Scoring Dynamics and Project Characteristics of 180DC Branches Worldwide

Data Insights for 180 Degrees Consulting Global

Lukas Bielmeier and Marta Fedeli

Aalborg University

ACKNOWLEDGMENTS

We would like to express our gratitude towards 180 Degrees Consulting for collaborating with us on this project, for which they have kindly provided us with information and support throughout the process.

TABLE OF CONTENTS

Problem Formulation

The analysis and data found within this paper is the outcome of a collaboration project with 180 Degrees Consulting (180DC). 180DC is the world's largest student consultancy for non-profits and social enterprises and provides high-quality consulting services to social impact organizations (180DC, n.d.a). They provided us with data concerning past consulting projects, previous clients' feedback and an internal scoring scheme for their different branches worldwide.

One problem that 180DC faces is that their branch and project data is collected manually in the form of Microsoft Excel sheets. As of now those Excel sheets, although maintained and updated rather regularly, do not allow to draw a lot of meaningful conclusions with regards to the overall project performance nor do they generate particularly valuable insights when being looked at. At most, they serve as a database on what projects have been conducted in the past and what branches were doing or not doing. As such, it is hard for 180DC to evaluate the reasons behind the success of singular projects as well as branches, and they also struggle to extract features that are characteristic for good performance.

As 180DC plans to introduce a Data Science & Analytics - Business unit to their organization in summer of 2021, they need a point of departure for their upcoming data analysts and wanted us to utilize their data for providing initial insights and for finding potential features that characterize good and bad projects which have been previously conducted.

Another subject that we were told is posing a challenge to 180DC is to put the client feedback in context to the internal 180DC feedback. As an international organization, 180DC has branches worldwide whose performance is being rated based on the so-called branch score: this score is calculated based off information entered into Compass, which itself is 180DC's online platform and internal scoring base. Notably, information on Compass is entered manually by the branches themselves. As such, the feedback from the client side does not play into this branch score at all. Instead, the client feedback is collected through polls and surveys and afterwards imported into the other Excel sheets. This could mean that a branch could enter all information on Compass correctly and in time, thus resulting in a great branch score, while the client it engaged with for a specific project is actually unsatisfied with the results the branch has produced for them.

This lack of direct connection between external and internal rating of branch performance may lead to wrongly classifying good or bad performing branches. Especially for a non-profit

organization like 180DC, it is crucial to be able to recognize which of their subsidies are doing well and which are not - at least to be able to effectively allocate resources based on performance.

Hence, as 180DC plans to introduce a Data Science & Analytics - Business unit to their organization and wants to improve their data insights and usability, the goal of our work is to provide valuable insights regarding the overall branch dynamics and project work as well as to isolate characteristics that are shared by the most successful branches and projects. Additionally, we will try to establish a connection between the internal and external scoring to see if the given branch score accurately depicts client satisfaction. Such an analysis could prove to be beneficial not only as a knowledge base for future project decisions, but also as foundation for their upcoming data analytics team. In addition, we hope that our work contributes to a better understanding of their clients and branches. As students of Development & International Relations as well as Innovation & Global Sustainable Development, we are naturally drawn to a work that provides an impact. As such, this collaboration project together with 180DC is not only interesting to us from a data science perspective, but also from a study background perspective. We hope that with the analysis we are conducting and the insights we are generating, we are not only helping 180DC in solving aforementioned problems, but also to complement our study knowledge with real world analysis applications. Moreover, aside from providing general insights in the scope of supporting the new Data Science & Analytics - Business unit, we also aimed to provide 180DC and the GLT team with more final considerations with regards to some of the features shared across branches and the projects they engage in, highlighting top performers and recurring patterns.

Introduction to 180DC

As aforementioned, 180DC is the world's largest student consultancy for non-profits and social enterprises, with operations in over 35 countries and branches based in over 150 leading universities (180DC, n.d.b). The organization is generally built around the Global Leadership Team (GLT), which is responsible for all administrative tasks of 180DC and oversees all global branches. The branches themselves are usually founded by students and usually bound to the university of the city the branch is being established in.

