Neural network tool to predict the matching score between a student from Code Gorilla and an employer

De Jong Anne, Licence/Bachelor of Computer Science

University Claude Bernard Lyon 1

09-05-2021 until 14-07-2022

Company	Supervisor company	Supervisor university
Code Gorilla Laan Corpus den Hoorn 106,9728 JR Groningen Tel +31 627896716 Email: info@codegorilla.nl	Chief Technical Officer Bart van Der Voort van der Kleij Tel +316 27896716 Bart@codegorilla.nl	Elodie Desseree Elodie.desseree@univ-lyon1.fr

1. Acknowledgments

Before I start with the content of this rapport, I would like to take this opportunity to thank the people that have made this internship possible.

Let me begin by thanking my supervisor, Bart van Der Voort van der Kleij as Chief Technical Officer at Code gorilla, for the guidance during the internship. And the weekly meetings were efficient and straight to the point. And for the flexibility in working hours, which allowed me to train multiple times a week.

Diem Do, CEO of Code Gorilla, is someone I would like to thank for making me feel welcome and appreciated from the first day. And to be included in the pitch and put on the company event. For the following compliment, I also need to include Johanna Spiller, CEO of Femture. Thank you for showing me what badass women in business look like.

And special thanks to the **Femture team** for including me on the Femture assessment day. It was terrific, and I have learned a lot. To see all these women working on a technical problem was genuinely inspiring.

My last thank you must go out to the entire office of Code Gorilla and Femture for the kickass Friday afternoon drinks. And thanks for showing me that I need to work on my Nintendo switch skills. You all have inspired me to participate in making a more inclusive workplace. And because of you, this internship has been a valuable experience for my professional career.

Figure 1 – Crowded office

2. Index

1. Acknowledgments 2
2. Index 3
3. Figures 3
4. The topic of the internship 4

Goals 4

The process 5

Reflexion 6

The Interest of the company 7

Process and results 7

5. Professional environment 8
6. Work description 9

Normal working day 9

Femture assessment day: 10

7. Findings of the experience 12

Skills learned 12

Integration/human relations 13

8. Conclusion 14

9. Cited works 16

10. Annex 17

3. Figures

Figure 1 – Crowded office 2

Figure 2 - Woman sitting on the floor 7

Figure 3 - Git History 9

Figure 4 - Femture logo 10

Figure 5 - Photo Femture day slogan 11

Figure 7 - Femture day explaining the assignment 11

Figure 6 - Femture day evaluating the presentations 11

Figure 8 - Talking in the office 13

Figure 9 – Woman’s office 15

The final assignment of the license/Bachelor of Computer Science is an internship. This internship has an average length of 12 weeks. This internship aims to discover how computer science has a place in the business world. On the LinkedIn platform, my network introduced us to each other. After multiple online conversations, the internship could start on nine May.

4. The topic of the internship

Code Gorilla has gathered information about its students. The **goal** of the internship is to **collect** all the **knowledge** present and to make a **prediction**. This prediction should give a score on whether a student is a good match with a company. This system should be able to predict, par example, the following:

- Which person has a good chance of **succeeding** and who needs extra guidance in a specific area
- Which person has by which kind of companies the most success, and by which **type compagnie** they shouldn't apply.
- For which person, it is clear that they need to have more **guidance** after they complete their education.

Goals

The goals that were predefined are:

- find out what Code Gorilla **needs**. Tell us which process a candidate needs to complete and which data we need to make a **clear education profile**. On which we can test future candidates.
- Implement a **profile categorization** model: The design and building of a profile categorization network for categorizing candidate profiles.
- Portal candidate dashboard: **summarize** the **processes** above on the dashboard. So that the educators can give an insight into the new candidates.

The tools, language, and programs will be discussed during the first week of the internship.

In this discussion, we decided to use **python** for the data science part. This part represents the making of the model. For this model, we wanted to use a **clustering** method. So that when we are forming the model, we could see which clusters of companies would fit with a particular set of students.

After approximately three weeks, we came up with an idea to integrate the neural network into a **tinder-like application**. This web application will be built in Vuejs because this is the coding language used frequently in this company. And therefore, the most logical language to use in this project. The end-product will be presented in a **presentation**. This presentation will answer all the goals mentioned above. The product will be a **prototype** with fictional **users** to demonstrate its functioning.

The process

The main goal of this internship is to **explore** a potential **tool** for Code Gorilla. This tool is supposed to predict if there is a match between a student and a company. I worked alone on the project. But I had to collaborate and gather information. For example, at the beginning of my internship, I needed to **understand** the **database**. And what the values in that database represent. For that, I needed to talk to different people within the company and talk about their **expertise**. After this "exploring" phase, I started to analyze the database. And began to brainstorm about how to transform the existing database into a database I could use for my neural network.

During my experimentation with the data, I did a lot of research into **neural networks**. And how I could use this method to predict a **solid match** between a student and a company. A neural network is a method of constructing a model to predict an outcome. It makes this prediction based on all the data in de database. For the neural model, the column's name isn't critical, only that it has a numerical value that it can interpret.

For this internship, I needed to predict a **scoring value**. But to make a valuable predicting model, I needed more diverse data. For that, I mainly talked to Brendon. Brendon takes care of the business side of Code Gorilla. With him, I brainstormed over the possibilities. But in addition to finding more data, we came up with an **idea** for a **web application**. We decided to work this idea out in the following brainstorming sessions.

After careful consideration, we implemented the neural network model into a **tinder-lookalike** web application. After talking to my internship supervisor, we decided to make a prototype website in Vuejs (this is a programming language). The backend of the web application is going to be with **firebase**. Vuejs and firebase were **new** programming tools to me. So, it took some time to figure out how to use them properly and efficiently.

