layers. The team is confident there are no independent variables in the data set that are copies or derivatives of the INJ\_SEV dependent variable.



*Figure 3.5.1 Fully Formed Un-Optimized Decision Tree model from Binned Data Set (snippet)*

# 4. Modeling

Each member of the team decided which model they felt like would produce the best result to accomplish our goal. The 4 models are Decision Tree, Random Forest, Support Vector Machine, and Artificial Neural Network. The Decision Tree and Random Forest are categorical based models. The Support Vector Machine and Artificial Neural Network are number based models. To accurately evaluate each model, we determined that we would use the same random seed and partitioning for every model. Each model will utilize the 70/30 train test split for validation purposes. The Random Forest model and the SVM model will also use the 10-Fold cross validation as a redundant verification.

## 4. 1 Decision Tree

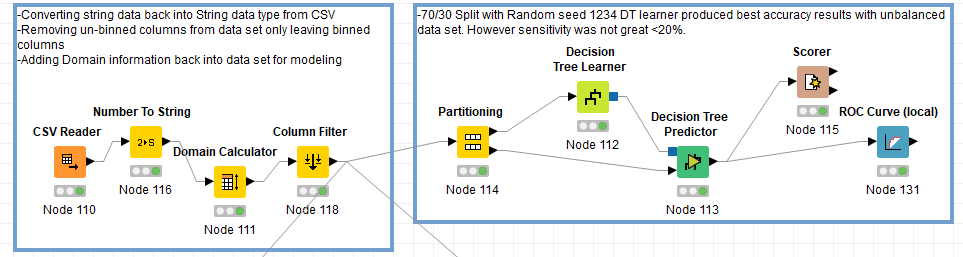


Figure 4.1.1

The decision tree (DT) model, while usually not the optimal model for prediction of categorical data sets, is useful in ascertaining if one or more of the independent variables in a data set are derived or duplicates from the dependent variable. The test used to determine this factor is the depth of the DT. If the decision tree is shallow (less than 3-4 levels), then it is likely that one or more independent variables is a derivative form of the dependent variable and needs to be removed from the data set. If the DT is fully formed and deep (>4 levels) then there likely are not any derivative independent variables based from the dependent variable. The data mining team decided to bin some of the categorical variables to simplify modeling of the data set and final deployment. The binned data set created a fully formed DT, so the team is confident none of the independent variables selected are derivatives of the dependent variable INJ\_SEV as shown in the DT models below:

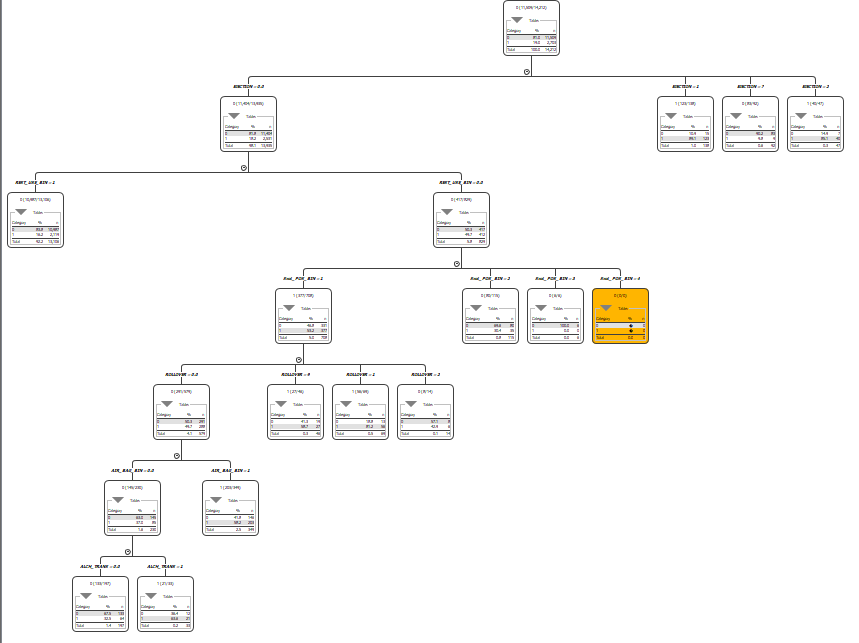


Figure 4.1.2 Optimized final Decision Tree Model



Figure 4.1.3 Un-optimized Decision Tree from Binned Data Set (Snippet)

The team attempted to use multiple splitting methods and modeling parameters to optimize performance of the DT model. The best DT model in terms of accuracy resulted from a simple 70/30 Training/Test split with model parameters set to:

* Quality Measure: Gain Ratio
* Pruning Methods: MDL and Reduced Error Pruning
* Min # Record Per Node: 20

This resulted in an accuracy of 83.569%, a sensitivity of 19% and specificity of 98%. Unfortunately, the sensitivity of the model is not good enough for the model to move forward in selection. Attempts to improve sensitivity using stratified sampling did not significantly improve sensitivity and while SMOTE sampling increase sensitivity it decreased accuracy to only 76%, which is not high enough to move forward in the model selection process.