180DC works primarily with organizations that are focused on improving education and health outcomes, achieving environmental sustainability, and/or alleviating poverty. Most of the

non-profits and social enterprises that aim to improve education, health or environmental outcomes, have not yet reached their full potential due to limited resources. 180 Degrees Consulting aims to improve the effectiveness of worthwhile organizations to enable them to achieve the greatest possible social impact through consulting services (180DC, n.d.b). The unique value proposition of 180DC is that they can offer high-quality consulting services without the usually high price-tag such services often entail. As a matter of fact, 180DC consultants are students that work on a volunteer basis: on the one hand, this ensures the cut on the price-tag and on the other hand, non-profits will receive counsel from highly driven and motivated students that believe in the work they do.

To sum up, the mission of 180DC is "to ensure that non-profits and social enterprises committed to education, health, and poverty alleviation can reach their full potential by meeting their demand for strategic and operational assistance, and in doing so developing the next generation of social impact leaders" (180DC, n.d.b).

We initially came to know 180DC's reality from personal experience (Lukas Bielmeier is the co-founder of the 180DC Lund branch in Sweden , currently serving as their Vice President) and after having a discussed with 180DC's Deputy Chief Executive Officer, Celine Hua-Ching, some potential data-related problems that we might be able to help out with, quickly, we were informed that there is a plan to soon introduce a Data Science & Analytics - Business unit to the organization. This was a great point of departure for our analysis and collaboration, as it meant that the insights, we are providing with regards to the overall branch dynamics and characteristics, as well as this project work, form an initial knowledge base for their upcoming data scientists. For these reasons, this collaboration and the project that followed, represented for us a perfect match, both in terms of study relation as well as personal interest. We will now proceed with introducing the rest of our work with a brief overview of the steps we will take, as well as the strategy and techniques we will employ to tackle our analysis.

Project Outline

As touched upon before, the goal of our analysis is to provide valuable insights regarding the overall branch dynamics and project work, and to isolate characteristics that are shared by the most successful branches and projects. Moreover, we want to establish a connection between the internal

and external scoring to see if the given branch score accurately depicts client satisfaction. We approach this challenge through the application of multiple data analysis methods as explained below.

After cleaning and processing our data, we start with some *Exploratory Data Analysis (EDA)* to get a better feel for the data at hand in terms of both branches' and projects' characteristics. EDA is simply the concept of visually laying out the data at hand.

Through *Network Analysis*, we will continue with creating a network of projects to further investigate the features that are shared by the different projects. Here we will also look more specifically at the characteristics of the top performing branches. Our idea here is to showcase the different characteristics that define successful projects and to see how they are shared among said projects.

In the next step, we will apply *Natural Language Processing* to pre-process the text data in our dataset as well as to turn it into text representations (numerical vectors): this will allow us to analyse it later on both by looking into recurring terms and shared macro-topics, thus exploring potential latent patterns within the feedback received. By doing so, we hope to discover certain themes within the written client feedback. For this reason, we will also apply standard *Unsupervised Machine Learning* (UML) steps and draw plots to give a more visual outlook on said hidden patterns. In general, Unsupervised Learning is a Machine Learning method where the model works on its own to discover information based on latent patterns.

After doing so, we will move on to complement our analysis with *Supervised Machine Learning* (SML), first with models based on text data from the reviews, then with models based on the numerical data from the same reviews, ultimately comparing the results between the two data types. In contrast to UML, in SML the “machine” is trained using data which is already “labelled”, which means that some data is already tagged with the correct answer (e.g. in our case the branch score of that specific branch). Here our goal is to accurately classify the branch score of the different subsidies of 180DC based on the project information at hand to address concerns about potential inconsistency between internal and external branch perception and ensuring that outside feedbacks are contextualized within the internal 180DC rating.

Finally, we will conclude this analysis by tackling the same classification task, this time with a *Deep Learning* model. Said model was built as a stacked neural network with multiple input channels - one for categorical data and one for text data. The idea behind Deep Learning is to discover intricate structures in datasets by backpropagating and changing internal model

parameters in this process (LeCun et al., 2015). We will also compare the results of a stacked multi input channel model to one where there is only one input channel. Again, we hope to put the external client feedback in context to internal branch feedback.

Ultimately, the analysis's results are summarized and presented in a brief conclusion.