Firebase has some excellent features that can be useful for creating a **complete** web **application**. It's a development platform where developers can mix and match the components to their product. For this web application, I will be using the following parts:

- Authentication
- Firestore database
- Storage
- Machine learning

Each of these is used to ensure the web application is functioning. The authentication keeps track of the user accounts created on the website. A fundamental feature is password encryption. Firebase takes care of the management of the passwords all by itself and encrypts the password without the developer interfering.

The Firestore database stores additional information, like roles on the website, usernames, telephone numbers, etc. This database is a **no SQL** database, meaning the queries[1] are called in predefined functions and not in code. In addition, to this database, we use storage. Serves to store, for example, the avatar picture of a profile.

The last element of firebase is the main idea of the internship. The machine learning module allows developers to implement their **machine learning model**. I'll create this model with TensorFlow, but more about this later. The machine learning module helps with hosting and further development of the model. This saves a lot of reconfiguration and effort because firebase handles the model's hosting and deployment.

Reflexion

During the internship, I needed to **learn** Vuejs, a programming language based on JavaScript. The programming in Vuejs was complex initially because I had **never used Vuejs**. So, there was a bit of a **learning curve**. The same goes for all the extensions used in the Vuejs application. Each extension has its way of being implemented and used. Especially the Firebase development platform was challenging at first. Since the security rules first denied all access. And if you are unaware of these rules, your code will not work (even if it's written correctly). But after a month, I was comfortable with the new tools and could use them correctly. And adapt its functionality accordingly.

For the model creation, I needed to use a **bibliography** called **TensorFlow**. I had heard of this bibliography before and always wanted to use it. But this bibliography was quite tricky. Even though they say on their site that this bibliography helps you make "simple" models. But I didn't want to give up because this topic could be helpful in my future academic career. So, I decided to follow a **crash course** in **neural networks,** which uses this bibliography to train its models. This crash course had a duration of 8 hours. After I passed the final test for this crash course, I felt more **confident**. Now at least, I knew how to start building a neural network.

Another **obstacle** was converting the existing **data** into data that I could use. I didn't want to redo the current database because that would be a waste of time. First, I tried to **duplicate** the **data** and store it in a separate file. And then implement that file into my "correct" database, except Firestore doesn't accept individual files. Firestore wants you to add a document when you create it. After another brainstorming session to find a **solution**. We decided to **change** the **approach**. The existing data would be training data for the model. This could be exported into an excel file and used to train a neural model. And all the **data** that is going to be gathered in the **future**. It would be stored in the **new database** in Firestore. So that from now on, the database will be compatible with the neural network. This solution was the best for both sides. And gave me also the opportunity to think about the construction of the database. Of which I learned a lot.

[1] A query is a request for information stored within a database management system

The Interest of the company

The matching tool was the idea of the company. The company can use this tool to improve **efficiency**. Because now an **employee** needs to **find** a good **match** between a student and a compagnie. This tool can make the match without the employee's assistance. The neural network can **pre-approve** a match. Now the employee can start to evaluate the match. Before using this tool, the employee needed to identify multiple possible matches and find the right one. The employee can take more time to **assess** a **single match** because the neural network gives a coefficient as a matching score. The employee can work from top to bottom. Where at the top, the highest coefficient represents the highest probability of a working match. This probability will decrease as we descend. The lowest **minimal acceptance** matching score is **0.5**.

Process and results

Process

The resulting product is a single-page interface. This means that a *website or web application dynamically rewrites the current web page with new data from the web server instead of the default method of a web browser loading entire new pages*. (Lawson, 2018) This way of coding **improves** the **user experience** because of the rewriting speed. Within this web application is the **regression prediction model**. This model is being evaluated with a subset of the data. We only use 80% of the data when we train the model. The other 20% we use to test the accuracy. This results in a number between 0 and 1. And during the tweaking of the model, we re-run this accuracy test. And this is how we prove that our model is working. The accuracy score will change throughout the different testing phases. And my interpretation of the accuracy score will influence how I change the testing conditions to produce a higher accuracy score.

Results

At the time of writing this report. There aren't many results to be shared. Because of the size of this project, there isn't much to report after a month of work. Because this month wasn't all programming, it was majorly exploring the tools and languages and researching the assignment. The research and exploration that I did are in the annex. In the following month, I'll be dedicated entirely to programming.

Figure 2 - Woman sitting on the floor

5. Professional environment

Inclusive labor market and **enrichment** of **ICT**. These are the keywords important for Code Gorilla. Code Gorilla is a company in Groningen, the Netherlands. They aim to help make the labor market more inclusive. They try to achieve this goal by **educating** people who have a distance from the labor market. Code Gorilla gives on-location education to the participants. There are different kinds of education: boot camp and basecamp. The difference between these two is the length and content.

During these education periods, the participants are **trained** in **basic computer skills**. And they get lessons from skilled professionals in, par example, HTML, CSS, and JavaScript. If the education has been **completed successfully,** Code Gorilla accompanies the students to a job. This is done by giving these students access to job coaches, additional training, and counseling. With all this help Code Gorillas **supports** the **participants** in finding the **right match** between the **company** and the participants.

An important observation is that these participants aren't just "normal" people. The people that participate in these educations have all **different backgrounds**. But they are all students who participate because they want to be re-educated to go into the IT domain. Code Gorilla helps them to cope with their eventual problems. And gives them an environment where they can use their coding skills. Besides, Code Gorilla allows them to work in teams.

