Project Report

INNOPOLIS UNIVERSITY

Car accident severity prediction

Students	Timur Sergeev Dinmukhametov Salavat
Course	Big Data Technologies and Analytics
Semester	Spring
Year	2023

Contents

Note: Follow this Moodle link for more info on CRISP-DM shared by Armen

1. Introduction

In this report we provided detailed explanation of every stage of our project from problem statement to deployment of trained and fine-tuned model.
We also discussed how our project relates to current business solutions and potential impact.
At the end we delivered possible ways of continuation of our work, because current work was limited due to semester time limitation and our workload.

2. Business Understanding

2.1 Current situation assessment

Contemporary map services can provide very accurate route time prediction and based on it recommend the best path to traverse. But there is no engine that could predict potential car accident appearance to prevent driver from choosing wrong way or to even warn him in cases of potential danger.

2.1.1 Inventory of resources*[Core]*

Resources used in project:

- US Accidents dataset
- US cities dataset
- PostgreSQL
- Avro
- Hive
- Zeppeline
- PySpark
- Virtual machine based on Apache Ambari with services needed for data processing.

2.1.2 Requirements, assumptions and constraints

In this project only open sources of data were used, any private information that could impose some restrictions was not included. Task of car accident prediction have very strict bounds of evaluation of result acceptability, that could be satisfied only performing more complex study.
Task of car accident classification is easier to perform and it's preciseness could be evaluated less strictly, in that case reaching approximately 80% of accuracy and f1-score will be enough to reach a success in this challenge.

2.1.3 Risks and contingencies*[Core]*

Proposed model could be non scalable in case of limitation of given train data that includes information about car accidents only in US.

2.1.4 Business Objectives *[Core]*

Main objective of current work is to provide the best route engine for map service that is able to make complex predictions according to diverse data.

2.2 Data mining objectives*[Core]*

In order to complete business objective we should solve task of car accident severity classification.

2.2.1 Business success criteria *[Core]*

To evaluate degree of success of our project, we could compare such metrics indicators before and after our module deployment, as CTR, amount customers, amount of time spent, amount of completed tasks.

2.2.2 Data mining success criteria *[Core]*

For car accident severity classification value of accuracy and f1-score close to 0.8 will be acceptable.

2.3 Project plan

In this work we propose whole pipeline of data process that ends up in precise prediction of potential car accident occurrence and it's severity.
Service based on project we developed could pass high-dimensional data from different sources (sensor, cameras) in real-time and give information about potential risk situation in traffic to map engine. This information could be used to analyze route according to possible car accidents that were not even happened yet. It also could prevent drivers from getting into an accident by warning them in cases of very high possibility of incident.
Car accident prediction based on data we have seemed to be more complex task that were postponed to future research.
In this stage of project we concentrated on classification of car accident that have already happened in a terms of caused delay on traffic. We have a scale of 4 values, where 1 indicates the least impact on traffic (i.e., short delay as a result of the accident) and 4 indicates a significant impact on traffic (i.e., long delay).

3. Data Understanding

The second stage of the CRISP-DM process requires you to acquire the data listed in the project resources. This initial collection includes data loading, if this is necessary for data understanding. For example, if you use a specific tool for data understanding, it makes perfect sense to load your data into this tool. If you acquire multiple data sources then you need to consider how and when you're going to integrate these.

3.1 Initial data collection *[Core]*

In this project we used data that takes car accidents. We found this dataset into kaggle datasets. Dataset consist of different parts, collected from multiple API. We have not encoutered with problems at all.

3.1.1 Big data pipeline: Stage I*[Core]*

We created PostresSQL database like presented in the assignment tutorial. We created schema with different datatypes and imported data into PostgreSQL. After that we created scheme into Avro to import that data into Hive. Because of our data is simple structured, we have not used any partition.

3.2 Data Description*[Core]*

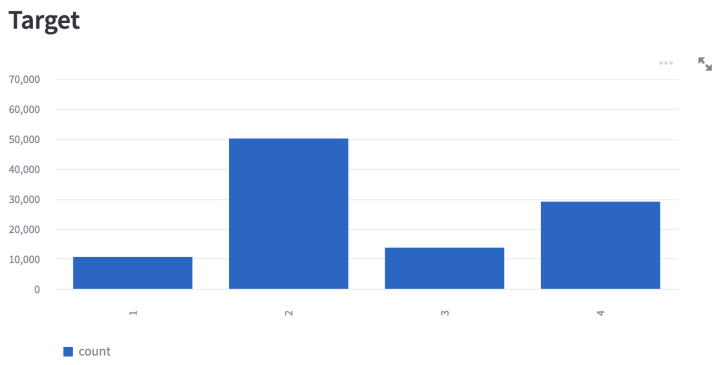

Figure 1: Target distribution

There 1 are most of the data are have 2 severity.