Choice of data and data preprocessing

For our analysis, we were given private access to some of the MS Excel sheets 180DC' Global Leadership Team makes use of to monitor the branches performance and to keep track of the projects that are being carried out.

For this particular project we chose to use one containing information on projects' reviews from 180DC' branches' clients and one with parameters and information in regard to the branches' Compass scores. As it was previously introduced and further explained below, since we want to tackle the GLT concern about rating harmonization, the two datasets needed to be imported and the more important features of the two, needed to be merged together to proceed forward with the analysis.

Moreover, as it often happens, "real world" data in its raw form can often be imperfect, incomplete or hard to manage. As above mentioned, 180DC manually collects its data and then stores it in the form of Microsoft Excel sheets: especially with regards to Clients Feedbacks and Reviews, this lead to a very sparse dataset, with many missing values, and that in general needed some extra pre-processing steps before being ready for analysis.

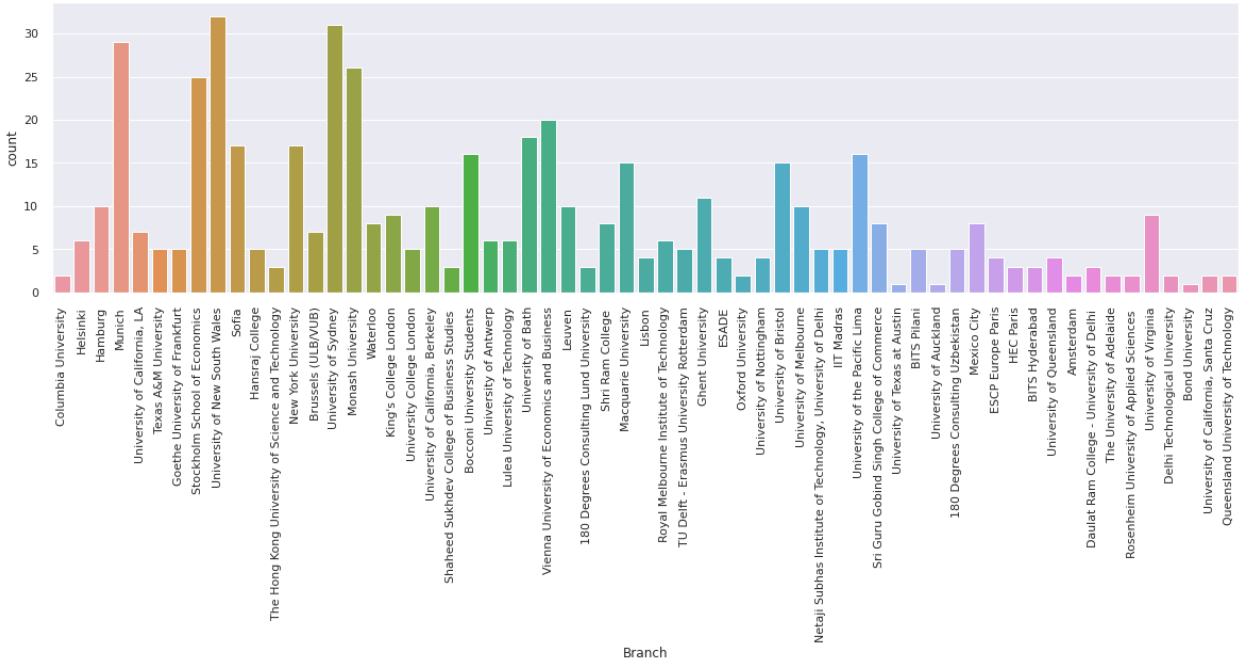
Some basic steps we took were dropping missing values from the dataset, thus choosing quality over quantity of data. Additional preprocessing steps regarded putting the data within the dataframe in the right format (converting yes-no answers into a binary 1-0 classification, or setting the date to datetime format).