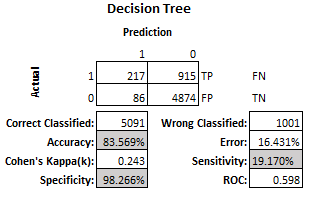


Figure 4.1.4 Decision Tree Performance

## 4. 2 Random Forest

The goal of the random forest model (RF) is to average multiple decision trees to reduce the variance. This model reduces the habit of overfitting the training data that decision trees are known for. Figure 4.2.1 shows the entire model from data input to evaluation.

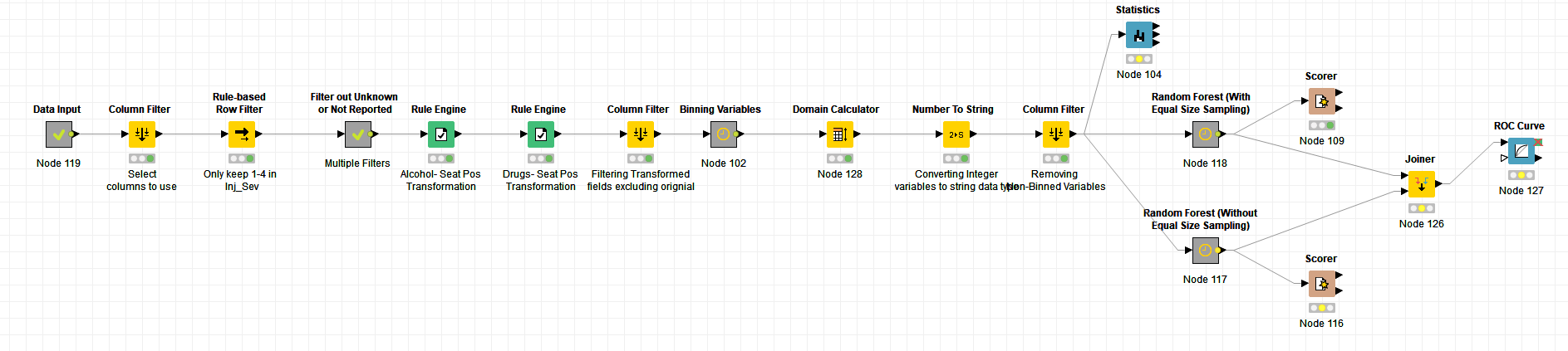


Figure 4.2.1

For the RF model we evaluated using the equal size sampling node and without the node. Figure 4.2.2 shows the model with this node and Figure 4.2.3 shows the model without the node.