There 2 are linear correlation with a target.

Amount of accidents are linearly growing 3.

There 4 are some of the accidents happens in the twilights.

More than half of the accidents 1 happens around traffic signal 5.

Big data pipeline: Stage II*[Core]* We build Hive table by using Avro. Hive reads avro schema and import all of the data into Hive Server.

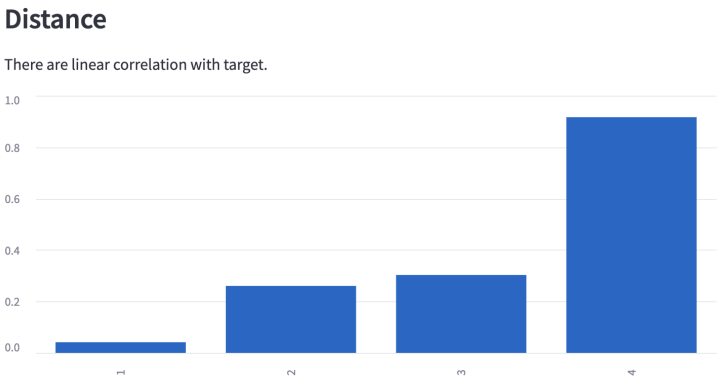

Figure 2: Distance

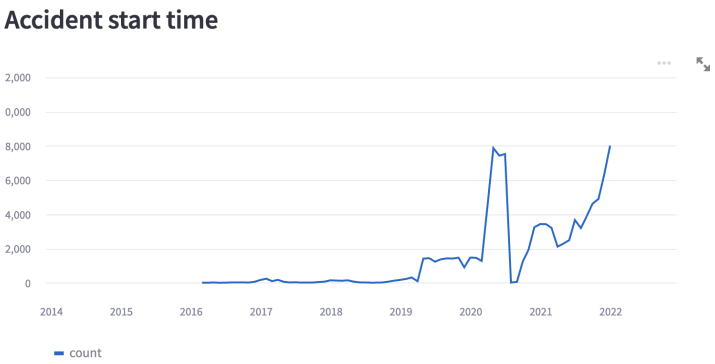

Figure 3: Time of the accident

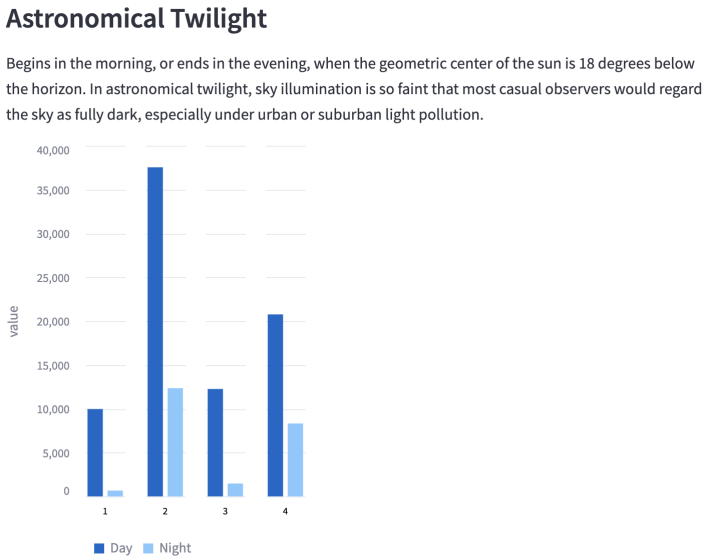

Figure 4: Astronomical twilight

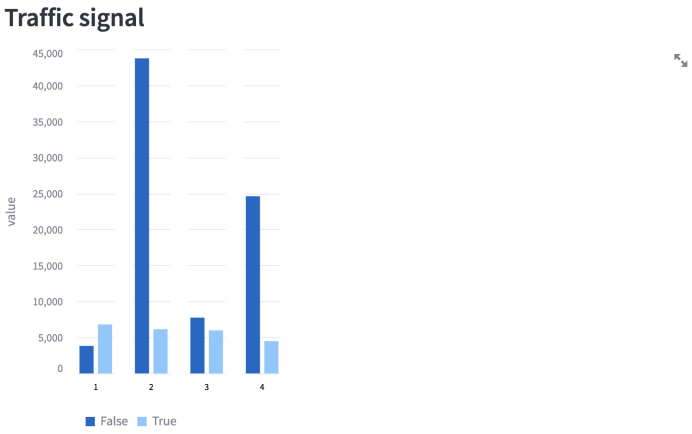

Figure 5: Traffic signal

3.3 Data exploration*[Core]*

See data exploration into data description section.

Data exploration report – We found that distance and traffic signal have postive correlation.

Big data pipeline: Stage II*[Core]* We collected the geographical features. We found the nearest city and calculated the distances between city and car accident. We got some of the time-based features. To get them we collected amount of car accident at each hour, we collected hour and day of the week then accident happen. After that we categorize categorical features with one hot encoding and normalize of numerical features into range from 0 to 1.

3.4 Data quality

Examine the quality of the data, addressing questions such as:

- Is the data complete (does it cover all the cases required)? No, we got a lot of nans. We remove them by using special functions.
- Is it correct, or does it contain errors and, if there are errors, how common are they? There are no errors in reading.
- Are there missing values in the data? If so, how are they represented, where do they occur, and how common are they? Around half of the rows have missing values.

4. Data Preparation

4.1 Data selection*[Core]*

We removed all rows with nans. Total we got around 900k of rows. After that we found that class 2 is oversampled, so we subsample class 2. In totat that we got 103k rows of data. We have 43 columns, but we sampled geographical features, weather and twilight columns.

Rationale for inclusion/exclusion –

- City - removed, have a lot of classes to encode.
- Distances - added, found distance between start and end of the accidents.
- DayOfWeek - added, there are some of the linear correlation.
- Hour - added, there are some hidden correlation with a target.
- DistCity - added, distance to nearest city.