Code Gorilla has won multiple awards like

- FD Jonge Talenten (2020). This is an award from the Financieel Dagblad. A Dutch daily newspaper focused on business and financial matters.
- IGNITE Award (2019). The goal of this award is to help social start-ups to become more professional.
- Start-up Award (2018) is awarded to innovative companies in the Netherlands that have solid technological developments and win an international impact.
- Social Impact Award 2017 is a reward for companies aiming to use their platforms to better the community.

Code Gorilla's office is divided into flex places, fixed places, and the education rooms. Every **room** with fixed places is used by **one team**. This team works together on the same problem or project. Then the significant/higher placed employers have their own office. The education rooms are used by the instructors and the participants of the boot camp and basecamps.

The company has 22 employers. Most of them are coding, marketing, and coaches. Until now, Code Gorilla has given **more than 120 people a job**. Finding the students a company (that is what we call the people who have completed a boot camp) is problematic for two reasons. One of the reasons is that the **labor market** now is exceptionally **challenging**. Most companies are **hesitant** to **hire** new staff. And indeed, for people with little or no experience. Secure a good match between a company and a student depends on many factors.

6. Work description

Normal working day

My working days had the same structure. This internship aimed to discover if a neural network tool could be helpful for Code Gorilla. In my opinion, to find out if something is **valid**, you must **build** it and see if it **works**. I divided my workdays as follows:

- Start: I would start with **updating** my **to-do list** for the day. And seeing which to-do's have priority above the other ones. And planning on how to finish them the most efficient way.
- Mid-section: executing the planning. And after completing a to-do, I would save my progress in **Git**. Which is software for tracking changes in files. I would never make a commit when the to-do wasn't finished. So that the git history was clean. Here is a **snippet** of the git **history**[2] in the table below.
- Afternoon: usually, I would **lose** my **concentration** around 3 pm. So, to make the most out of this timeframe. I would use this time to **write** this rapport. Or I would document the research I had done to finish my to-dos.

COMMIT ID	COMMIT DESCRIPTION
86df148	login, logout, register new user check
4e27c83	Redirect after registration of new user
af76ae2	Adding of scoped in <style> attribute
d348cc1	Add user functions
f5859c1	Start user registration
2221dcb	Save before I add firebase to this project.

Figure 3 - Git History

[2] This is the translated history. All the git commentary is written in Dutch.

Femture assessment day:

Figure 4 - Femture logo

Besides the regular working days, I also had the opportunity to **participate** in the **Femture** assessment day. First, let me tell you more about Femture. The mission of Femture is best described on its LinkedIn page:

Our mission is to close the tech talent gap by recruiting, re-training, and retaining female tech talent. In our 4-month Femture Bootcamp, we create a safe and empowering learning environment in which women learn how to code and get trained in professional skills and soft skills by senior women in tech. We want to up the percentage of female talent in tech and set young women on their path to becoming tomorrow's CTOs and tech entrepreneurs. (Femture , 2022)

The assessment day can be seen as a kind of **character test**. During the day, the candidates are closely **monitored** while executing the **assignment**. The assignment for this assessment day was to create a text-based game. This needed to be realized in **teams** and within three hours. The functioning and environment of the teams were more important than the product. At the end of the three hours, the teams needed to give a **presentation** to present their final product. Also, this presentation needed to be self-reviewed about how they experienced the teamwork and overcame their difficulties. This is the description of the day on the candidate side. But now, let me tell you the role I played on this day.

During the assignment, I was a **tech**-**mentor** and **observer**. First, let me explain the tech-mentor role. During the assignment, the teams had two credits, which could be used to ask **questions** to the tech-mentor. These questions could be about everything they wanted. But during the day, most questions were about how to get the code working (JavaScript and CSS) or the end presentation.

The second role was the observer. All the people that were assisting during the day were observers. There was an office that we called “the war-room” for this day. A whiteboard was divided into three spaces: yes, maybe, and no. On this board was where all the photos of the candidates were placed. During the day, we would move these pictures on the board. And on the end of the day, we would **decide** who could **enroll** in the Femture boot camp and who wasn’t ready. It is worth noting that we only **judge** people on their **functioning** in the **teams**. And how the candidates deal with **problems**. And not on the product they presented at the end of the day. For us, the process was more important than the result. Another critical element was how the candidates would react to feedback. All these observations were taken to the war room and discussed with the other observers.

Reflection on the assessment day:

Figure 5 - Photo Femture day slogan

This assessment day impressed me because I have never seen so many women coding and working together in teams to solve a technical problem. Usually, in hackathons or even at the university, the women coding are a **minority**. And to be assisting on a day like this and seeing all those women **wanting** to get a **career** in **technology**. It lifted my spirits and gave me a lot of energy to continue my career in tech.

I had never evaluated people before and with the Femture Assessment Day, I learned something which can be helpful for **future evaluations**. At first, I did not know what kind of trait to look for or which behavior was not expected from the candidates. While observing, it became clear that I recognized a lot of character traits that I also possess. I then knew what people on the observer side were looking for. I should work on some of these character traits to enlarge my chances.

Another **vital skill** that I learned from this assessment day is **communication**. When the candidates asked for my technical opinion. I needed to respond in a **correct pedagogical** way. Because most of the time, just giving them the solution wouldn't help them in the learning process. So, something I needed to be a little more **cryptic** about the answer. And try to **guide** them to the **right** solution. I also remembered that most of these women didn't have coding experience. So, the coding explanation must be in non-programming language (much like this rapport). I learned from this way of thinking how to help solve a problem with my coding experience. And throughout the day, I improved my way of explaining the solution.

Here are some photos were taken on the assessment day to give an impression. Pictures are made by Stella Dekker.