Visual Exploratory Data Analysis

As first thing after data preprocessing, we carried out a brief Exploratory Data Analysis focusing on gaining more visual and intuitive insights about the dataset at hand, while also providing the reader with a more general understanding of the work carried out by 180DC. The first two plots will focus more on branch characteristics; the latter, more on the characteristics shared by the

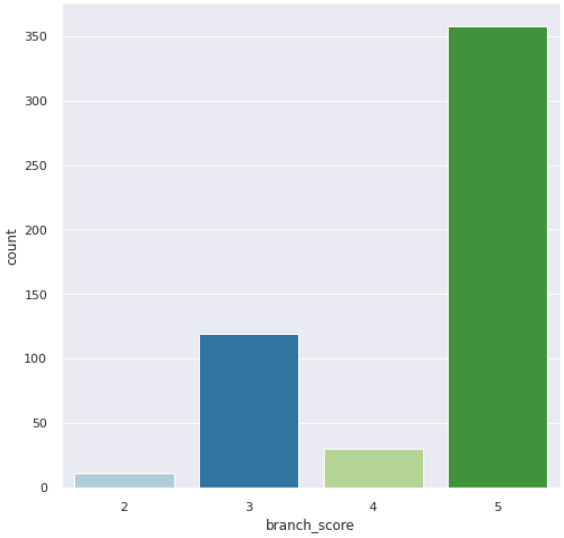
projects specifically (for the latter graphs' interactive rendering, please refer to the provided notebook).

We therefore begin with plotting out a value count for the number of projects each branch has conducted and that has been recorded within the MS Excel reports aforementioned. This is to give first, an impression of how equally distributed the projects are within the whole dataset of branches, which is valuable for the subsequent data processing, and secondly, it visually lays out branch performance in terms of project amount. As shown below, the best five performing branches in this regard are three branches from the Asia-Pacific group, the University of New South Wales (32 projects reported), the University of Sydney (31) and Monash University (26), and two from the EMEA group, the Munich branch (29) and the Stockholm School of Economics (25). This plot, however, generally highlights the presence of quite a significant variance in the number of projects each branch has carried out and reported.



Project-count per branch

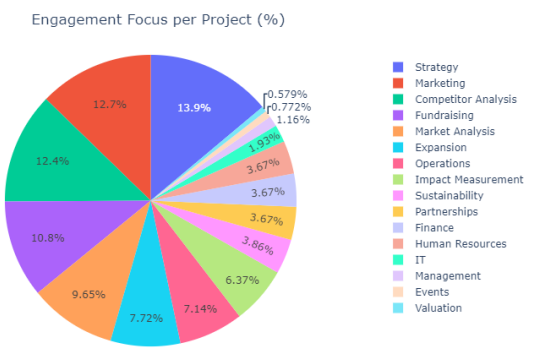
Secondly, since later on in the analysis we will work with multi-label classification models, we shine a light on the level of label-balance in the dataset: in our case, said label is the score Compass score assigned to each branch, which ranges from 1 to 5.



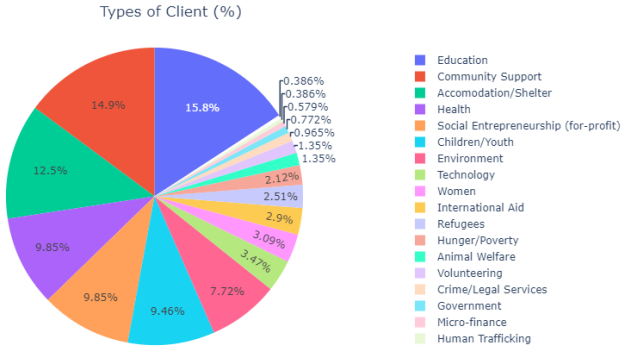
Multi-class "branch_score" distribution

As shown by the plot above, our dataset is extremely unbalanced. Nevertheless, like aforementioned, we made the conscious decision not to further drop any of the "classes" of ranking to avoid compromising an accurate depiction of the scoring across branches and the subsequent analysis. This implies that we must be aware of potential issues of false accuracy later on, and that we would have to carefully look into other metrics (such as recall and precision) when drawing conclusions about models' performance.

The next two visualizations are meant to showcase two of the most important project features. One is the focus of engagement of the project, which indicates what client problem the project was concerned with. The other is concerning the types of clients that the projects of 180DC were working with.