4.2 Data cleaning*[Core]*

Data cleaning report We removed all rows with nans.

4.3 Data construction

We created new features based on time and geographical data.

Derived attributes We found distance between start and end of the accidents. We added distance with nearest city.

4.4 Data integration

We used additinal dataset with geographical features of nearest city.

Merged data We computed BallTree to get nearest city position and calculated distance with point.

Aggregations We computed hour and day of the accident.

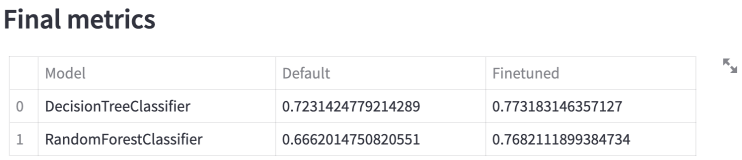

Figure 6: Final metrics

5. Modeling

In this step, the datasets built and prepared in the previous phases were used to serve a modeling tool, with the purpose of finding the model that would best explain the existing data and predict the target variable for new observations.

5.1 Select modeling technique

We used to models: Decision Tree and Random forest.

Modeling technique*[Core]* Models taken from initial pyspark core library.

Modeling assumptions This models takes numerical values as the features, so we encoded all of our features.

5.2 Generate test design

We splitted our dataset into train and test datasets with ratio 60/40. We specified a random seed for future reproducity.

Test design After the splitting we used test dataset for evaluating model performance.

5.3 Build model*[Core]*

Run the modelling tool on the prepared dataset to create one or more models.

Parameter settings For the decision tree we used maxDepth (is how deep tree will computed) and maxBins (maxinum amount of bins) to tune hyperparameters. **Models** – DecisionTree and RandomForest. **Model descriptions** – DecisionTree computes one tree. It may be very deep and have a lot of leafs. In RandomForest usually we got a lot of trees, such that most of them are not deep.

5.4 Assess model*[Core]*

We may use DecisionTree to enterpret result of the model by using set of the experts. In most of the cases DecisionTree works better but have low performance. In our case we most likely will use RandomForest because it has better performance.

Model assessment – After all the finetuning process we got f1 result 6.

Hyperparameters tuning: Decision Tree

	f1	maxDepth	maxBins
0	0.7042	5	16
1	0.7108	5	32
2	0.7145	5	64
3	0.7293	7	16
4	0.7346	7	32
5	0.7379	7	64
6	0.7669	10	16
7	0.7747	10	32
8	0.7808	10	64

Figure 7: Decision tree hyperparameter tuning.