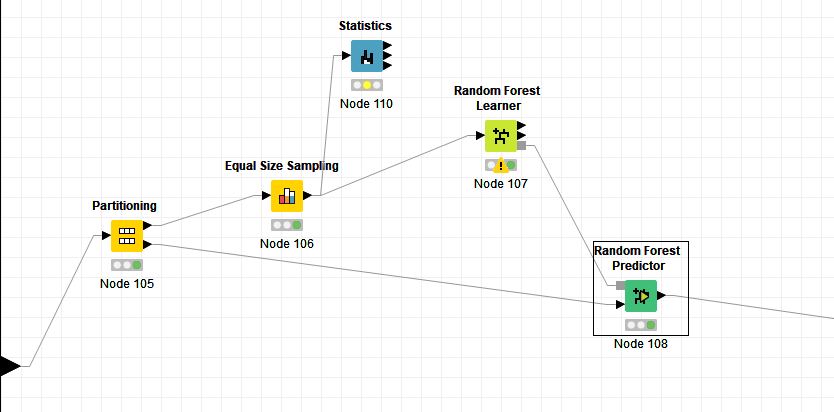


Figure 4.2.2

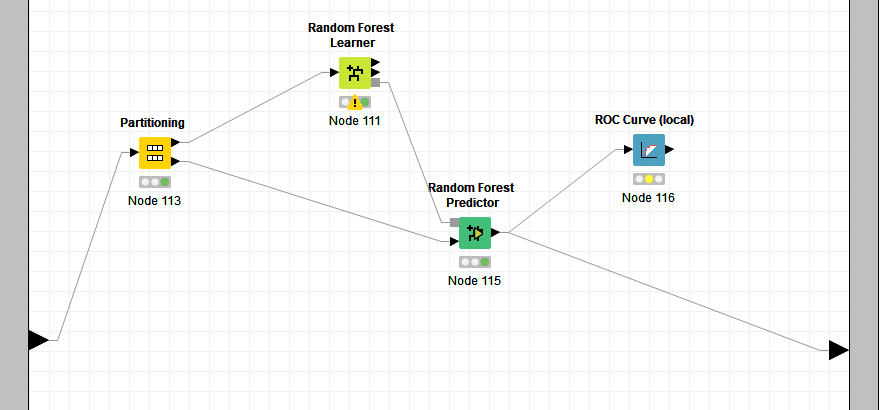


Figure 4.2.3

The settings used for the model are:

* Minimum node size: 5
* Split Criterion: Information Gain Ratio
* Number of Models: 800
* Random Seed: 12345

Without using equal size sampling the RF model produced an accuracy of 83.290%. But the specificity of the model is only 17.933%. This model does produce a lot of false positives, meaning that customers who should get a discount rate will be charged like a customer who is higher risk.

With equal size sampling the accuracy of the model is 73.211%. The specificity has increased to 60% and the sensitivity is 76%. This is a more balanced model but the accuracy is below our ideal goals. We utilized K-Fold Cross validation using 10-fold validation to verify our findings for this model. We discovered that the accuracy was not much different with 70/30 split than the 10-fold validation. Figure 4.2.4 shows the 4 RF models compared.

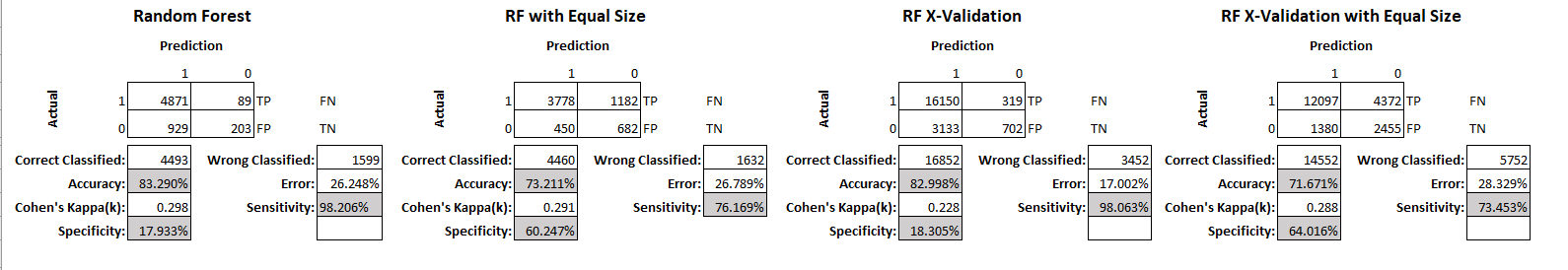
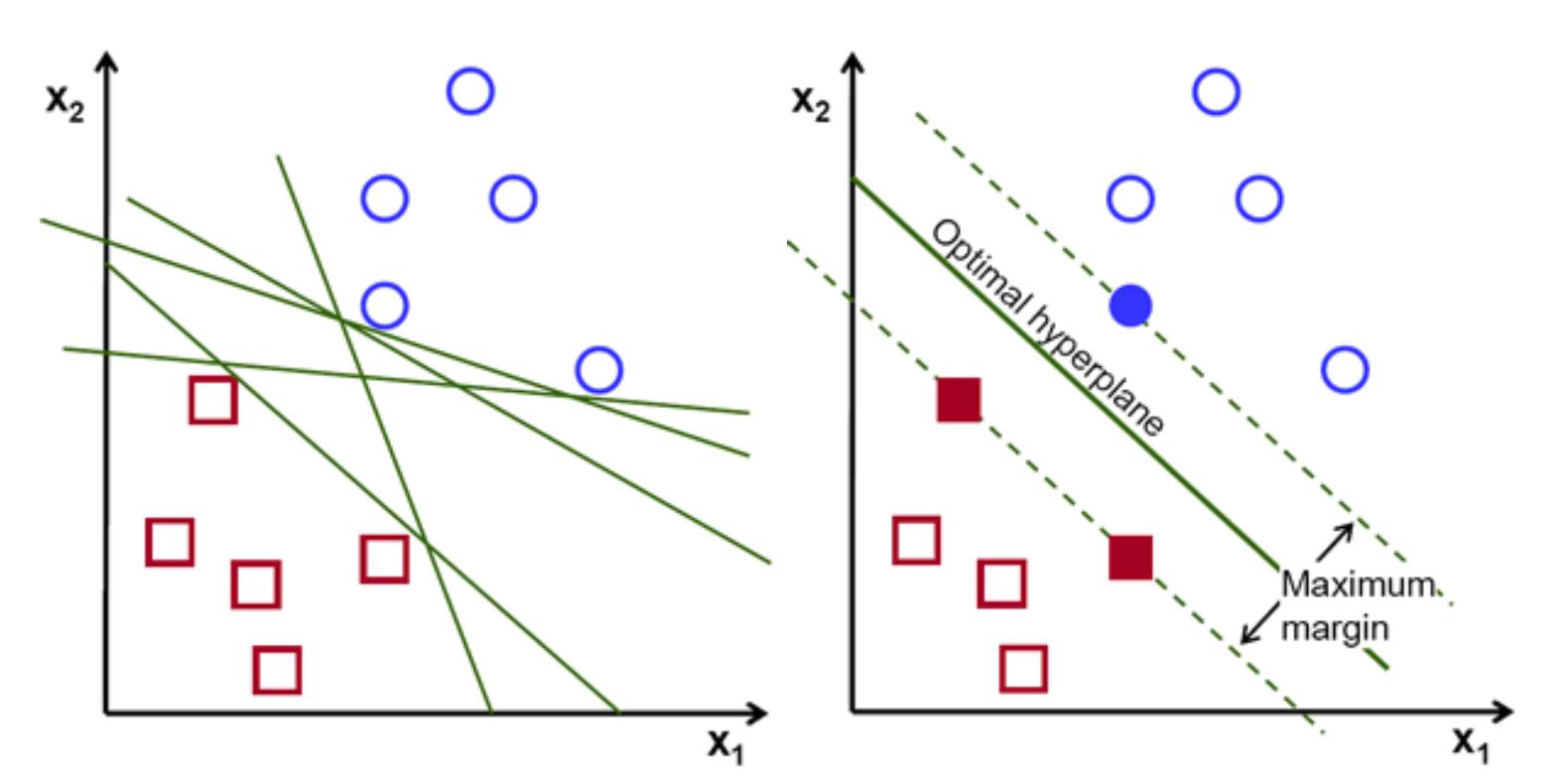