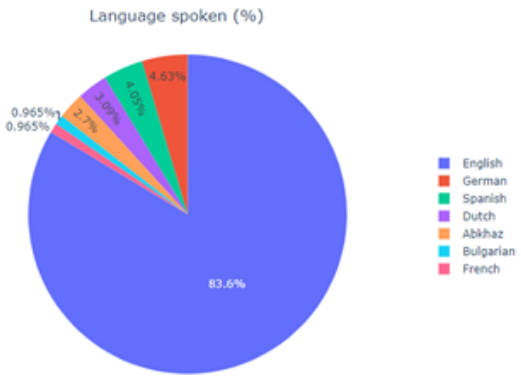


Engagement Focus per project – pie chart



Client type – pie chart

Four types of engagement focus are particularly popular within 180DC branches' projects, taking up half of the share for said projects' engagement type: they are respectively Strategy with 14% share (72 projects), Marketing at 13% (66 projects), Competitor Analysis at 12% (64 projects) and Fundraising at 11% (56 projects). As far as client types 180DC branches engage with, there are three that make up for more than a fourth of the share and are respectively clients from the field of Education (82 projects, 16%), those from Community Support (77 projects, 15%) and those dealing with the field of Accommodation/Shelter (65 projects, 13%). These, together with clients from Health, Social Entrepreneurship, Children/Youth and Environment which immediately follow, make up for almost 75% of the type of client 180DC works with. These numbers confirm that 180DC is a very diversified organization, supporting clients with various types of expertise. For our subsequent analysis, it will also be interesting to check if a particular project focus or a certain type of client is associated with a higher branch score.



Project language – pie chart

As later in the analysis we will focus on NLP, we also plot a distribution of the languages client's feedback are reported in. With this clear visual in mind, we will be able to interpret as well as clean the text accordingly, for example in regard to specific languages' stopwords.

Network Analysis

In general, networks are graphical representations of the relationships (edges) between variables (nodes). Network analyses allow to estimate complex patterns of relationships and the network structure can be analysed to reveal core characteristics of the network (Hevey, 2018). Within our Network Analysis section, we want to create a network of projects and their characteristics.

The goal for our project network is to showcase the different characteristics that are central to projects, with a special focus on the top performing projects. Aside from the projects, we also included multiple project features. The first feature we added is the branch, as this allows us to spot how central certain branches are to the projects.

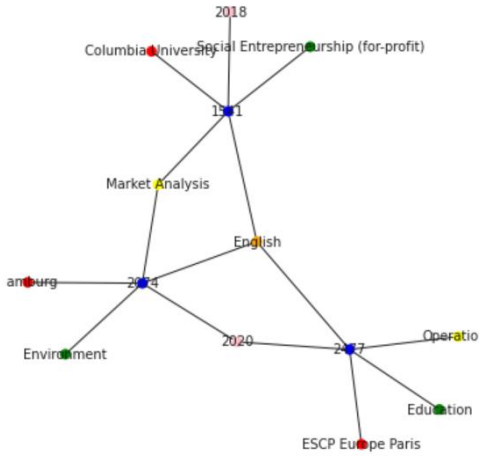
Next, we included "language" to our network as an indicator of how diverse the projects under this feature. Moreover, two aspects we will particularly investigate are the engagement focus of the projects as well as the type of client the project was concerned with. Our hope is that these features will provide some insights as to whether a certain type of client, or a certain engagement focus, are correlated with a higher branch performance. Finally, we also added the year of the project to see how the projects have developed throughout time.

We then tackled building our network in terms of nodes. Having only projects as nodes and their characteristics as node attributes would result in a network of standalone nodes that share some characteristics without showcasing much information. As aim to capture the importance of these characteristics to the overall projects, we decided against adding them as node attributes and to add them as actual nodes. Although this brings along some challenges, we consider this approach useful as it will allow 180DC to get a better feel for how central certain project features are to the overall project network. As such, our network will not only contain the *project_id* as nodes, but also nodes that represent a certain *branch*, *type of client*, *year*, *engagement focus* of the project or the *language* that the project was conducted in. Hence, we will build a network of nodes that are not all the same kind. In general, the following structure applies to our network:

- Blue nodes = project id

- Red nodes = branch name
- Green nodes = type of client
- Yellow nodes = engagement focus
- Orange nodes = language
- Pink nodes = year

An example of this network for three projects of 180DC is provided below.

