MSIS 5633 – Predictive Analytics Technologies

(Online Section)

Group Term Project – Fall 2020

Prediction of Car Crash Data

Due Date

December 10, 2020

By

Abhishek Bhale

Divya Peddapeta

Philip Anderson

Shweta Parida

Table of Contents

Executive Summary

For this project, our group's task was to determine the injury severity of a car crash based on the datasets we were provided. The datasets that we were given to work with were Accident, Vehicle, Person, and Distract datasets. We used the KNIME analytics platform in order to create our data mining models. The Primary Keys of data for the Accident dataset is Case Number, for Vehicle data, it is Case number and Vehicle number and for the Person dataset, it is Case number, Vehicle Number, and Person number. Then there is a variable that determines how severe the accident was on a scale of 1 to 4 and the rest were unreported. Between all 4 data sets, there are a total of 22,960,278 unique records. Our only target variable (dependent variable) is Injury Severity and all others can be considered as predictor variables. Out of 115961 rows and 198 columns we filtered them down to 7181 rows and 22 columns. We are to analyze all factors of the car crashes in order to determine which variables impact the injury severity the most. We created different classification models in order to help us find out more about each variable's correlation to injury severity. The list of column variables we used are Number of Total Motor Vehicles, Number or Persons in Motor Vehicle Transport, Month of Crash, Day of Week, First Harmful Event, Manner of Collision, Relation of Junction (Specific Location), Type of Intersection, Light Condition, Atmospheric Conditions, Number Injured in Crash, Alcohol involved in Crash, Vehicle Model Year, Vehicle Body Type, Travel Speed, Extent of Damage, Roadway Surface Condition, Traffic Control Device, Critical Event Pre-crash, Age of Person, Seat Position, Air bag and Injury Severity. We created different data mining models by using the CRISP-DM process to analyze the entire data set. CRISP-DM is a 6-step process that includes steps in the following order Business understanding, Data Understanding, Data Preparation, Data Modeling, Evaluation, and Deployment. Preprocessing of the data is required in order to understand which columns variables should be used as predictor variables and be applied to which specific decision model in order to give the most accurate results. The data mining models we used were Decision Tree, SVM, Random Forest, Neural Network (MLP), Logistic Regression, and KNN.

CRISP-DM Process

1. Business Understanding

The problem that we are trying to solve is predicting the injury severity in car crashes. We want to find the level of injury severity that people suffer from when they are involved in a car crash. We are trying to solve this problem based on the accident data, the vehicle data, and the persons data. Our Dependent variable is in numeric format. It has values from 0 to 9, where 1, 2 value stands for low injury severity and 3, 4 value stands for high injury severity. All the other variables are either not reported or show no injury. Hence, to convert it into a binary format we are grouping the values 1 and 2 as low, and 3 and 4 values as high. Now, the dependent variable is in binary string format and can be used directly for set-based models like Decision Tree and Random Forest. For the number-based model, we will convert it into binary numeric variables 1 and 0 using the Rule Engine.

2. Data Understanding

We have 4 SAS data files - Accident, Distract, Person & Vehicle, that give us the details of car crashes that took place in the US for the year 2018. The datasets are a combination of different data types. Following are the different data files that we have and their details:

The Accident Data file has 51 columns with 48443 records, which gives information like the total number of vehicles in the accident, the number of persons involved in an accident, the time during which the accident happened, weather & light conditions, manner of the collision, harmful events, and other related variables. Whether there was alcohol involved in the crash, the location of the car relative to the junction. Here, the primary key is CASENUM (case number).

The Vehicle Data file has 87 columns with 86105 records, which give information about vehicle details like its age, type, model year, surface condition of the road, traffic control devices, and other related data. Here, the primary keys are CASENUM (case number) & VEH_NO (Vehicle number).

The Person Data file has 54 columns with 130230 records, which provides details regarding the people involved in the crash like their age, seat position, the severity of their injuries, and other similar data. Here, the primary keys are CASENUM (case number), VEH_NO (Vehicle number) & PER_NO (PersonId).

The Distract Data file has 11 columns with 86131 records, which includes information related to the distraction due to which the driver caused the accident and the location of the incident. Here, the primary keys are CASENUM (case number) & VEH_NO (Vehicle number).

The data related to different crashes in America for the year 2018 is captured. It includes data on the accident, the details of the person and the details of the vehicle. The data that is captured here is directly from an item on the police crash report or by interpreting the information that was provided in the crash report. It was obtained from the review of the crash diagrams, the police officers' written summary of the crash or with combinations of data elements on the report.

3. Data Preparation

The 4 data files, Accident, Distract, Person & Vehicle are of SAS file type. In order to read the data to the table in the database in KNIME tool we have used SAS7BDAT Reader. The four SAS data files now needs to be merged to form a single consolidated data set.

The **Joiner** node is used to join the tables in the database-like way. The Accident and Vehicle tables are joined with right outer join mode with primary key ‘CASENUM’. Then the merged data Person data is added with Left outer join with the primary keys ‘CASENUM’ and ‘VEH_NO’. The merged data of the 3 datasets are then combined with a distract dataset using a joiner node with left outer join mode with primary keys ‘CASENUM’ and ‘VEH_NO’.

Column filter is used to remove the unwanted columns and the imputed columns. After removing the unwanted columns, values attributing to the target variable needed to be included in the dataset.

A **Row filter** has been used to select the dependent variable which is INJ_SEV ranging from 1 being the low degree of injury to 4 being the fatal injury.

A preprocessing node **Numeric Binner** has been used to categorize the numbers into 2 categories high and low. After setting the bin median as 2.5 based on the number of attribute codes. This combines the 1 and 2 as low level and 3 and 4 value as high level. A new column is created and appended to the original dataset showing the newly binned attribute values as Low & High.

There are few values in each of the columns which indicate Unknown, Not Reported, Reported as Unknown. These are the values that are present in the data elements: HARM_EV, DAY_WEEK, RELJCT2, TYP_INT, NUM_INJ, BODY_TYP, DEFORMED, VSURCOND, P_CRASH2, AGE, ALCOHOL, MAN_COLL, LGT_COND, WEATHER, TRAV_SP, MDRDSTRD and VTRAFCON. A preprocessor node **Rule-based row filter** has been used where multiple matching rules or expressions are defined as TRUE & FALSE matches to filter out these indicated values. So, the filtered table contains all known and reported values, necessary for building the predictive models.