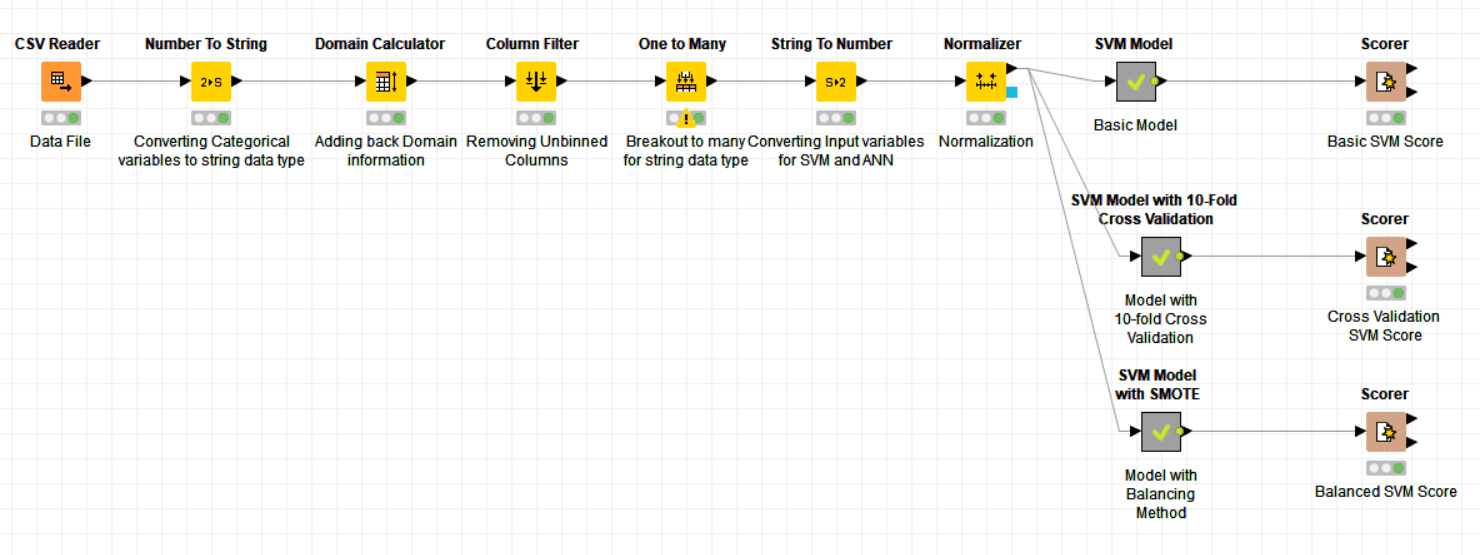
Figure 4.2.4

The Random Forest model will be compared to the other models to determine if it’s the best model for our organization to use.

