Abstract

Data Science Cornerstone Report

by Jörg Bergmann

as of 15th October 2020, Darmstadt, Germany

Published on website:

www.herbfrisch.de

Git Hub link:

https://github.com/herbfrisch/jbs_cornerstone

1 Introduction Section

In this section we discuss the business problem and who would be interested in this project.

1.1.Introduction/Business Problem Car Accident Severity Prediction

1.1.1.Introduction/Background

This section defines the business problem wth risks in the mobility area. It is about to solve the decision problem, to drive or not to drive with a car under certain known conditions from the current location to a destination at a planned time in realation to the risk for having an accident.

1.1.2.Problem

In the following, this report is to predict the severity of an accident. The scenario could be described as follows: Say, you are driving to another city for work or to visit some friends. It is rainy and windy, and on the way, you come across a terrible traffic jam on the other side of the highway. Long lines of cars barely moving. As you keep driving, police car start appearing from afar shutting down the highway. Oh, it is an accident and there's a helicopter transporting the ones involved in the crash to the nearest hospital. They must be in critical condition for all of this to be happening. Now, wouldn't it be great if there is something in place that could warn you, given the weather and the road conditions about the possibility of you getting into a car accident and how severe it would be, so that you would drive more carefully or even change your travel if you are able to?

1.1.3.Interests

In the following this is handled as a data science problems that targets the car drivers audience in the first place, but is also meant to help all involved stakeholders to mitigate risk in the mobility, insurance, healthcare area, and at least for the family of the driver.

This is exactly what will be handled in this report: to predict the severity of a possible car accident on the base of available car accident data from the past and current driving conditions.

1.2.Introduction to the Car Accident Severity Data

1.2. Data acquisition and cleaning

Data

In this section the data that will be used to solve the problem is described. It contains the explanation, why the is date is adequate for the problem and is used. In the discussion part, examples of data is provided.

1.2.1 Data sources

To deal with accidents data the shared data for Seattle city is used as an example:
https://s3.us.cloud-object-storage.appdomain.cloud/cf-courses-data/CognitiveClass/DP0701EN/version-2/Data-Collisions.csv
Metadata descriptions:
https://s3.us.cloud-object-storage.appdomain.cloud/cf-courses-data/CognitiveClass/DP0701EN/version-2/Metadata.pdf

1.2.2 Data cleaning

After open the CSV file and check what type of data is contained. The first column colored in
yellow is the labeled data. The remaining columns have different types of attributes. A selection of these attributes are used to train the model. Also most of the observations found are of sufficent quality to train and test the machine learning model to be build in this case.

The label for the data set used is severity, which describes the
fatality of an accident. It shall be notice that the shared data has unbalanced labels. So the data is balanced, otherwise, a biased ML model would be created.

1.2.3 Feature selection

The following is a list of attributes or features that are used. For description of each attribute, it is refered to the web link on the CSV file.
SEVERITYCODE, STATUS,
ADDRTYPE

- Alley
- Block

• Intersection

INTKEY,

with a collision

LOCATION,

SEVERITYCODE,

● 3—fatality

● 2b—serious injury

● 2—injury

● 1—prop damage

● 0—unknown

PERSONCOUNT,

- Collision address type:
- Key that corresponds to the intersection associated
- Description of the general location of the collision
- A code that corresponds to the severity of the collision:
- The total number of people involved in the collision

PEDCOUNT,

This is entered by the state.

PEDCYLCOUNT, entered by the state.

VEHCOUNT,

entered by the state.

INCDATE, INCDTTM, JUNCTIONTYPE, INATTENTIONIND,

- The number of pedestrians involved in the collision.
- The number of bicycles involved in the collision. This is - The number of vehicles involved in the collision. This is
- The date of the incident.
- The date and time of the incident.
- Category of junction at which collision took place
- Whether or not collision was due to inattention. (Y/N)

UNDERINFL, - Whether or not a driver involved was under the influence of drugs or alcohol.

WEATHER,

of the collision.

ROADCOND, LIGHTCOND,

PEDROWNOTGRNT, granted. (Y/N)

SPEEDING, (Y/N)

ST_COLCODE,

collision. For more information about these codes, please see the State Collision Code Dictionary. Codes: 0-5, 10-32, 40-57, 60-67, 71-74, 81-84.

SEGLANEKEY, occurred.

CROSSWALKKEY,

HITPARKEDCAR car. (Y/N)

- A key for the lane segment in which the collision
- A key for the crosswalk at which the collision occurred. - Whether or not the collision involved hitting a parked
- A description of the weather conditions during the time

- The condition of the road during the collision. - The light conditions during the collision.
- Whether or not the pedestrian right of way was not
- Whether or not speeding was a factor in the collision.
- A code provided by the state that describes the

In addition, there is probably need to do some feature engineering to improve the predictability
of the model as follows:
From the ST_COLCODE, a smaller set of cathegories could be defined. From INCDATE the day of the week could be calculated.
From INCDTTM the part of the day could be calculated: morning, afternoon, evening, night.
The target or label columns should be accident " severity" in terms of human fatality, traffic delay, property damage, or any other type of accident bad impact. These terms are cathegories and are constructed from the last attributes listed above.
Then, the built severity data set is applied building a machine learning model.