Driver factor has been considered important, as it contributes to the risk of occurring crashes, leading to injuries. A **Row filter** has been used to capture driver data. The variable SEAT_POS (attribute value = 11) is used to capture the driver's data. This is the seat position of the driver in the given data.

We have categorized the Month variable into seasons as Fall, Winter, Spring and Summer, and Days variable into weekdays and weekends using **Rule Engine**.

To know if the car is worn out as it increases the chances of getting into an accident or breaking down in the middle of the road, causing an accident, Vehicle age is derived. The age of the vehicle is calculated using the **Math Formula** (2018 - MOD_YEAR).

New variables or columns are created that contain the derived data. We need to remove the unwanted columns that were left in the model after applying the rule engine, rule-based row filter,

math formula, and numeric binner. So, to remove processed & unwanted variables, a **column filter** is being used, and all the clean data is fed to our model for prediction.

We have used the **Normalizer** node to convert the numeric values between 1's and 0's. This type of normalized data is required by the Logistic Regression model and the ANN model. This node transforms the numeric columns linearly with min-max normalization with minimum value 0 and maximum value to 1.

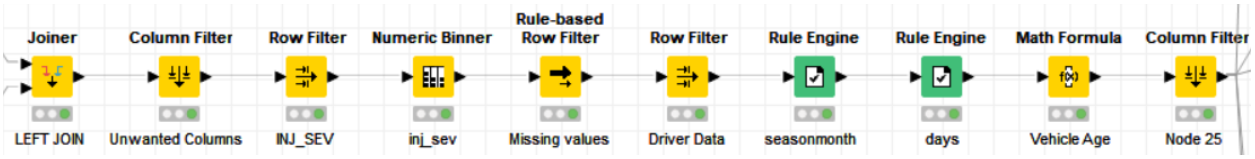
This **Partitioning** node is used to split the data for training and testing purposes with a ratio of 70% to 30%.

In the **K-Fold** cross validation, each row in the input is used in K-1 iterations for training the model and in only one iteration for prediction.

The **X-Partitioning** node is used to split the data into 10 partitions (folds).

The **X-Aggregator** node is used to combine the data from the learner and the testing data.

These are the following nodes that were used to preprocess the data before feeding it to the models:



Finally, we built different models by first selecting around 40 variables that we thought could be important for modeling. Then we started removing certain columns and fed it to our model. By doing this we checked how different variables impacted the accuracy of the model. This Trial and Error method led us to using these variables for our models. Following are the variables that were fed to our model:

Sr. no.	Variables	Description
1	VE_TOTAL	This data element is the number of contact motor vehicles that the officer reported on the police crash report as a unit involved in the crash.
2	MONTH	This data element records the month in which the crash occurred.
3	DAY_WEEK	This data element records the day of the week on which the crash occurred.
4	HARM_EV	This data element describes the first injury or damage producing event of the crash.
5	MAN_COLL	This data element describes the orientation of two motor vehicles in-transport when they are involved in the "First Harmful Event" of a collision crash. If the "First Harmful Event" is not a collision between two motor vehicles in-transport it is classified as such.
6	RELJCT2	This data element identifies the crash's location with respect to presence in or proximity to components typically in junction or interchange areas. The coding of this data element is done in two sub-fields (see also C21A) and is based on the location of the "First Harmful Event" of the crash.
7	TYP_INT	This data element identifies and allows separation of various intersection types.
8	LGT_COND	This data element records the type/level of light that existed at the time of the crash as indicated in the police crash report.
9	WEATHER	This data element records the prevailing atmospheric conditions that existed at the time of the crash as indicated in the police crash report.

10	NUM_INJ	This data element records the number of persons injured in the crash and is derived by counting all persons with “Injury Severity” of (1, 2, 3, 4, or 5) in the crash. This count includes fatally injured occupants.
11	ALCOHOL	This data element records alcohol use for drivers, pedestrians, cyclists and other types of non-motorists (except occupants of motor vehicles not in-transport) involved in the crash. The data element is derived from “Police-Reported Alcohol Involvement” in the Person data file.
12	MOD_YEAR	This data element identifies the manufacturer's model year of this vehicle.
13	BODY_TYP	This data element identifies a classification of this vehicle based on its general body configuration, size, shape, doors, etc.
14	TRAV_SP	This data element records the speed the vehicle was traveling prior to the occurrence of the crash as reported by the investigating officer.
15	DEFORMED	This data element records the amount of damage sustained by this vehicle as indicated on the police crash report based on an operational damage scale.
16	VSURCOND	This data element identifies the attribute that best represents the roadway surface condition prior to this vehicle’s critical pre-crash event.
17	VTRAFCON	This data element identifies the attribute that best describes the traffic controls in the vehicle's environment just prior to this vehicle's critical pre-crash event.
18	P_CRASH2	This data element identifies the attribute that best describes the critical event which made this crash imminent (i.e., something occurred which made the collision possible).

19	AGE	This data element identifies this person's age at the time of the crash, in years, with respect to their last birthday.
20	SEAT POSITION	This data element identifies the location of this person in or on the vehicle.
21	INJ_SEV	This data element describes the severity of the injury to this person in the crash using the KABCO scale.
22	MDRDSTRD	This data element identifies the attribute(s) which best describe this driver's attention to driving prior to the driver's realization of an impending critical event or just prior to impact if realization of an impending critical event does not occur.
23	AIR_BAG	This variable tells us whether the airbag was available and deployed in the car crash or not.

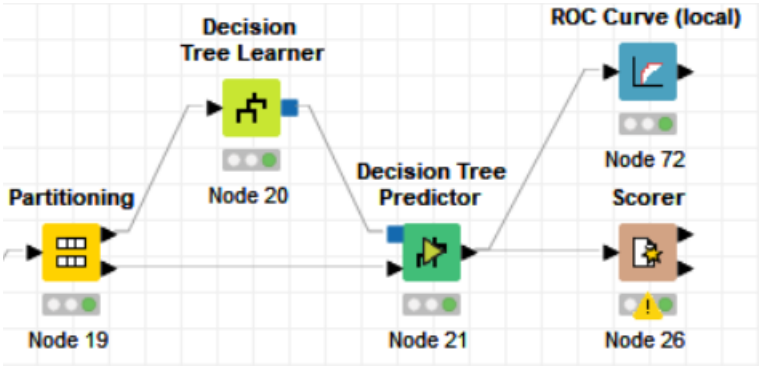
4. Data Modeling

We have created 3 set-based models such as Decision Trees, Random Forest and Naive Bayes. And 3 number-based models such as Logistic Regression, Artificial Neural Network, and K Nearest Neighbor.