Figure 7 - Femture day explaining the assignment

Figure 6 - Femture day evaluating the presentations

7. Findings of the experience

Skills learned

I gained skills in two domains: **web development** and **python**. Let's first discuss the web development part. The website application is constructed with Vuejs. *Vue.js is an open-source model–view front-end JavaScript framework for building user interfaces and single-page applications* (Wikipedia, sd). Vuejs uses typescript, a superset of JavaScript maintained by Microsoft. I used the Vuejs **framework** to **create** the **website**. On this website, the neural model will **gather** its **information**. And displays relevant information to its users. To store this information, I used Firebase.

Firebase is a **cloud computing** and development platform from Google Cloud. From this platform, I deployed multiple functionalities like authentication for users. This (pre-made) process enables me to **focus** on the **primary goal** of this internship. And avoids losing time with hashing and hiding user-sensitive information. Google Cloud services take care of this for me.

The information I am gathering is stored in the database of firebase called Firestore. This is a **no-SQL** database, meaning getting information can be written in JavaScript instead of SQL. The same goes for the storage inside the firebase. This allows me to store, for example, a user photo and other data that is more than a few keywords. The **model** I have made is stored in the **machine learning** part of firebase. This part handles the deployment of my model and saves me the trouble of creating an application programming interface.

This model is built with **TensorFlow**. TensorFlow is, as its website says, "an end-to-end open-source machine learning platform." Loosely translated into non-technical language, this **platform** gives a **complete** and **functional solution** without services from other companies or websites. Within TensorFlow, I mostly used Keras, a subpart of the TensorFlow that permits beginners in machine learning to build relatively easy models. They also provide **clear** and **helpful documentation** to get started.

For the training of the model, I needed a lot of data. But in this case, I didn't have a lot of data available. And therefore, I needed to **create** some **fake** accurate **data**. This means data with the **right proportions** that doesn't mess up the **dynamics**. For this, I used **tabular data augmentation**, which gave me enough data to train the model.

In addition to learning the skills mentioned above. I followed a **Kaggle introduction cursus** into deep learning. In this cursus, I learned the **basics** of **neural networks**. And how they can be constructed using Keras. Additionally, in this cursus, they explained the **different layers** of neural networks.

Working methods

Aside from libraries and coding, I also participated in a **scrum** training. Scrum is a framework for **developing**, **delivering**, and **sustaining products** in a complex environment, with an initial emphasis on software development (Wikipedia, 2022). The training was planned for Code Gorilla students, but extra spots were free. So, I took the opportunity, and I **participated** in a training. Before this training, I only heard of other ways of developing a product like **agile**. But I believe that scrum is a **more efficient** way of coding in a project. Because it can **adapt** more manageable to the **demands** of the clients, I will try out the scrum approach in a future team project.

Apart from the **technical** working methods, I've also experienced the **stand-up**. This is a kind of meeting that happens **once** a **week**. In this meeting, everyone stands up, as the name indicates. During this meeting, everyone needs to give a **quick summary**. This summary consists of a short **recapitulation** of their work **last week**. And give a quick **update** on what they will be doing **next week**. Also, this would be the perfect meeting to **introduce tasks** you need help with, for example, in my last week of writing this report. In the stand-up, I asked if someone wanted to **proofread** this report. At the end of the meeting, the tasks are divided and written on the board.

Integration/human relations

During this internship, I tried to **learn** as much as possible **from** the **people** around me. Most people in the office were older than me. And have had other jobs before the current one. During the lunch breaks, I would get to know them. Here at the office, the whole office **eats lunch together** at noon. The food is placed on big tables in the kitchen. And then everybody prepares their meal. Because of this way of lunching, I quickly knew the whole office.

Another thing I **observed** when I was working in the flex workplaces is the **way** of **asking** for **help** or for someone's time. Most employees plan meetings to answer each other's questions, resulting in efficient communication. And for the short questions are posed on **Slack**. Slack is a communication platform designed to help communication in the workplace. And, of course, there is still the "old" way of asking a question; walking by someone's desk. When this last way of asking is chosen, people still ask first if the employee in question is busy or not.

Figure 8 - Talking in the office

8. Conclusion

To summarize the focus of this report in one sentence, I would give this: predicting a match between a student and a company with the assistance of a neural network. A neural network is a deep learning method used for regression prediction. Regression means that outcome of the forecast will be a number. For this assignment, we want to predict a matching score. This matching score would reflect the odds of the compatibility of a student and a company. This neural network is built with different layers, each a different task. Each of these tasks improves the way that the network is learning. And how the better the network learns, the easier it adapts and gives the correct forecasts. But of course, this neural network isn't valuable alone. That's why we decided to build a prototype website around the neural network after some brainstorming sessions. This allowed me to learn Vuejs, a new programming language for me. Google Firebase supports the web application. Firebase handles all the user authentication. And allows me to deploy my custom model and use the data stored in the firebase store and storage.

At the moment of writing this report, there are not many results to be shared. There is a baseline web application created with Vuejs and corresponding extensions. But this one is still in the developing phase and therefore isn't ready to be shared yet. As I have mentioned before, the first month of this internship is mainly spent researching and discovering the tools I will use. But I could answer most of the questions posed in the beginning, what Code Gorilla needs before everything is a consistent database. This will be constructed through the website, where every user must create an account. They fill out a questionnaire depending on their role (Business or student). This allows Code Gorilla to quickly gather data and directly deploy it to the neural network behind the web application. And with a tinder-like application, the users and the businesses can see the matching score. Besides the company and student role, there is the Code Gorilla role. This allows visiting a dashboard with the newest users and the statics.