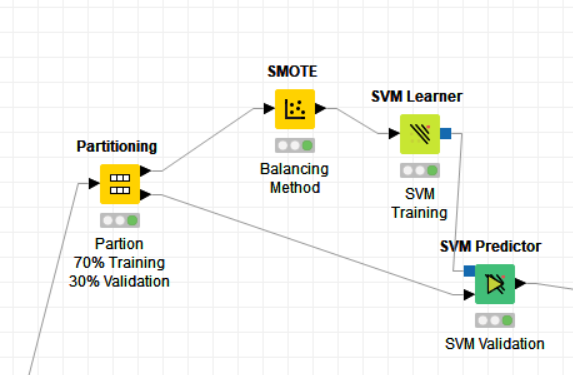
## 4. 3 Support Vector Machine

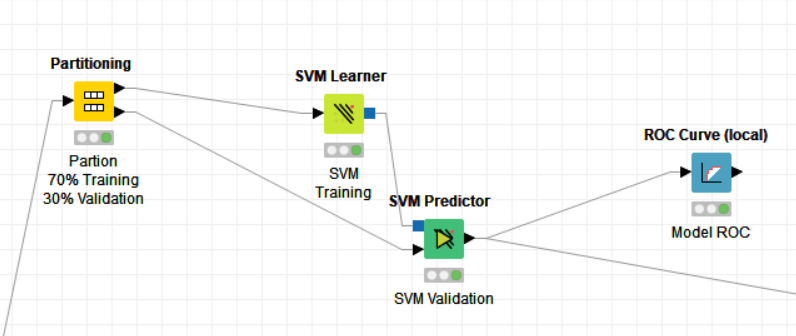
A support vector machine (SVM) is a supervised learning model that has the objective to find a hyperplane in a n-dimensional space that distinctly classifies datapoints. SVM’s can be used to solve regression and classification problems. A visual representation of SVM can be seen in Figure 4.3.1 below. The predictive modeling workflow used by our team to produce the SVM can be seen in Figure 4.3.2.

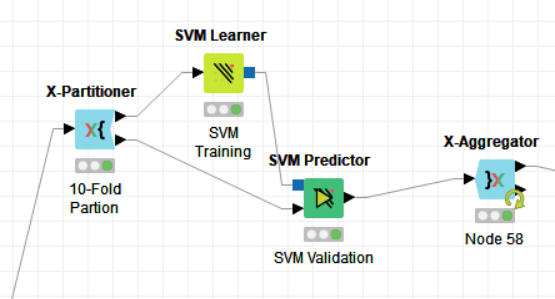
 *Figure 4.3.1*

 Figure 4.3.2

For the SVM model we evaluated using the SMOTE node and without the node. Also, on the model without SMOTE we performed 10-fold cross validation. SMOTE was applied to oversample the minority class (1). Figure 4.3.3 shows the model with this node, Figure 4.3.4 shows the model without the node, and figure

 *Figure 4.3.3*

 *Figure 4.3.4*

* Figure 4.3.5*

The settings used for the training model are:

* Overlapping Penalty: 1.0
* Kernel: RBF
* Parameter (sigma): 0.1
* Random Seed: 12345

The model with and without SMOTE, produced similar results. The accuracy was 80.630%, the sensitivity is 2.954%, and specificity of 98.725%. By oversampling the minority class our sensitivity and specificity did not see improvements. The results can be seen in Figure 4.3.6 below. After performing 10-fold cross validation on the basic model without SMOTE, we did not see significant improvements with an accuracy of 80.575%, sensitivity of 3.207%, and specificity of 98.591%. The results can be seen in Figure 4.3.7 below.

 *Figure 4.3.6*

 *Figure 4.3.7*

While the model was able to identify low severity injuries significantly well it did not meet the goal of identifying high severity injuries and potentially should not be used for deployment.

## 4. 4 Artificial Neural Network

Neural networks, while inspired by the biological brains of humans, are beyond exact simulation due to their large size. As a result, data scientists use the Artificial Neural Network (ANN). This contains representations of biological entities such as synapses, axons, and dendrites. ANN contains at least one hidden layer of processor elements that perform the model’s work.

Our team chose to use RProp MLP (MultiLayer Perception) model for our ANN due to it being the best fit for tabular datasets and classification predictions problems compared to other ANNs such as PNN.