Decision trees

For the Decision tree, the preprocessed dataset is split into training and testing with proportions of 70% and 30% respectively by using single Partitioning to balance the data. The Partitioner node helps in creating data for training and testing. The random seed is set to 7654321. The testing set of the data is fed to Decision Tree Predictor. The Decision Tree Predictor tests the testing set with the help of training set. Scorer is added to the Decision Tree Predictor and is configured as first

column = ‘INJ_SEV_binned’ and second column = ‘Prediction (INJ_SEV_binned)’. Once the scorer is executed a confusion matrix is generated.



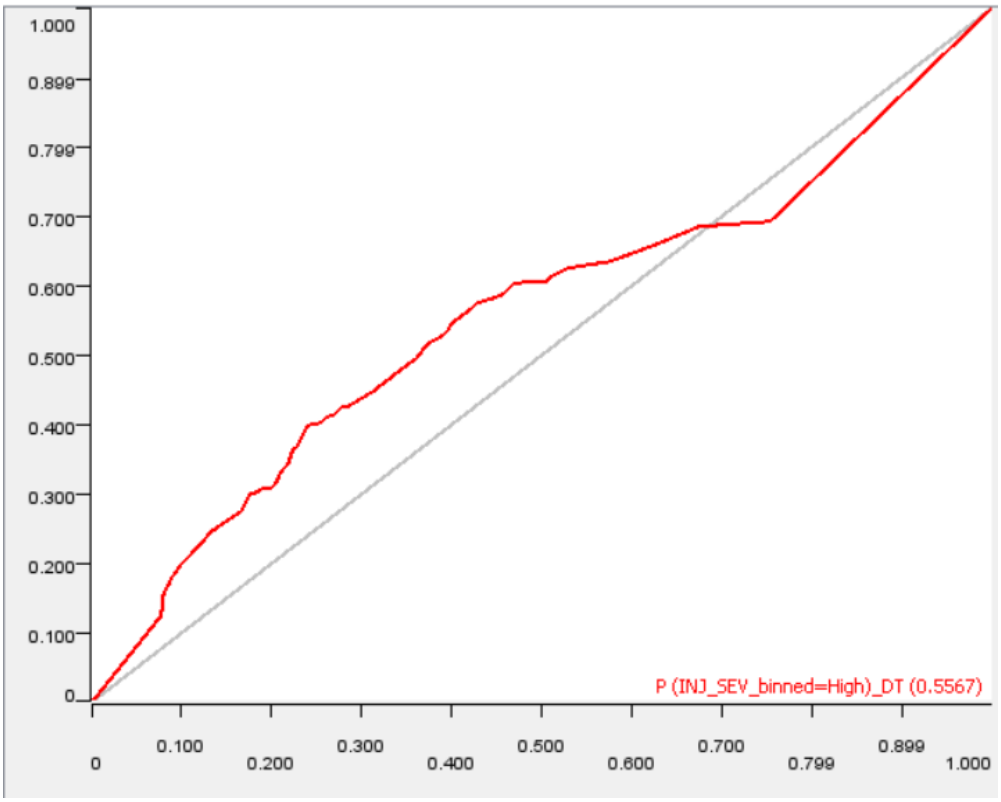
INJ_SEV_binned \ ...	Low	High
Low	1474	284
High	283	106

Correct classified: 1,580 Wrong classified: 567

Accuracy: 73.591 % Error: 26.409 %

Cohen's kappa (κ) 0.111

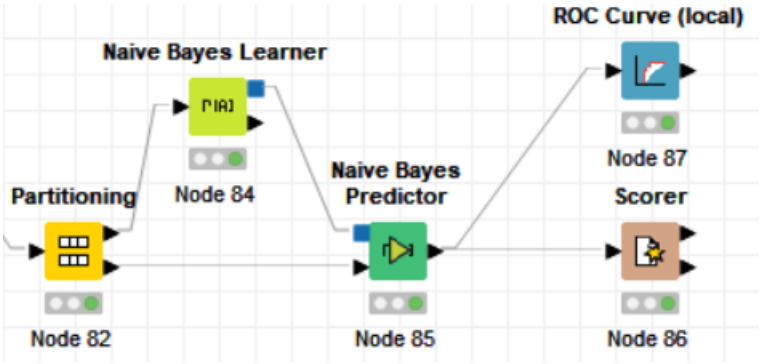
I TruePositives	I FalsePositives	I TrueNegatives	I FalseNegatives	D Sensitivity	D Specificity
1474	283	106	284	0.838	0.272
106	284	1474	283	0.272	0.838
?	?	?	?	?	?



The accuracy of the Decision Tree model is **73.59%** with **1474** True-Positives and **284** False-Negatives. Sensitivity and Specificity are shown in the above accuracy table. The ROC value is **0.5567**.

Naive Bayes

For the Naive Bayes model, the preprocessed data set is split into training and testing with proportions 70% (5026 rows) and 30% (2155 rows). The random seed is set to 7654321 and Stratified Sampling is set to INJ_SEV_binned. The test data is fed through the Naive Bayes Predictor which in turn runs the testing set against the training set being processed through the Naive Bayes Learner. A ROC Curve node is then added to show the trade-off between sensitivity and specificity. Following are the metrics that we generated from this model:



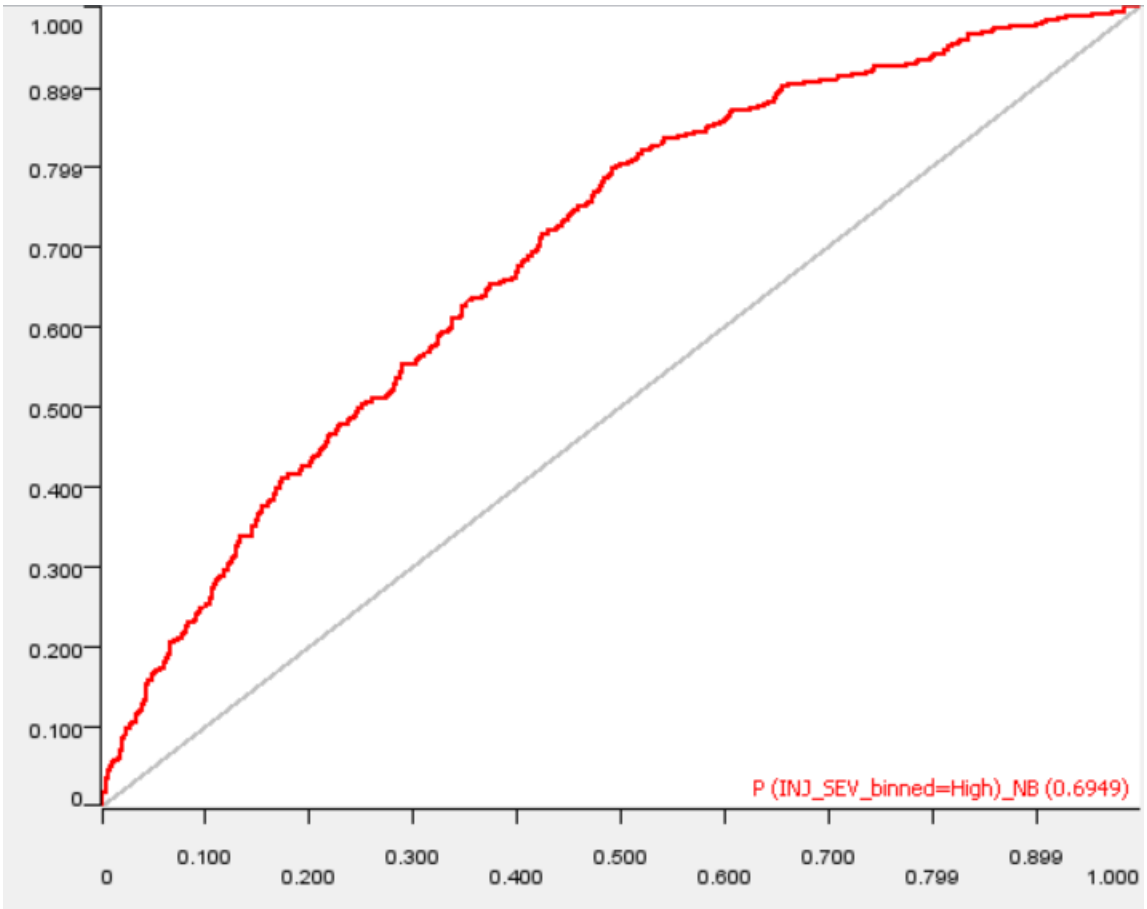
INJ_SEV_binned \ ...	Low	High
Low	1523	241
High	259	132

Correct classified: 1,655 Wrong classified: 500

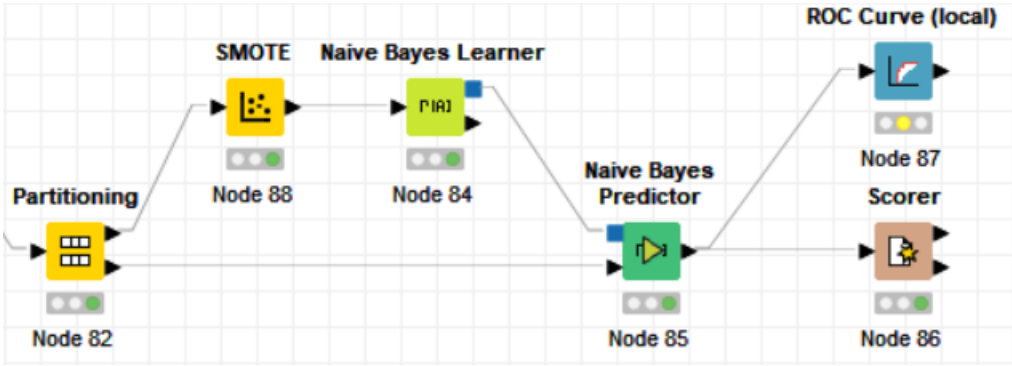
Accuracy: 76.798 % Error: 23.202 %

Cohen's kappa (κ) 0.205

TruePo...	FalsePositi...	TrueNega...	FalseNegati...	Sensitivity	Specificity	Accuracy
1523	259	132	241	0.863	0.338	?
132	241	1523	259	0.338	0.863	?
?	?	?	?	?	?	0.768



To balance out the data before feeding it to the predictor we thought of using a SMOTE node. But when used it drastically drops the accuracy value by 20%. Hence, we thought it is better to not use the SMOTE node, so that we can have a better accuracy for the Naïve Bayes model.



INJ_SEV_binned \ Pre...	Low	High
Low	975	789
High	107	284

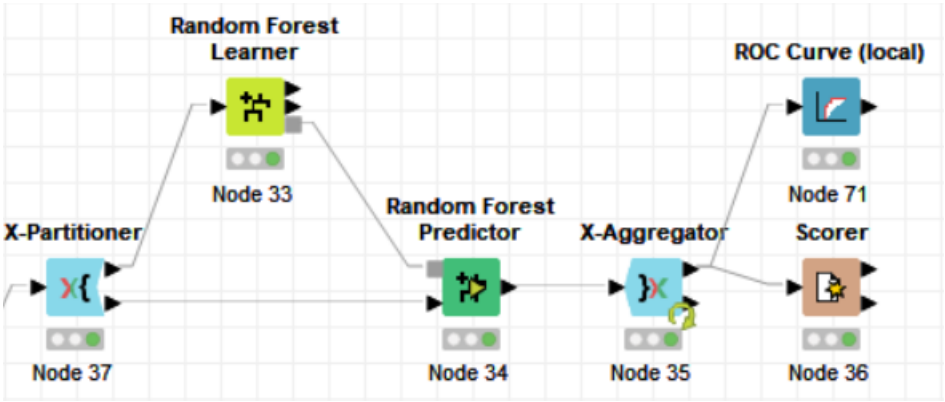
Correct classified: 1,259 Wrong classified: 896

Accuracy: 58.422 % Error: 41.578 %

Cohen's kappa (κ) 0.166

Random Forest

Random Forest Model generates reasonable predictions across a wide range of data. For the Random Forest Model, the preprocessed dataset is split into training and testing with proportions 67% and 33% respectively (default settings for 10 validations) by using X-Partitioner to balance the data. The X-Partitioner node allows to set the number of cross validation iterations that should be performed. The random seed is set to 7654321. The testing set of the data is fed to Random Forest Predictor. The Random Forest Predictor tests the testing set with the help of training set. A X-Partition Aggregator Node is added to collect all the data of every loop for each iteration. These are the different metrics that we obtained from this model:



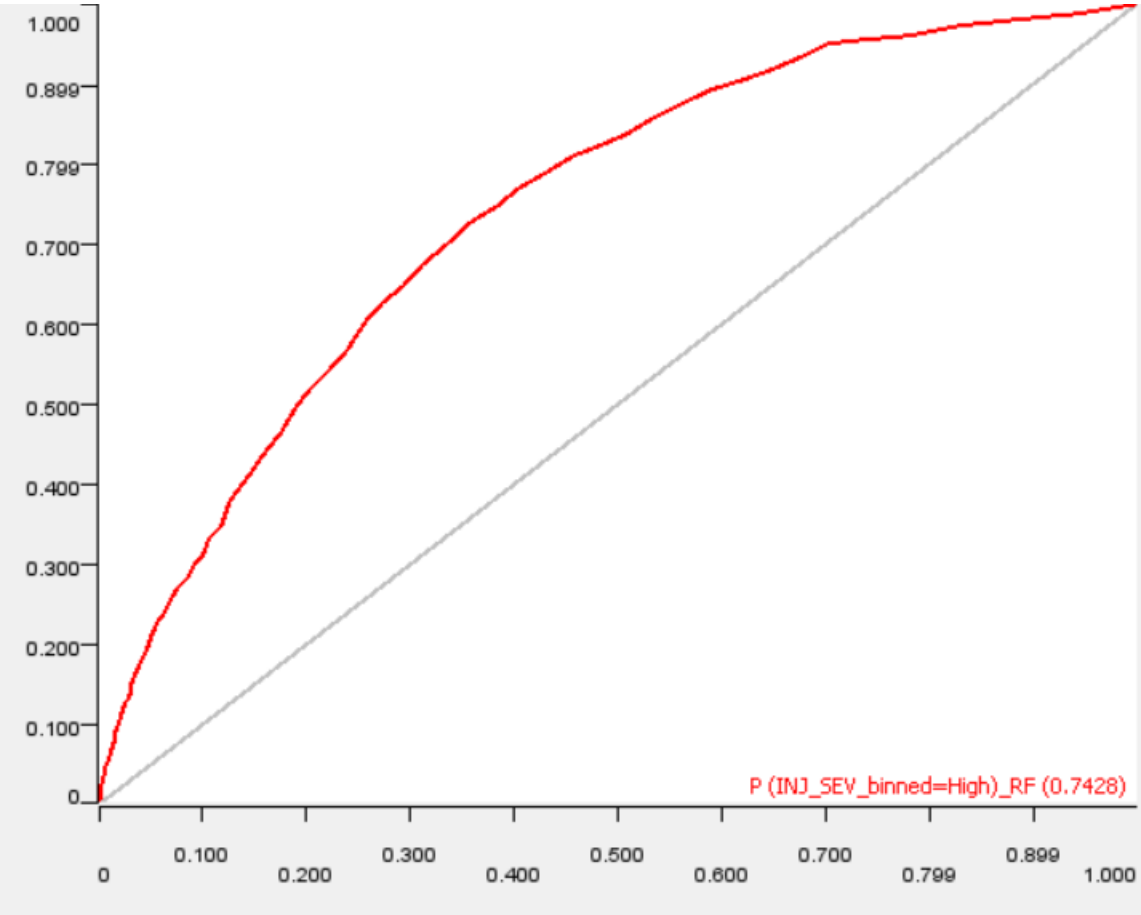
INJ_SEV_binned \ Prediction (INJ_SEV_binned)	Low	High
Low	5733	146
High	1145	157

Correct classified: 5,890 Wrong classified: 1,291

Accuracy: 82.022 % Error: 17.978 %

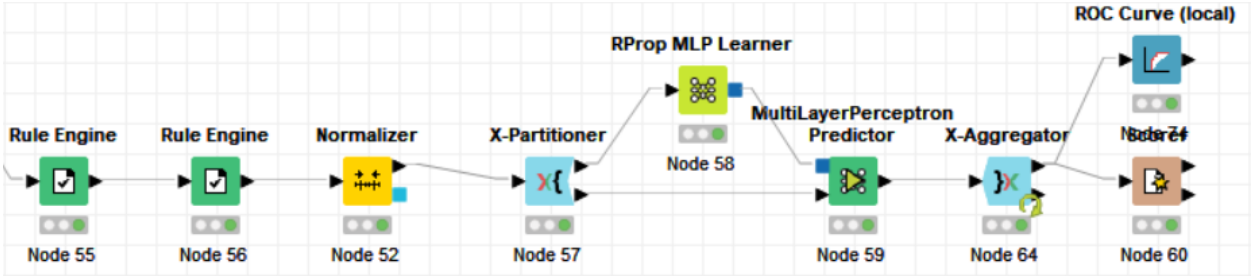
Cohen's kappa (κ) 0.137

TruePo...	FalsePo...	TrueNe...	FalseN...	Sensitivity	Specificity	Accuracy
5733	1145	157	146	0.975	0.121	?
157	146	5733	1145	0.121	0.975	?
?	?	?	?	?	?	0.82



Artificial Neural Network (MLP)

The partitioned data has been used to build an ANN model for predicting the injury severity associated with crashes. This numeric based predictive model consists of a learner and predictor nodes. The learner node develops an Artificial Neural Network, based on the target variable INJ_SEV_binned. In 100 maximum iterations, there are 10 hidden layers, each layer having 10 hidden neurons. The trained Multi-Layer Perceptron model is generated and fed to the predictor to compute the expected output. The classification table has 762 rows and 26 columns. Following are the metrics obtained from this model:

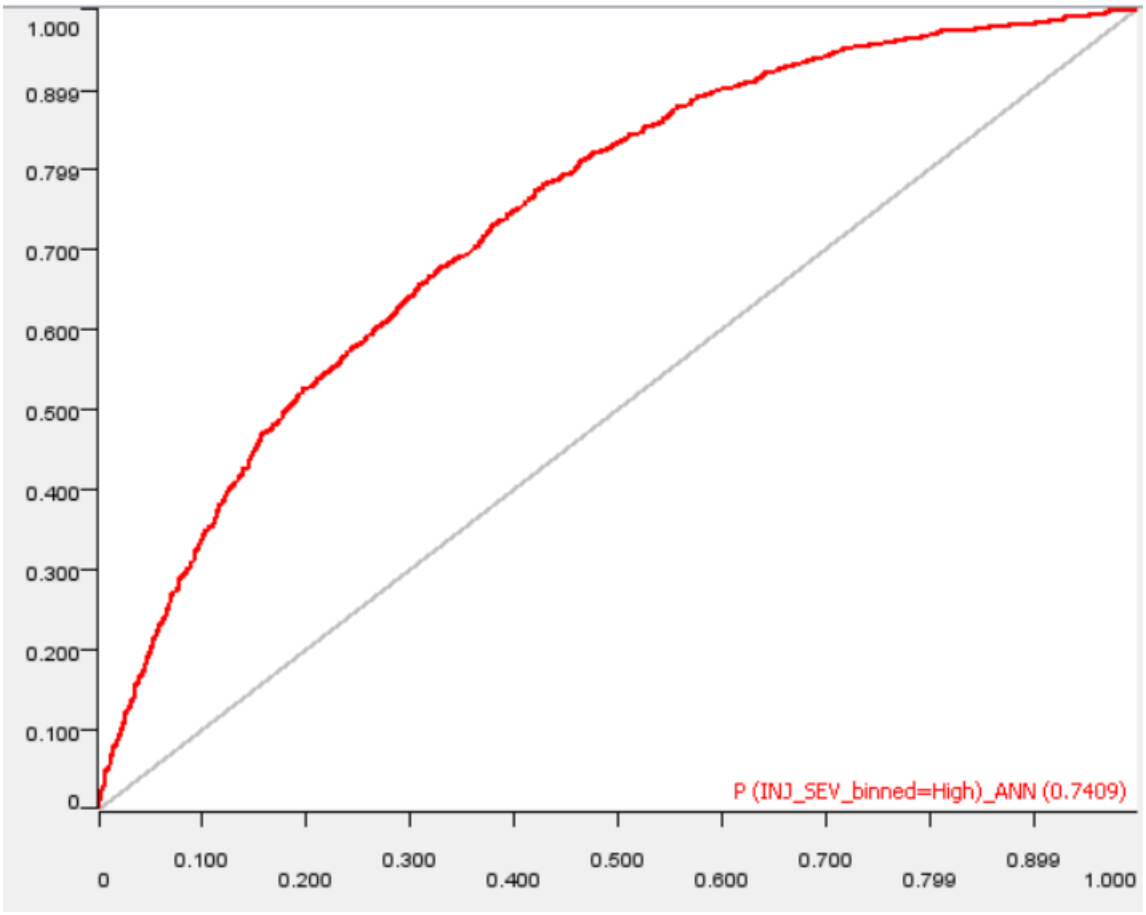


INJ_SEV_binned \ Prediction (INJ_SEV_binned)	Low	High
Low	5753	126
High	1170	132

Correct classified: 5,885 Wrong classified: 1,296

Accuracy: 81.952 % Error: 18.048 %

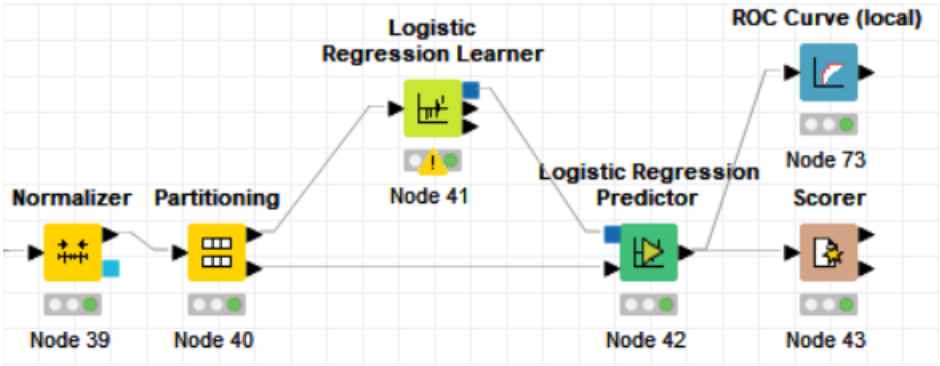
Cohen's kappa (κ) 0.116



I TruePo...	I FalsePo...	I TrueNe...	D Precision	D Sensitivity	D Specificity	D Accuracy
5753	1170	132	0.831	0.979	0.101	?
132	126	5753	0.512	0.101	0.979	?
?	?	?	?	?	?	0.82

Logistic Regression

The filtered dataset with 22 columns and 7181 rows is first fed to the Normalizer. Then the normalizer converts data in between min and max values. It converts the data to 1 and 0 binary format. This data is necessary for the Logistic Regression model to work. The next node is the Partitioning node. Here, the data is split relatively 70% and 30%. We split it based on stratified sampling on inj_sev_binned. The random seed is set to 7654321. The training data is now sent to the Logistic Regression Learner and the testing data is sent directly to Logistic Regression Predictor. Our dependent variable is selected as "INJ_SEV_binned" and the reference category is considered as "high". In the Logistic Regression Predictor node, the predicted data column and the testing dataset is fed. It appends a new column that splits into a binary variable. The final data is fed to the Scorer and the ROC curve(local) node. Following are the metrics that are obtained from this model:



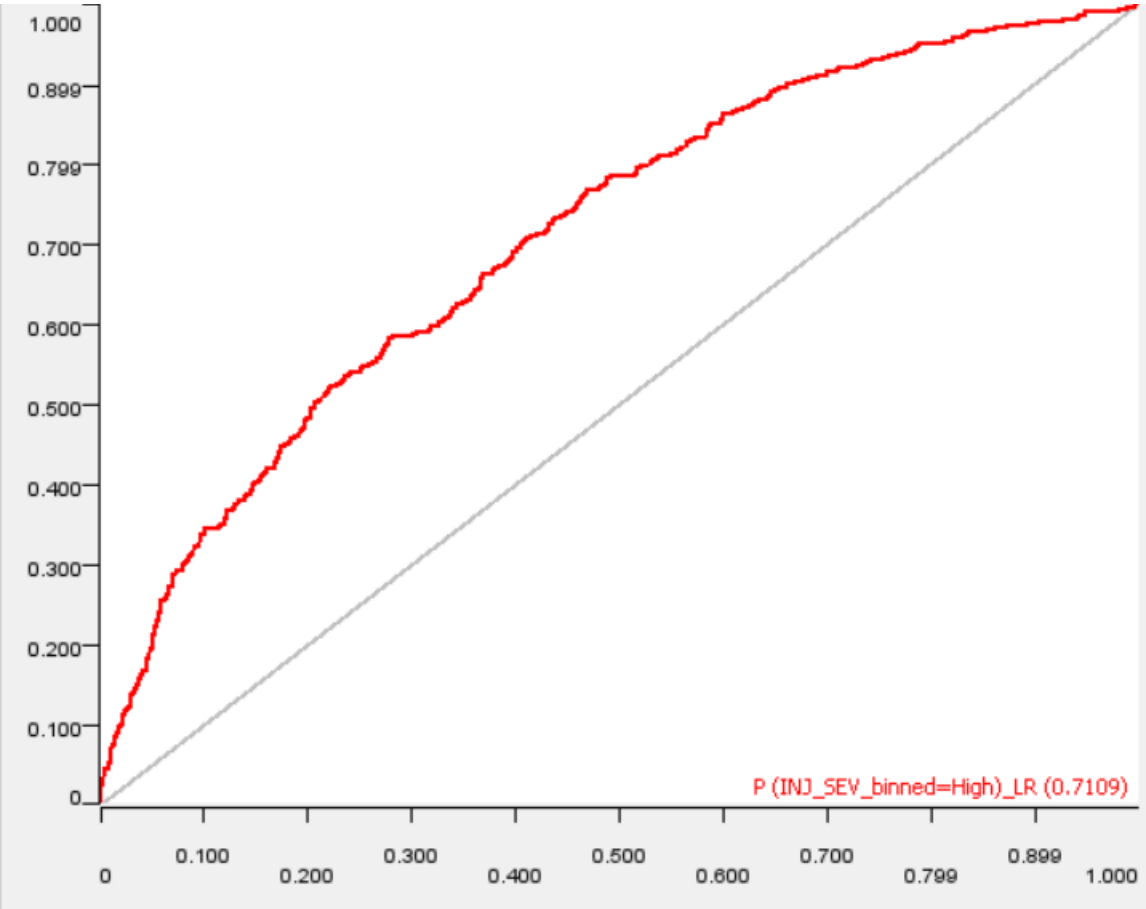
INJ_SEV_binned \ Prediction (INJ_SEV_binned)	Low	High
Low	1747	17
High	368	23

Correct classified: 1,770 Wrong classified: 385

Accuracy: 82.135 % Error: 17.865 %

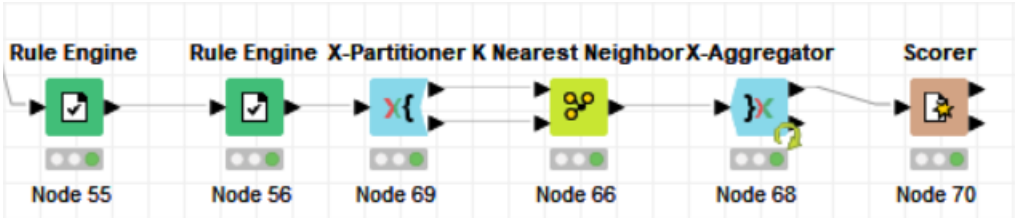
Cohen's kappa (κ) 0.076

TruePo...	FalsePo...	TrueNe...	FalseN...	Sensitivity	Specificity	Accuracy
1747	368	23	17	0.99	0.059	?
23	17	1747	368	0.059	0.99	?
?	?	?	?	?	?	0.821



K-Nearest Neighbor

The KNN model assumes that similar things exist in close proximity. In other words, "birds of a feather flock together". This numeric predictor model consists of learner and predictor nodes in order to build out and predict the injury severity of car crashes. The X-Partitioner node is set to 10-fold, Stratified Sampling Column is set to 'INJ_SEV_binned' and Random Seed box is set to 7654321. After the data process through the KNN node we connect the output to the X Aggregator input. Target column is set to 'INJ-SEV_binned' and the Prediction Column is set to 'Class [kNN]'. Output is then connected to the Scorer node. Following are the different metrics that are obtained from this model:



INJ_SEV_binned \ Class [kNN]	Low	High
Low	5356	523
High	1076	226

Correct classified: 5,582 Wrong classified: 1,599

Accuracy: 77.733 % Error: 22.267 %

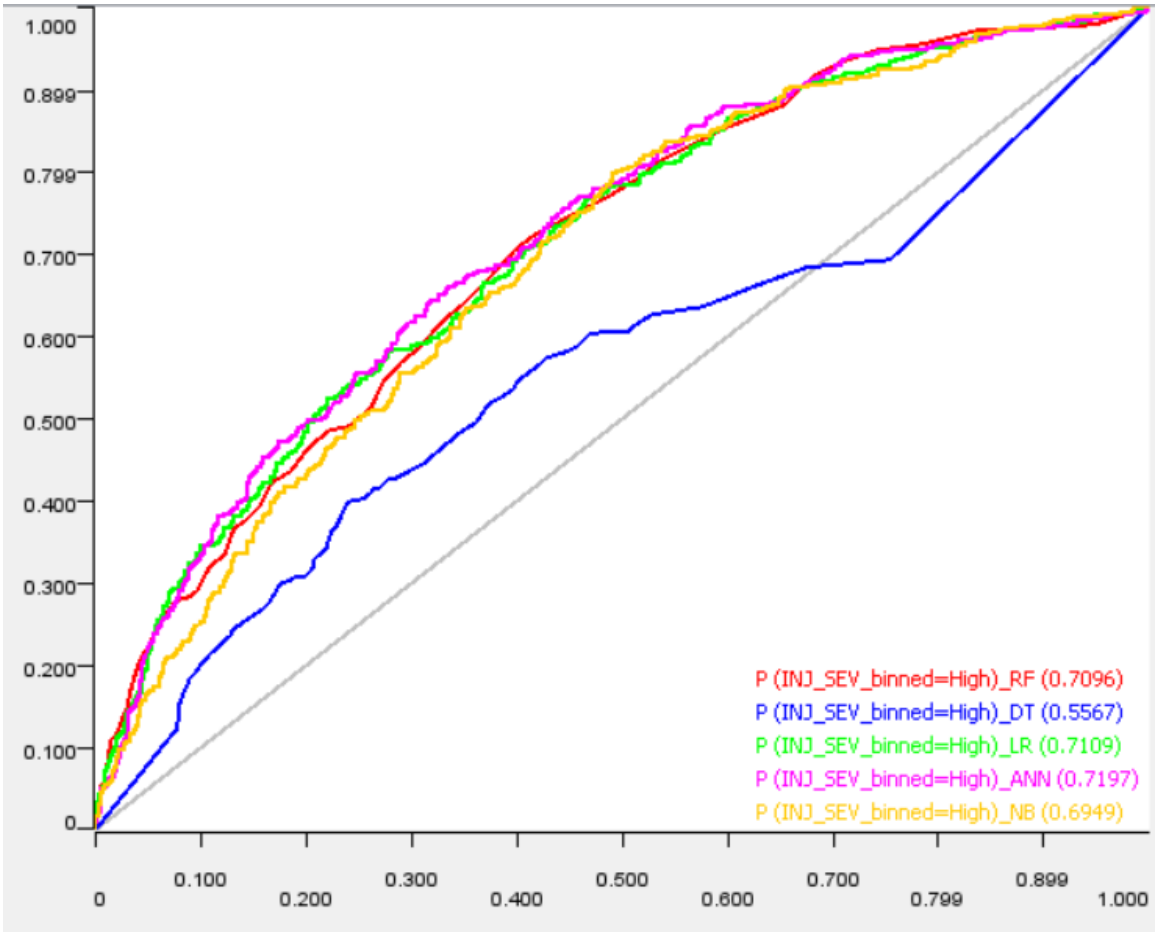
Cohen's kappa (κ) 0.101

TruePo...	FalsePo...	TrueNe...	FalseN...	Precision	Sensitivity	Specificity	Accuracy
5356	1076	226	523	0.833	0.911	0.174	?
226	523	5356	1076	0.302	0.174	0.911	?
?	?	?	?	?	?	?	0.777

5. Evaluation

We built our prediction models to predict the severity of injury in a crash. By building all the different models we saw that the Logistic Regression model and the Random Forest model gives the best accuracy. They classify the data into True Positives and True negatives with an accuracy of **82.13%** and **82.08%**. The Artificial Neural Network model comes second to the Logistic Regression model as it gives an accuracy of **81.95%**. The K-Nearest Neighbor model and the Naive Bayes model perform similarly giving an accuracy around **77%**. We found that the Decision Tree model has a comparatively poor performance. It has an accuracy of **73.59%**.

This seems to be true because Decision Tree is a set-based model type and all the data that we have is in numeric format. This tells us that the number-based models would be preferred for such data types. We had assumed that the number-based model would work the best for this data type which turned out to be true.



After completing our modeling process and combining the ROC curves for all the models we have found that ANN and Logistic Regression have the maximum Area under the curve. ANN has a ROC value of **0.7197** and Logistic Regression has a ROC value of **0.7109.** Also, we can see that Decision Tree has a poor performance as it has ROC value of **0.5567**.

6. Deployment

The findings of our modeling techniques indicate that it is a perpetual discovery. It can assist many sections of workers like the lawmakers, the medical organizations, and insurance companies.

The lawmakers can use these findings to explore and decide what laws are required to be included to avoid accidents and crashes on roads. They can determine which laws to enact and what public issues need more attention. These analysis results can guide them to improve the existing laws as well. This will also promote administrators' commitments to data-driven initiatives, enact

strategies supportive of data usage, and build a culture that prioritizes data as a strategic asset to guide decision-making.

The medical organizations can utilize the findings to prepare themselves with first aids for emergencies and medication. The healthcare sector always deals with unprecedented events and by these analyses, it can improve their number of healthcare services. The analysis results can also enhance its quality of care, productivity, and efficiency.

Policymakers and policy analysts can use the inferences to decide what medical claims and insurance policies need to be incorporated in the healthcare policy. The findings can support them in developing new policies and implementing and monitoring major transformations in existing policies. It will be an advancement of the insurance market, resulting in the promotion of affordability and expansion of medical coverages of policyholders.

The lessons derived from the analysis will encourage further case studies, open doors for opportunities for various markets, and improve opinions while making informed decisions.

Conclusion

In this project, we understood that data preprocessing is the most important step to build good predictive models. The data preparation part took the most amount of time when compared to the other steps. While building predictive models, we found that the Logistic Regression model is the best fitting model to predict the severity of the injuries that occurred in the car crashes. It gave an accuracy of 82.13%, which means that it classified the True Positives and False Negatives more accurately. It indicates that it has the best chance of giving reliable prediction results based on the selected variables when compared to other predictive models.

We also tried to use the SMOTE node in our Naïve Bayes model. And found that by oversampling the data fed to the model led to decrease in the accuracy. This dataset seemed to favor the number loving models as those models gave an average accuracy of around 80%. The analysis performed would be helpful for assessing the consequences of different safety-related policies and trying to understand what factors lead to the high severity of injury in the case of a car crash.

References

1. Meier, A., Gonter, M., & Kruse, R. (2014). Precrash classification of car accidents for improved occupant safety systems. *Procedia Technology*, *15*, 198-207.

2. Rosman, D. L. (2001). The Western Australian Road Injury Database (1987–1996):: ten years of linked police, hospital and death records of road crashes and injuries. *Accident Analysis & Prevention*, *33*(1), 81-88.

3. Barraclough, P., af Wåhlberg, A., Freeman, J., Watson, B., & Watson, A. (2016). Predicting crashes using traffic offences. A meta-analysis that examines potential bias between self-report and archival data. *PLoS one*, *11*(4), e0153390.

4. Park, S., Jang, K., Park, S. H., Kim, D. K., & Chon, K. S. (2012). Analysis of injury severity in traffic crashes: a case study of Korean expressways. *KSCE Journal of Civil Engineering*, *16*(7), 1280-1288.

5. Weiss HB, Kaplan S, Prato CG. Analysis of factors associated with injury severity in crashes involving young New Zealand drivers. Accid Anal Prev. 2014 Apr;65:142-55. doi: 10.1016/j.aap.2013.12.020. Epub 2014 Jan 6. PMID: 24456849.