At the beginning of the internship, we thought about the profile categorization model. This can be done with a different clustering model. I did most of the research for this problem and documented it for the company in a document. This document can be found in the annex. To give a quick summary of this document: clustering can be done in different ways, but for categorizing users and businesses, it is easiest to use either the Gaussian Mixture Model or K-means. The silhouette coefficient and Dunn's index can evaluate these clustering models. These two evaluations give a numerical value in return, which helps modify the clustering.

The last question posed by Code Gorilla is about the portal candidate dashboard. The neural network tool isn't going to be implemented in the already existing candidate dashboard for time reasons. We prefer to finalize the web application with the neural network. The advantage of this approach is that we know in advance if this neural network works. And if Code Gorilla is optimistic about its functioning, it will be easier to adapt the neural model to be displayed in the dashboard.

If I were to have more time, I would make the front end more modern. For now, the design of the web application is basic. Because we are more interested in the code behind the mechanism, making the front end more modern could quickly improve the user experience. Besides the front end, for now, this web application will only be used by Code Gorilla. But if it was going to develop this platform further, it could be used by more people and businesses.

This internship has allowed me to observe the working culture and assess if this is the job I want to do when I finish my master’s. The topic’s that I had to use in this internship—where the majority of topics that hadn’t been covered in my courses at the university. Because there are the topics that I was going to study in my master of Artificial Intelligence at the University of Linz, this internship has permitted me to check out if this part of Computer Science fits me. And I’m glad to report that I have made the right choice for my master’s.

If I look even further in the future, I would like to see if I can implement neural networks in autonomic drones. The knowledge that I have gathered about neural networks. It makes me curious if this can also be implemented in drones. But I can research this idea and perhaps even develop it later in my academic career.

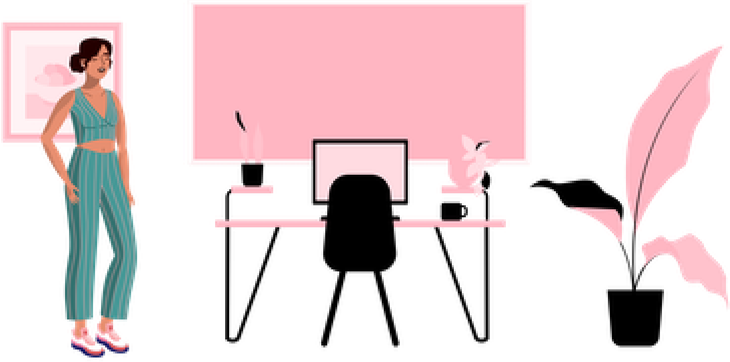


Figure 9 – Woman’s office

9. Cited works

Femture . (2022). *Femture*. From Linkedin: https://www.linkedin.com/company/femture/about/

Lawson, K. (2018). *What's Single Page Application Architecture? How Does it Work?* From bloomreach: https://www.bloomreach.com/en/blog/2018/what-is-a-single-page-application

Wikipedia. (2022, 05 26). *Scrum (software development)*. From Wikipedia: https://en.wikipedia.org/wiki/Scrum_(software_development)

Wikipedia. (sd). *Vue.js*. From Wikipedia: https://en.wikipedia.org/wiki/Vue.js

Besides these websites that I have cited in this rapport, I have used (and will be using) during my internship the following documentation:

- TensorFlow documentation, used for the creation of the model: https://www.tensorflow.org/api_docs
- Keras documentation, used for the neural network layers: https://keras.io/api/
- Pandas' documentation, used to prepare the data for the training of the neural network: https://pandas.pydata.org/docs/
- Vuejs documentation for the creation of the web application: https://vuejs.org/guide/introduction.html
- Vuex documentation for the retention of the user's data while their connection to the web application: https://vuex.vuejs.org/guide/
- W3schools, used for the CSS styling of the web application: https://www.w3schools.com/cssref/default.asp
- Firebase, as explained in this report, is used for the backend services of the web application: https://firebase.google.com/docs
- Stack overflow to help with resolving errors.

The cartoon pictures are made on blush. Design and from the collection Women Power are accessible through this link: https://blush.design/fr/collections/wvg442lqogOhDCjGCJTw/women-power.

10. Annex

Research into clusters and neural networks

This research is part of an internship at Code Gorilla. For this internship, I'm going to explore a potential matchmaking tool. This tool will make a match between a student and a company; this will be done with a neural network and clusters. In this research summary, I'm going to present my findings.

Content

Clusters....................1
- K-means....................1
- Gaussian mixture model....................2
- Spectral clustering (SC)....................2
- Hierarchic clustering....................2
- Evaluation cluster methods....................3
  - Data analysis....................3

Data processing....................3
- Data augmentation....................4
- Categories....................5

Neural networks....................6
- Creation of the neural network....................6

Bibliografie....................8

Clusters

Data clustering is part of machine learning. Machine learning can be divided into supervised and unsupervised learning. Supervised learning is when a model learns a relation between labeled input and output. When using unsupervised learning, there isn't a requirement for labels. Then the model learns from non-preprocessed data. This way of education focuses on relations and patterns. For this project, we are looking for clusters. (Seldon , 2021)

The advantage of unsupervised learning is the approach for which you need to get to know your database. The way of working is as follows. First, the data scientist inserts specific parameters for the first version of the model. Then the model can analyze the data itself. And with this method, the model learns to look for trends and relations.

An important aspect to notice is that all the methods of the bibliography sci-kit learn cannot process NaN[3] or empty values because the method needs numerical values. A solution for this problem is to ignore the column with the NaN value. But this way, you quickly lose a lot of data. Another solution is to use *sklearn. impute*__. This function replaces empty values with statistics like mean, median, or most frequent value in the column.