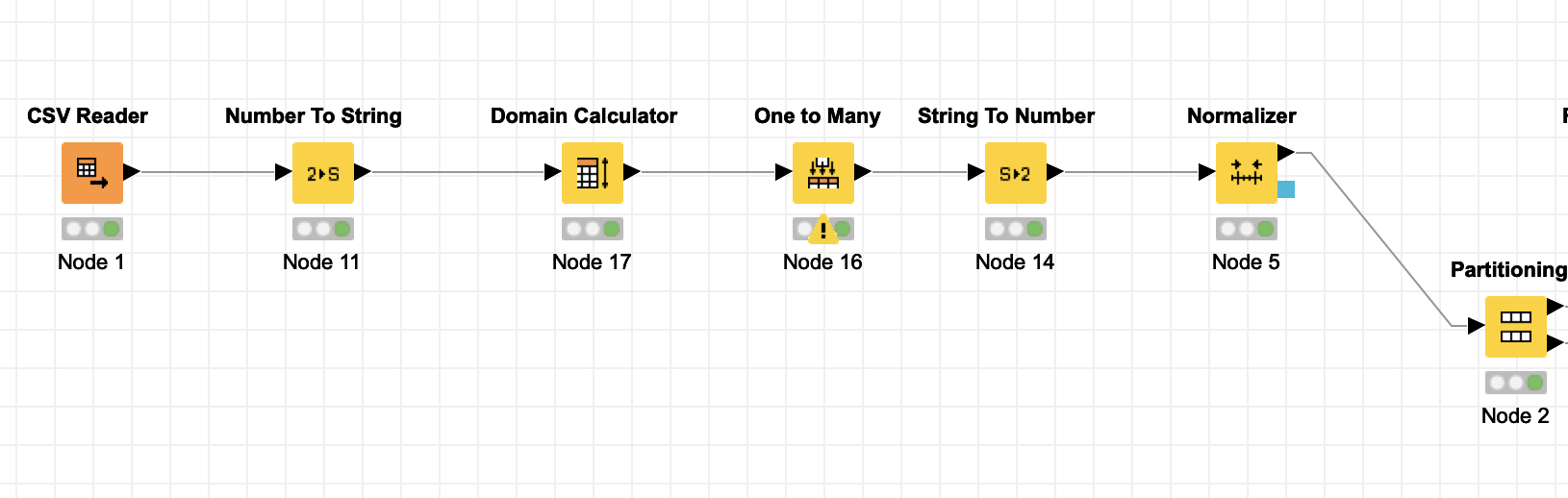


Figure 4.4.1

Before we could process the data for our model, we had to make the data numeric in order to be compatible with MLP (Figure 4.4.1). We want to convert classes into many columns for each value. In order to do this, the data would have to be converted to strings and go through a domain calculator for Region, Model, ACC\_Type, and Age. Once we did the One to Many conversion, the data is converted back into numeric types and normalized.

The partitioning is on a 70/30 split that draws randomly with a seed of 12345. Experiments were done with stratified sampling, but this did not make any difference in the outcome.

For our Learner, we chose the following properties:

* 100 Maximum number of iterations
* 1 Hidden layer
* 10 Hidden neurons per layer

The results of the model without SMOTE (Figure 4.4.2) were not amazing but turned out better than first expected. The model resulted in an 80.68% accuracy with a 90.26% sensitivity and 35.33% specificity.

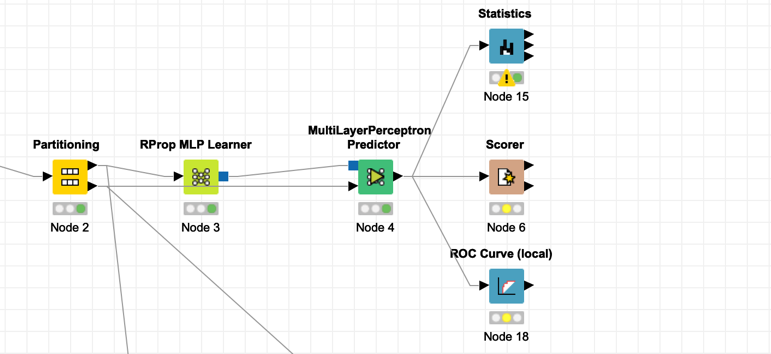


Figure 4.4.2

The results of the model with SMOTE (Figure 4.4.3) were slightly worse than without. The model resulted in a 76.9% accuracy with an 83.57% sensitivity and 44.7% specificity. The results can be seen in Figure 4.4.4.