Hyperparameters tuning: Random Forest

	numTrees	maxDepth	f1
0	10	5	0.6602
1	10	7	0.7155
2	10	10	0.7518
3	20	5	0.6735
4	20	7	0.7236
5	20	10	0.7588
6	30	5	0.6777
7	30	7	0.7285
8	30	10	0.7641

Figure 8: Random forest hyperparameter tuning.

Revised parameter settings – We got best parameters for both of the models. See results in 7 and 8.

Big data pipeline: Stage III*[Core]* After the one-hot encoding and numerical encoding we use usual numeric converting to get input for the model. Finally we got numerical features that we used to train both of the models.

6. Evaluation*[Core]*

Objective that we got are coorelated with our business goal, so we cover all the requirement from the business.

Assessment of data mining results – We may predict class of the accident, so in nearest feature we may assume that accident will happen or not. In total we got best accuracy with RandomForest model.

Approved models We got nice f1 - around 0.8, so we may use it to predict future car accidents. We may use this new predicted feature to use predict appearance of accident time.

Big data pipeline: Stage III*[Core]* Result of the assignment are located at folder outputs.

7. Deployment

We got model from the training procedure and after the applying all postproduction processing

Deployment plan – Summarise your deployment strategy including the necessary steps and how to perform them.

Big data pipeline: Stage IV*[Core]* Include in this section, the details of presentation stage.

7.1 Limitations and Challenges*[Core]*

What could have been investigated if given more time? What have been difficult when solving the problem and getting answers?

8. Contributions and Reflections on own work*[Core]*

The authors of this report and the contributors of the project are presented in Table 1. (**Fill in the contributions table.**)

Stages	Salavat Dinmukhametov	Timur Sergeev	**Total**
Introduction	50%	50%	1
Business understanding	50%	50%	1
Data understanding	50%	50%	1
Data preparation	50%	50%	1
Modeling	100%	0%	1
Evaluation	0%	100%	1
Deployment	100%	0%	1
Pipeline's Stage I	50%	50%	1
Pipeline's Stage II	50%	50%	1
Pipeline's Stage III	100%	0%	1
Pipeline's Stage IV	0%	100%	1

Table 1: Contributions table

8.1 Report summary*[Core]*

Write a summary of your report.

- We predicting type of the car accident. This information will be useful to predict time delay in the nearest future.
- We used both geographical and time-based features. To encode them we subsample information from time feature and find distance in geographical features.
- We split information into train and test set. After that we fine-tune model with Cross-Validation and Grid-Search.
- We used additional dataset to get geographical coordinations. We used nearest city to the position of the accident.
- We would be use BallTree, and not try to implement it from scratch.
- What else is there that you would have changed about this project? No

References

[1] Tim Berners-Lee, Robert Cailliau, Ari Luotonen, Henrik Frystyk Nielsen, and Arthur Secret. The world-wide web. *Communications of the ACM*, 37(8):76–82, 1994.

[2] University of Skövde. Regulations, forms and templates, 2022. URL https://www.his.se/en/research/doktorandhandbok/regulations-forms-and-templates/.

[3] John M Swales and Christine B Feak. *Academic writing for graduate students*. University of Michigan Press Ann Arbor, MI, 1994.

A. Appendices [Optional Section]

If your paper includes appendices, they are to be placed at the very end of your paper. If you have conducted a survey or interviews, the questionnaire and interview guide are to be included as appendices. If you have made some ML implementation, attach the code. If you are unsure about what to include as an appendix, consult with your lecturer or supervisor. Make sure to name the appendices correctly, like A, B and C.

A.1 LaTeX usage examples

This chapter is included solely as a help and reference for using LaTeX. It should not appear in the final document.

A.2 Introduction

This chapter shows examples of using LaTeXfor common operations.

A.3 Styles

Styles such as **bold**, *italic*, and underline can be applied to text. You can also use apply colors, and *combine* **styles**. It is recommended to use only bold to emphasize, and not abuse this resource.

A.4 Listings

With itemize you can create unnumbered lists:

- strawberries
- Peaches
- Pineapples
- Nectarines

Similarly, enumerate allows you to create numbered lists:

1. Prepare the memory of the TFG
2. Prepare the presentation
3. Present the TFG
4. Apply for the Bachelor's degree

A.5 Subsections

Subsections can be defined with the subsection command:

A.5.1 First subsection

This is a subsection

A.5.2 Second subsection

This is another subsection.