Another step in the processing of the data is the scaler. If your data contains a lot of highs or lows. Then you need to apply the bibliography of sci-kit to learn what normalizes these values. This scaler removes the median and scalers the data to the quantile range. Another way to do this is to use the mean and the variance. (Arvai, sd)

In the next part, I'm going to describe different cluster methods. And with these methods, I'm going to experiment with the data of Code Gorilla.

- K-means
- Gaussian Mixture Model
- Spectral clustering
- Hierarchic clustering

K-means

This method consists of a few steps—first, the algorithm chooses *k* centroids (here is *k,* a value you have determined yourself). Then the algorithm visits all the points. And for every point, the algorithm adds the point to the closest centroid. And recalculated the new centroids of the clusters. These last two actions keep repeating themselves until the centroids stay the same.

This disadvantage of k-means is that the choice for the number of clusters is made before the algorithm is lanced. The elbow method and the silhouette coefficient can choose the number of clusters. For both ways,

[3] NaN stands for Not A Number. And also indicates the missing values.

you lance the k-means algorithm multiple times. And then you place the outcome in a graphic. And from this graphic, several *k* can be deducted.

For the elbow method, you use the kmeans.ineratia_. And you search in the table to the tipping point. For the silhouette, the silhouette_scores are saved. The silhouette score is a combination between other points in the cluster and the distance to the other points in the clusters. These scores vary between -1 and 1. The higher the score, the better it is.

Gaussian mixture model

The model uses a standard Gaussian distribution, as its name implies. This method combines multiple normal distributions. And uses, therefore, factors for each normal distribution. This results in a graphic like below where every distribution has its color.

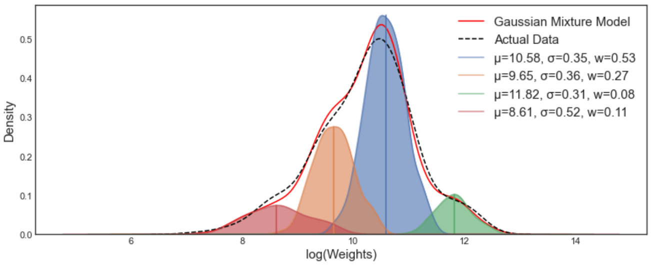


Figure 10 - visual explanation of the Gaussian mixture model

Where k-means falls short, we could support with the Gaussian mixture model. K-means is a hard clustering method. This means that a point belongs to one cluster. And there is no indication of how many specific data points belong to a cluster. For that, we need a Gaussian mixture model. (Carrasco, 2019)

Spectral clustering (SC)

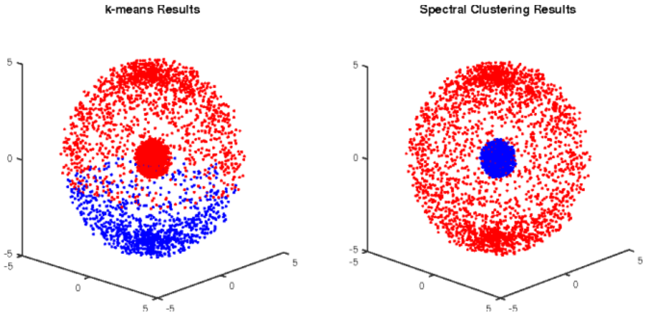


Figure 11 – The difference between k-means and spectral clustering

The difference between spectral clustering and k-means is in the distribution of the points. A spectral cluster can recognize if there is a circle within a circle (see the image below). The left graphic is the result with k-means. And for each point, the method calculates to which cluster it belongs. But therefore, the circle is divided into two parts. But when you use the spectral method, the method recognizes the circle's middle point as more density. And that the central points a proper cluster is.

Hierarchic clustering

This way of clustering is based on a dendrogram. A dendrogram is a tree-like structure representing a system's relation. This has an entirely different approach than other clusters between data pointing methods. This method starts with multiple clusters, one for each data point. And for each step, two clusters are merged. Which two clusters are being merged depending on the

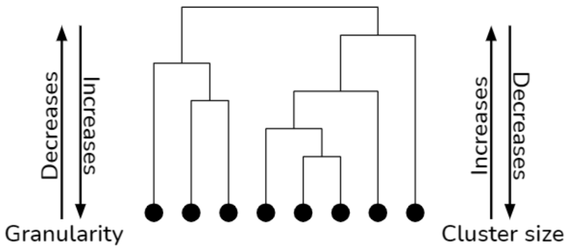


Figure 12 - Explanation of a dendrogram

subtype within the clustering? Subtypes that are mentioned in the documentation:

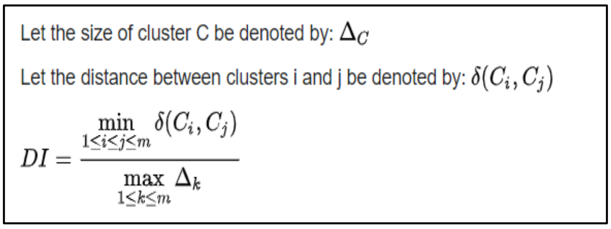
- Ward: has a purpose as slight as a possible standard deviation within all clusters
- Average: minimizes the average distances between all observation pairs of clusters.
- Single: minimizes the distance between the closest observations of pairs of clusters.