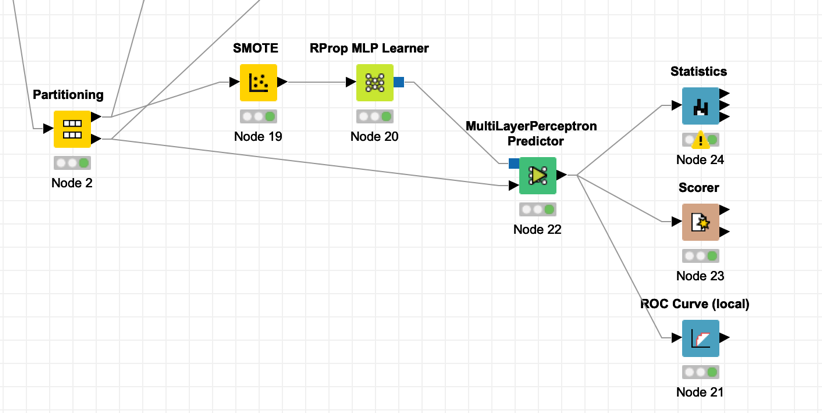


Figure 4.4.3

Results:

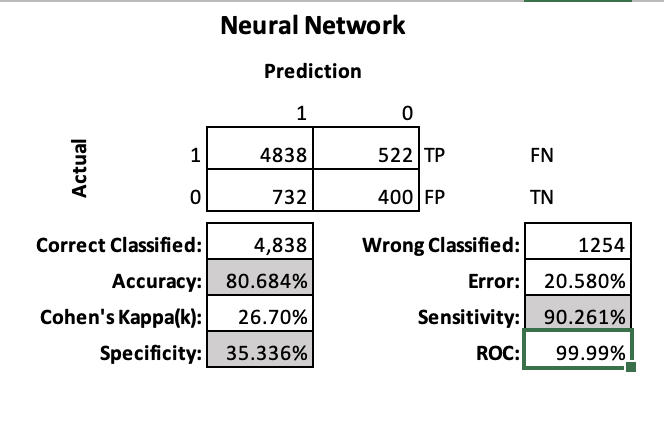
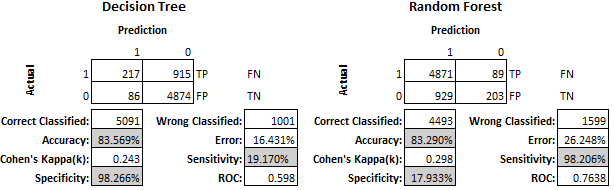
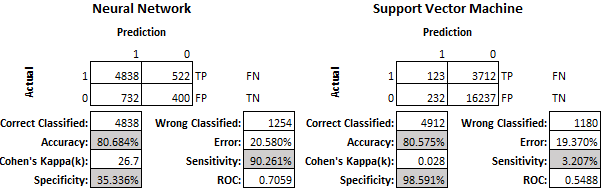


Figure 4.4.4

# 5. Evaluation

In our evaluation, we have the following results:

****

****

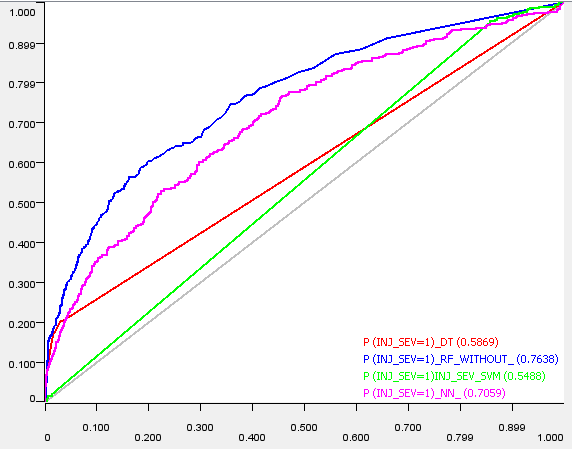


Figure 5.1.1 Combined ROC Curves for DT, RF, SVM and ANN

As we can see from the numbers above, the best model is Random Forrest. This model does produce a lot of false positives, meaning that customers who should get a discount rate will be charged like a customer who is higher risk. However, for our insurance purposes, charging more for some people versus undercharging others is a better outcome.

## 5.1 Variable Dependence

We utilized the Random Forest Learner node to look at variable dependence. Figure 5.1.1 shows the output of the attribute statics.