2 Data Section

2.1 Data

In this section we describe the data that will be used to solve the problem and the source of the data.

2.2 Data Wrangling

2.3 Pre-processing

3 Methodology Section

3.1 Methodology

This section which represents the main component of the report
where we discuss and describe any exploratory data analysis that we did, any inferential statistical testing that we performed, if any, and what machine learnings were used and why.

We will use the following machine learning models for car accident prediction:

- K-Nearest Neighbors (KNN)
- Decision Tree

3.2 K-Nearest Neighbors (KNN)

In this Project we will use K-Nearest Neighbors to predict a data point, whether SERVERITYCODE is 1 or 2.

K-Nearest Neighbors is an algorithm for supervised learning, where the data is 'trained' with
data points corresponding to their classification. Once a point is to be predicted, it takes into account
the 'K' nearest points to it to determine it's classification.

In this case, we have data points of SERVERITYCODE 1 and 2. We want to predict what the star (test data point) is.
If we consider a k value of 3 (3 nearest data points) we will obtain a prediction of class SERVERITYCODE 2 which is the worst case for an traffic accident.
Yet if we consider a k value of 6, we will obtain a prediction of Class SERVERITYCODE 1.

In this sense, it is important to consider the value of k. But hopefully from the resulting diagram,
we should get a sense of what the K-Nearest Neighbors algorithm is.
It considers the 'K' Nearest Neighbors (points) when it predicts the classification of the test point.

A number of required libraries must be loaded.

3.3 Decision Tree

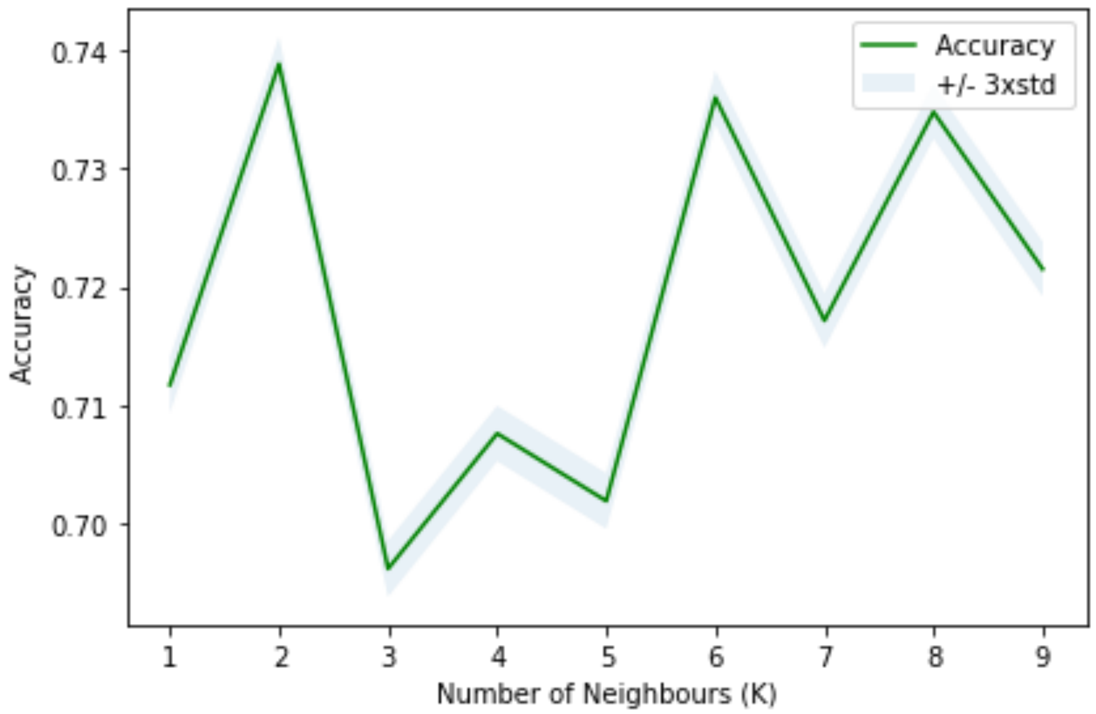
4 Results Section

Evaluation for KNN

Accuracy evaluation for KNN
In multilabel classification, accuracy classification score is a function that computes subset accuracy.
This function is equal to the jaccard_similarity_score function.
Essentially, it calculates how closely the actual labels and predicted labels are matched in the test set.

```
print("Train set Accuracy: ", metrics.accuracy_score(y_train, neigh.predict(X_train)))
print("Test set Accuracy: ", metrics.accuracy_score(y_test, yhat))
```

```
Train set Accuracy:  0.7127332985627423
Test set Accuracy:  0.707589521495722
```



```
print( "The best accuracy was with", mean_acc.max(), "with k=",
mean_acc.argmax()+1)
```

```
The best accuracy was with 0.7388560261962607 with k= 2
```