A.6 Images and figures

All images and figures in the document will be placed in the "fig" folder. They can be included as follows:

Figure 9: An example of image

Note that the figures are automatically numbered according to the chapter and the number of figures that have previously appeared in that chapter. There are many ways to define the size of a figure, but it is advisable to use the one shown in this example: the width of the figure is defined as a percentage of the total width of the page, and the height is automatically scaled. In this way, the maximum width of a figure would be 1.0 * textwidth, which would ensure that it is displayed at the maximum possible size without exceeding the margins of the document.

Note that LaTeX tries to include figures in the same place where they are declared, but sometimes this is not possible due to space constraints. In those cases, LaTeX will place the figure as close to its declaration as possible, perhaps on a different page. This is normal behavior and should not be avoided.

A.7 References

Notice how the "label" command has been used several times in the source code for this section. This command allows you to mark an element, be it a chapter, section, figure, etc. to make a numeric reference to it. To reference a "label", use the "ref" command including the name of the reference:

Examples of styles are shown in the A.3 section.

The subsection A.5.1 explains...

In Figure 9 we see that...

This saves us from having to directly write the indexes of the sections and figures we want to mention, since LaTeX does it for us and also takes care of keeping them updated in case they change (try moving this chapter to the end of the document and see how all referenced indices are updated automatically). Also, "ref" references act as hyperlinks within the document that take you to the referenced element when you click on them.

It is usual to name "label" with a prefix indicating the type of element to find it later more easily, but it is not mandatory.

A.8 Code snippets

Code snippets can be included via listing:

```
num = float(input("Enter␣a␣number:␣"))
if num > 0:
   print("Positive␣number")
elif num == 0:
   print("Zero")
else:
   print("Negative␣number")
```

Code extraction 1: Python code

A wide variety of languages are supported:

```
public class Test {
    public static void main(String[] args) {
        System.out.println("Hello,␣world!");
    }
}
```

Code extraction 2: Java coode

Code snippets can also be referenced via label/ref: Code snippets 1 and 2.

A.9 Links

You can link to an external website using the url command: https://www.example.com. A link can also be linked to text using the href command: example domain.

A.10 Citations and bibliography

In LaTeX, bibliography items are stored in a bibliographic file in a format called BibTeX, in the case of this project they are in "bibliography.bib". To cite an element, use the "cite" command. You can cite both scientific papers [1] or books [3] as well as

web links [2]. Citations are automatically numbered and included in the bibliography section of the document.

Note how bibliographic items stored in "bibliography.bib" have an associated tag, which is the one included when citing them using cite. Adding a reference to the bibliographic file does not make it automatically appear in the bibliography section of the work, it is necessary to cite it somewhere in it.

A.11 Equations

LaTeX has a powerful engine for displaying mathematical equations and an extensive catalog of mathematical symbols. The math environment can be activated in many ways. To include simple equations in a text, they can be surrounded by dollar signs: $1 + 2 = 3$, $\sqrt{81} = 3^2 = 9$, $\forall x \in y \ \exists \ z : S_z < 4$.

More complex equations can be expressed separately and are numbered: equation 1.

$$\lim_{x \to 0} \frac{e^x - 1}{2x} \overset{\left[\frac{0}{0}\right]}{\underset{\text{H}}{=}} \lim_{x \to 0} \frac{e^x}{2} = \frac{1}{2} + 7 \int_0^2 \left(-\frac{1}{4} \left(e^{-4t_1} + e^{4t_1 - 8} \right) \right) \, dt_1 \tag{1}$$

There is here an extensive list of symbols that can be used in math mode.

A.12 Special characters and symbols

Some characters and symbols must be escaped in order to be rendered in the document, as they have a special meaning in LaTeX. Some of them are:

- The dollar sign $ is used for equations.
- The percentage % is used for comments in the source code.
- The euro symbol € often causes problems if typed directly.
- The underscore _ is used for subscripts in math mode.
- Quotes must be expressed 'like this' for single quotes and "like this" for double quotes. Spanish quotation marks can be expressed "like this".
- The backslash \ is used for LaTeX commands.
- Other symbols that must be escaped include the braces { }, the ampersand &, the hash #, and the greater-than > and less-than < symbols.