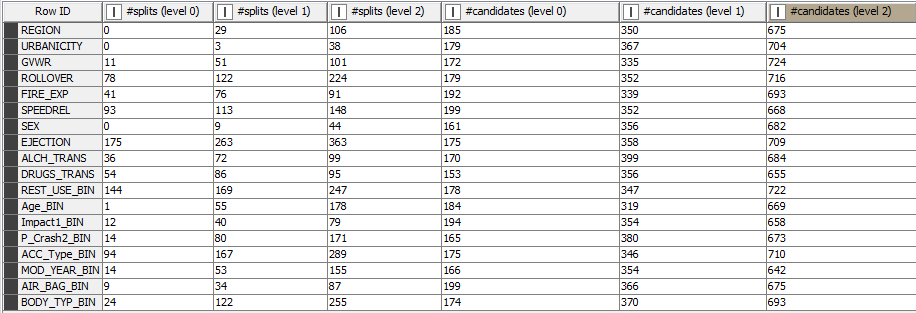


Figure 5.1.1

There are a total of three levels in this model. We used the formula Score = (Splits level 0 / candidates level 0) + (Splits level 1/Candidates level 1) + (splits level 2/candidates level 2) to create a score column. Figure 5.1.2 shows the output with the Score column sorted descending.

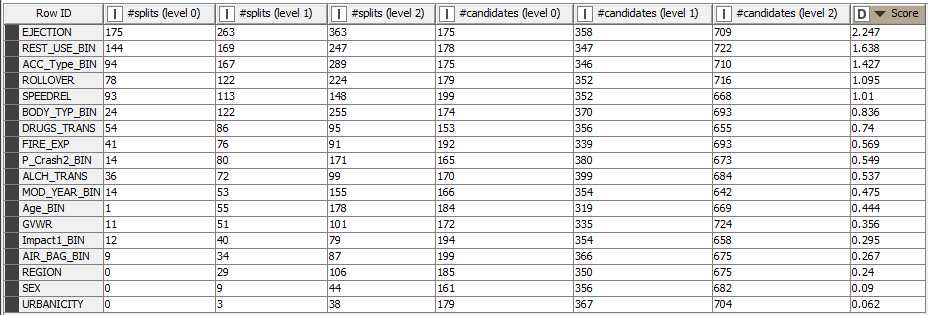


Figure 5.1.2

From this figure we can see that Ejection is the most important variable, followed by Rest\_use. Our knowledge of accidents would tell us that since Ejection is so important to injury severity, that wearing seat belts reduces injury severity. The least important variables are Urbanicity and Sex. It really doesn’t matter where you are or your gender when determining the severity of the injuries. Urbanicity is important for considering frequency of accidents. In our dataset there were 15,525 accidents in the City and 4779 accidents in rural areas.

From this knowledge, we can see that the most important factors are determined from the crash itself. The least important variables are the knowledge beforehand information. Like vehicle information, age and sex of the driver, do not give much evidence of injury severity. They do give indications of frequency of accidents. Figure 5.1.3 shows the histogram for age. From this we can see that the younger groups have more accidents then the older groups. This information still assists us with determining premium pricing.

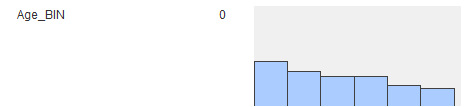


Figure 5.1.3

In Conclusion we can determine the severity of a crash injury with 83 percent accuracy. This model meets all of our goals that we set when we started this project. It will be a great tool for our insurance underwriters to use to help determine premium pricing.

# 6. Deployment

Since the Random Forest model met the minimum standard for all evaluation criteria to move forward into production the Stark Insurance Board of Directors and CEO Tony Stark voted to move forward with the data-mining project and to work on deploying the Random Forest model into Stark Insurance IT production environments. Peter Parker CIO and three application development teams will work on integrating the Random Forest Model into the following Stark Insurance systems:

• Premium Pricing System (Sets pricing for auto insurance premiums)

• Salesforce CRM system (used for sales and customer tracking/interactions)

• JARVIS data warehouse (used for analytics, reporting and long-term storage)

Work on all three integrations will happen simultaneously as the Premium Pricing System and JARVIS data warehouse both must integrate with the Salesforce CRM system in order to function properly. The data-mining team will now be under the direction of Peter and his application development teams and will act as subject matter experts on the Random Forest Model and data science applications. The deployment phase of this project will begin on 5.7.20 and will at minimum take at least 6 months. A better estimate on the deployment timeline will take place on 6.2.20 after the deployment teams have met once or twice and have a better idea of the risks and resources required to complete the integrations.

# References

*On The Road*. www.nsc.org/road-safety/safety-topics/fatality-estimates.

Insurance, Allstate“Insurance Quote.” *Allstate Online Shopping*, purchase.allstate.com/onlineshopping/warmup/1.

Insurance, Farmers. “Insurance Quote.” *FARMERS FAST QUOTE*, esales.farmers.com/fastquote?LOB=A.

Insurance, GEICO. “Insurance Quote.” *GEICO*, auto-buy.geico.com/nb.