Evaluation for Decision Tree

```
print("DecisionTrees's Accuracy: ", metrics.accuracy_score(y_dtree_testset,
predTree))
```

```
DecisionTrees's Accuracy:  0.7446216682511179
```

So Accuracy for Decision Tree is at 74%

Accuracy classification score computes subset accuracy:
the set of labels predicted for a sample must exactly match the corresponding set of labels in y_true.
In multilabel classification, the function returns the subset accuracy.
If the entire set of predicted labels for a sample strictly match with the true set of labels,
then the subset accuracy is 1.0; otherwise it is 0.0.

Fig. for decision tree available when prerequisite python libraries installed.

5 Discussion Section

In this section we discuss any observations you noted and any recommendations you can make based on the results.

5.1 Selection of machine learning models

We used the following machine learning models for car accident prediction:

- K-Nearest Neighbors (KNN)
- Decision Tree

5.2 Accuracy of the models

The obtained accuracy of both seems to be suitable for resolving the problem of car accident prediction on current available weather observations and conditions for a planned car travel.

5.3 Technical Installation Prerequisites

For the models we need a number of python libraries. To install the right version of these in the development environment without having any python error messages when developing the models was a tedious job.

For KNN the necessary libraries could be installed and a graphical plot could be printed.

For Decision Tree the necessary libraries most could be installed. We experience timeouts when installing the libraries for Decision Tree and a graphical plot could not be printed. This problem needs to be fixed in the near future.

5.4 Unresolved Error Messages

The following error message when preparing the decision tree plot could not be resolved:

```
InvocationException: GraphViz's executables not found
```

5.5 Integrated Development Environments IDEs

We used to different IDEs:

- IBM Cloud, Jupyter Notebook
- Anaconda on iMac OSX Jupyter Notebook

Both work fine for the development of the machine learning models

Setup the specific technical environment for Jupyter could be improved for machine learning problems.

Elapsed installation times for prerequisite python libraries in both IDEs are not acceptable long.

5.5 Other machine learning models

5.6 Business Understanding phase
The initial phase to understand the project's objective from the business or application perspective could be resolved.

Translation of this knowledge into a machine learning problem with a preliminary plan to achieve the objectives could be resolved.

5.7 Data understanding phase
Collecting or extracting the dataset from various sources such as csv file or SQL database could be resolved. csv File was used for that task. A SQL database dis not apply.

Determining the attributes (columns) that are useed to train the selected machine learning model could be resolved. Also, assessing the condition of chosen attributes by looking for trends, certain patterns, skewed information, correlations, and so on was resolved initially, but could be improved. At leaset we found only two severity categories, which depend not only on the current weather conditions. These patterns need further studies.

5.8 Data Preparation phase
Data preparation included all the required activities to construct the final dataset which were fed into the selected modeling tools. Data preparation was performed multiple times and it included balancing the labeled data, transformation, filling missing data, and cleaning the dataset. This was until now the major effort of the project: data cleansing. The available data quality needs in general more attention, which can not be assumed without effort. This is also true for the right data available for statistics problems to be resolved with the selected models.

5.9 Modeling phase
In this phase, in general various algorithms and methods can be selected and

applied to build the model including supervised machine learning techniques. We selected only two: KNN and decision tree. Furthermore, SVM, XGBoost, decision tree, or any other techniques could be selected as well. This is for further studies. In general, a single or multiple machine learning models for the same data mining problem could be selected. At least, herewith only two machine learning models were selected. At this phase, stepping back to the data preparation phase was often required. This was also high effort prone.

5.10 Evaluation phase

Before proceeding to the deployment stage, the model needed to be evaluated thoroughly to ensure that the business or the applications' objectives are achieved. Certain metrics could be used for the model evaluation such as accuracy, recall, F1-score, precision, and others. For this project, only accuracy was calculated ans evaluated for the selected machine learning models. Both have acceptable values. Other metrics are for further studies.

5.11 Deployment

In general, as he deployment phase requirements varies from project to project, the report is deployed to a website of the author. As this can be as simple as creating a report, developing interactive visualization, or making the machine learning model available in the production environment, the working files a submitted to the authors' Git hub. In this environment, the possible customers or end-users can utilize the model in different ways such as API, website, or so on. At least, this work is published to everyone interested.

Published as a blog on: www.energing.de

Guthub: https://github.com/herbfrisch/jbs_cornerstone

6 Conclusion Section

In this section we conclude the report.

As the work is still in progress, this conclusion part is of preliminary status.

Accident severity probability prediction is feasible on the basis of weather data for K-Nearest Neighbors (KNN) and Decision Tree machine models.

Additional machine learning models ar for further studies.

In depth assessment the condition of chosen weather and accident attributes by looking for trends, certain patterns, skewed information, correlations, and so on is for further study.