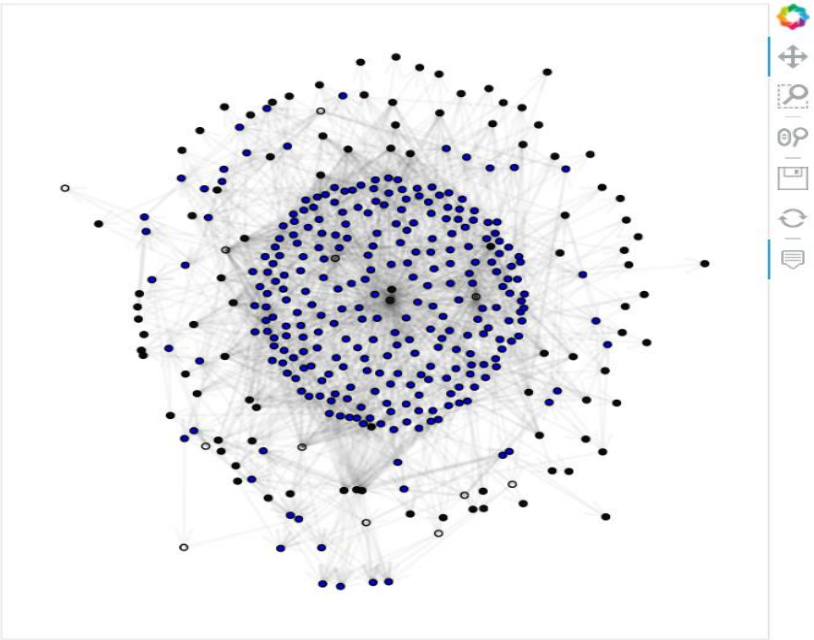


Exemplatory project network using three projects

From the above results we can already observe some characteristics that are shared between projects. It seems like all three projects were conducted in English, while two of the projects also shared market analysis as the focus of engagement. Two other projects, for example, share the year the project happened in. The other features are unique to each of the three projects.

As mentioned earlier, the main focus of our network analysis is to isolate features that are characteristic for successful projects. As such, we have to find a measure by which we define a successful project and map our project data for this metric. We chose as measure for success the *net promoter score* that a project was given. This score is given by the client the project was conducted with and is the main indicator for project success. In general, this net promoter score can take on a value from 0 - 10, with 10 being the best possible score. Hence, we filtered the data for all projects that were rated with a net promoter score of 10 and created a network out of those top projects.

We visualized this network through *holoview* and *bokeh*, which are two visualization packages that can be used for network plotting. The network for the top projects and their characteristics can be observed below. Please note that while *bokeh* and *holoview* are made for interactive visualizations, the picture below is not. As such, havering over nodes will not result in node information being presented. For the full interactive network visualization please refer to the provided HTML.



Interactive visualization of top project network

Looking at the above network, some interesting insights can be drawn. As expected, the most central feature is again the English language, which is due to the reason that more than 80% of projects have been conducted in English. However, we can also observe that next to English as a language, *Strategy* has evolved to a very central part of the top project network (it is the second most central black dot). It seems like this was a focus of engagement that the clients particularly valued. In terms of client type, it doesn't seem like there are specific client types that can be connected to more successful projects.

Next, it can also be observed is that the years 2019 and 2020 have moved more in the center of the visualization. In contrast, the other years (2018, 2017, 2016 and 2015) can be found more in the periphery. It should be noted here that we have slightly more observations from 2020 than from other years, however, not significantly more. As such, this network behaviour may suggest

that the project quality of 180DC has increased throughout the years, as more projects in recent years have received top rating. Moreover, certain related nodes are really close to each other - for example, we can see that Children and Youth (both client types, found towards the mid-left of the visualization) are very close together. Accommodation and Shelter too, both a type of client, can be seen closely together towards the bottom of our visualization, indicating that these client types appear to be closely related to each other.

Natural Language Processing

Natural Language Processing (NLP) is a branch of artificial intelligence that deals with the processing and analyzing of human language in a way that makes it possible for computers too to understand and interpret it.