(Pedregosa, 2011)

Evaluation cluster methods

Data analysis

After we did a data analysis with the abovementioned me, we need to evaluate cluster methods represent the correct clusters. And which methods aren't ideal to use. The silhouette coefficient and Dunn's Index (DI) are the most common evaluation methods. The Silhouette coefficient is defined with the following mathematical formula $s = \frac{b-a}{\max(a,b)}$ where a is the average distance is between a point and all the other points in the same cluster. And b the average distance is between a point and all the other points in the closest cluster (This is not the cluster of the point itself).



Let the size of cluster C be denoted by: Δ_C

Let the distance between clusters i and j be denoted by: $\delta(C_i, C_j)$

$$DI = \frac{\min\limits_{1 \leq i \leq j \leq m} \delta(C_i, C_j)}{\max\limits_{1 \leq k \leq m} \Delta_k}$$

Figure 13 - Mathematical formula Dunn's Index

The score that comes out of this calculation is always between -1 and +1.

- -1 -> compact clustering
- 0 -> overlapping clustering
- +1 -> wrong clustering

Dunn's Index is a value that indicates if a cluster is compact or if the cluster has distanced itself enough from other clusters. The higher the DI, the better the clustering.

Data processing

"A more general rule of thumb is that the number of observations should be proportional to $n = \frac{1}{d^p}$ where p = number of features and d = the maximum spacing between consecutive or neighboring data points after each feature is scaled to the range 0-1." (Sevey, 2017)

If we want to follow this general rule of thumb, the dataset needs to be much bigger. This we can achieve by data augmentation. In the following parts, I'm going to elaborate about:

- Data augmentation
- Normalization
- Categorization

Data augmentation

The data available in Code Gorilla is limited. So, to make an efficient and correct model, we need to augment the amount of data. Within data science, there are multiple technics. The majority of these technics are for image and text generation. But I need augmentation for tabular data. As a goal to eliminate underrepresented groups. So that the model is well trained

First, we initialize a training model with the data we have. From this model, we take the latent variables' mean or variance. Latent variables are variables that aren't directly observed but variables that are a derivative of a mathematical model. These latent variables are the starting. With these variables, we call the *predict_df* function.

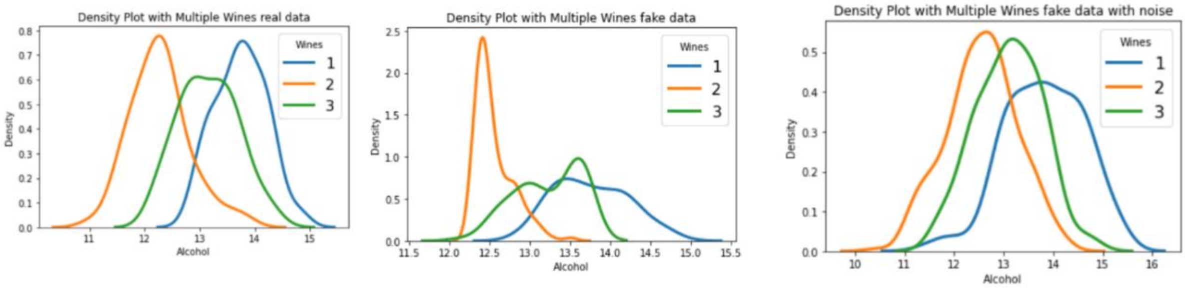


Figure 14 From left to right: the actual data, the fake data without noise, and the fake data with noise.

This function generates fake data. For the columns of which the type is an integer, we need to change the type with the help of the NumPy library. The same goes for the categorical columns. This is the first way how we can augment the data. A little bit more advanced way is to use the *predict_with_noise* method. This method takes a standard distribution sample and multiplies that with a number + 1. This is close to the PCA method. The PCA method is a statistical procedure that summarizes the information into smaller sets with indexes. This helps with the analysis. (Satorius, 2020). The *predict_with_noise* method generates data that is closer to the real world. From the GitHub notebook, the author shares the following graphics to show the connection between the three variances.

These graphics show that the fake data with noise is close to the real data distribution. The conclusion from another notebook that is wholly focused on using fewer latent factors shows that the data is more spread and, therefore, further away from the original data. So, for this project, latent factors need to be equal to the number of columns of the dataset.

Vanuit deze grafieken is te zien dat de neppe data met ruis heel erg lijk op de verdeling van de echte data. De conclusie vanuit een ander notebook die volledig gefocust is dat als je minder latente (Schmidt, 2021)

Categories

Data in neural networks are numerical values that are normalized. That means that the values are between 0 and 1. But the results that we expect from this network are not numerical. The same goes for the input.

In data science, converting raw data to a type with which a computer can do calculations is always challenging. Data is This is because the real-world data isn't perfect. A lot of values are missing. There are a lot of not numerical values. And most of the data is split into categories. For the transformation techniques, we can choose two kinds: deterministic and automatical techniques. (Adeel, 2021)

For the deterministic, we have the following :

Name	Description	Example
Ordinal encoding	Every category gets a number.	
One-hot encoding	Every category has a place in a vector.	Image that the vector has 3 values: (0,0,1) which means that it's category 3, (0,1,0) means category 2, etc.
Leave-one-out encoding	Every category is the mean of the variables with the same value.	
Dummy Variable Generator	Each category gets a unique binary value.	
Hash-based encoding	Numeric values are generated through a string type by applying a hash algorithm.	A good hash function is SHA-256, which gives each input a 256-bit output.

(Gosh, 2018)

Next to these methods, we also have an automatic encoding that uses intermediary images. These images are then pulled through a different classification model. This generates an output. The advantage of automatic encoding is that the algorithm handles the missing values. But for this project, the amount of too small to apply this algorithm.

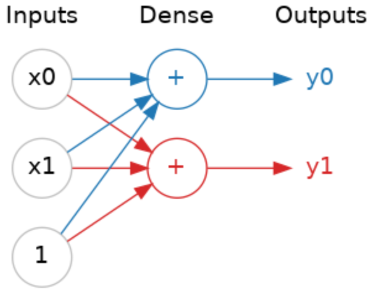
Neural networks

Neural networks are part of the “deep learning” technique. The main goal of machine learning is to find useful information out of data. And that machines are learning to do this more efficiently through experience. Let’s get back to deep learning, their neural networks are used to replicate a human brain. The “deep” represents the layered structure in these networks. The more complicated the functions, the lower in the network. Each layer is divided into neurons that each have a simple task. In this part, I’m going to describe the steps of creating a neural network

- Transformation of data
- Creation of a neural network
- Training of the neural network

Creation of the neural network

The Keras module is an API made to make machine learning accessible for people. Keras minimalizes the number of actions. And gives understandable error messages. Keras is for this project a good choice because the API is made for experiments and has clear and extensive documentation. Keras is also used a lot on Kaggle. Kaggle is a daughter enterprise of Google and an online community for data science and machine learning. I have followed on Kaggle a mini cursus that explains how Keras can be used optimally. This cursus explained to me the steps on how to build a neural network.



A neural network consists of neurons that each have a simple task. A neuron has one or multiple inputs, a bias, and an output. This translates into the following mathematical function $y = w_0x_0 + w_1x_1 + \cdots + w_nx_n + b$. De w_n is the weight of the connection. When multiple neurons a set of the same input have, they are called a dense layer. During the transformation of the neural network, the weights are adapted until the outcome is acceptable. This can be done with different activation functions like the rectify function. But the activation formula depends on the kind of layers and the way that these layers are structured. But in general, you always have these three layers: input, hidden, and output. All the calculations happen in the hidden layer.

Python has multiple libraries that you can use to build a neural network. You can create a network as I described above with Keras. But there are also other libraries like Scitlearn where the creation of networks happens by calling the function of which you can change multiple parameters.

In the notebook of Sushas Maddali about the prediction of the car prices. He shows how he can transform the data. And how he can put them relatively simple into a model. The models that he used are:

- Linear Regression
- Support Vector Regressor
- Neighbors Regressor
- PLS Regression
- Decision Tree Regressor
- Gradient Boosting Regressor
- MLP Regressor

These models originated from Scitlearn. Then he plots a line through the ratio between the expected and the actual outcome. At the end of the notebook, he compares them by looking at the mean absolute error and the mean squared error. And lets his choice be influenced by these mathematical values. I want to try to apply the same process. So that I can see the difference between the models and choose the best one. (Maddali, 2022)

Bibliografie

Adeel. (2021, 10 18). *Data Transformation Methods : Deep Neural Networks for Tabular Data* . Opgehaald van Towards Data Science : https://towardsdatascience.com/data-transformation-methods-deep-neural-networks-for-tabular-data-8d9ebdeacc16

Arvai, K. (sd). *K-Means clustering in Python : A practical guide* . Opgehaald van Real Python: https://realpython.com/k-means-clustering-python/#:~:text=The%20k%2Dmeans%20clustering%20method,the%20oldest%20and%20most%20approachable

Carrasco, O. C. (2019, 06 03). *Gaussian Mixture Models Explained* . Opgehaald van Towards Data Science : https://towardsdatascience.com/gaussian-mixture-models-explained-6986aaf5a95

Femture . (2022). *Femture*. Opgehaald van Linkedin: https://www.linkedin.com/company/femture/about/

Gosh, P. (2018, 06 18). *Leave one out encoding for categorical feature variables on spark.* Opgehaald van pkghosh: https://pkghosh.wordpress.com/2018/06/18/leave-one-out-encoding-for-categorical-feature-variables-on-spark/

Lawson, K. (2018). *What's Single Page Application Architecture? How Does it Work?* Opgehaald van bloomreach: https://www.bloomreach.com/en/blog/2018/what-is-a-single-page-application

Maddali, S. (2022, 04 08). *Car Prices Prediction.* Opgehaald van Github: https://github.com/suhasmaddali/Car-Prices-Prediction

Pedregosa, F. V. (2011). Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, 2825--2830.

Satorius. (2020, 08 18). *What is Principal Component Analysis (PCA) and How It Is Used?* . Opgehaald van Satorius: https://www.sartorius.com/en/knowledge/science-snippets/what-is-principal-component-analysis-pca-and-how-it-is-used-507186#:~:text=Principal%20component%20analysis%2C%20or%20PCA,more%20easily%20visualized%20and%20analyzed.

Schmidt, L. (2021, 04 10). *Deep learning for tabular data augmentation.* Opgehaald van Data Science Blog von lschmiddey: https://lschmiddey.github.io/fastpages_/2021/04/10/DeepLearning_TabularDataAugmentation.html

Seldon . (2021, 10 16). *Supervised vs Unsupervised learning* . Opgehaald van Sheldon: https://www.seldon.io/supervised-vs-unsupervised-learning-explained#:~:text=The%20main%20difference%20between%20supervised,processes%20unlabelled%20or%20raw%20data.

Sevey, R. (2017, 08 04). *How much data is needed to train a (good) model*. Opgehaald van DataRobot: https://www.datarobot.com/blog/how-much-data-is-needed-to-train-a-good-model/#:~:text=If%20you're%20trying%20to,expect%20to%20have%20trustworthy%20results.

Suganya, R. S. (2012). Fuzzy c-means algorithm-a review. *International Journal of Scientific and Research Publications*, 47-48.