If previously, through Network Analysis, we inquired about the characteristics 180DC's branches share across projects, through NLP we take a closer look at the text of the reviews and the feedback such projects have received. Therefore, on the one hand, we employ NLP on the corpus of text data at hand, to discover shared features, latent patterns and macro-topics that can characterize a branch, its team and its projects. On the other hand, NLP will also be implemented as a preprocessing step to create a Supervised Machine Learning Model later in the project.

First, after having processed and cleaned the full text of the reviews, we begin our analysis by inspecting the corpus of the collected words through simple word counts techniques.
Since often a graphic object can convey a message with the highest immediacy, the first tool chosen for exploring the tweets text was the word cloud displayed below.

180DC Clients written feedback – word cloud

The world cloud promptly gives a general understanding of the work performed by 180DC. Like above mentioned, 180DC's branches do in fact rally their members to work together as a team on different projects, providing their counsel and recommendations. Moreover, as displayed in the functional notebook, a more precise word count also highlighted a feeling of general satisfaction shared across project's reviews: terms like "professional", "good" and "great" were in fact among the most used, right after those above mentioned describing the general work of 180DC (and that are shown in the word cloud). This already hints to an alignment between outside reviews and internal ranking: as we previously saw through Visual EDA, most of branches were top performers (branch_score = 5) and the fact that a positive sentiment shared across reviews is so prominent, is a sign of an outside-inside ranking alignment.

We will further inquire and draw further insights on this alignment in the following Supervised Machine Learning and Deep Learning sections.

Topic modeling (LSI and LDA)

Moving forward to a deeper level of NLP analysis, the dataset was scouted to find potential macro-topics that clients and consultants were most commonly reporting on. To do so, we first had to process our data one step further, by creating text representations. Machines are not able to understand human language and text like humans do; hence they need to be fed language in a sort of numerical representation: the Bag-of-Words (BoW) and Tf-IDf models are techniques that convert text sentences into numeric vectors.

Since Tf-Idf creates vectors in a way that better highlights a term's relevancy over a mere count occurrence, we use the text converted into this specific format to build a Latent Semantic Indexing (LSI) model first and a Latent Dirichlet Allocation (LDA) model later for topic modeling purposes.

Topic modelling is built around the idea that the semantics of a text are determined by "hidden, or *latent* variables" and models like LSI and LDA aim at uncovering these specific variables (the topics) that shape the texts' meaning (Xu, 2018).

Out of the two, LSI achieved a better performance in separating five macro topics more coherently. As always, when dealing with text and topic modeling and especially, when dealing

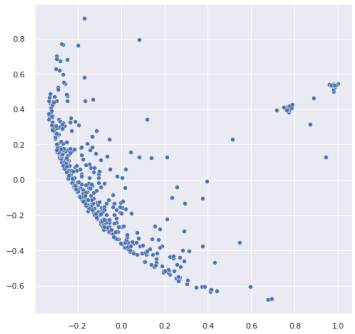
with "real world" language data which might somewhat lack in quality, reading into not very clearly defined macro-topics can be something that requires a bit of an interpretation stretch.

As shown in the functional notebook provided, out of the five topics, two mostly collected random terms like preposition and articles in German and Spanish. Although considerably stopwords, these terms not listed in the standard NLTK packages and slipped through the cleansing process but we decided not to go back and manually remove said stopwords in order to show that the model is indeed working, as well as its logic in defying macro-topics based on terms (so in this case, defying an “Unlisted Stopwords” topic).

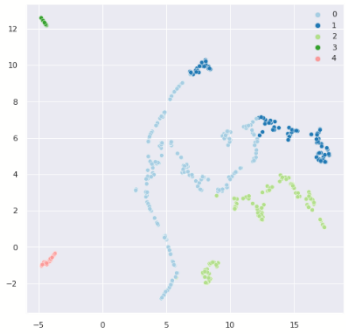
Topic 0, 3 and 4 too, although appearing not to be very clearly defined, show some sort of logic emerging; topic 0 seems in fact to collect words referring to the input and the work that went into each project by the specific branch as shown by terms such “work”, “time”, “good” and “project” for example. Topic 3 on the other hand, seems to group more terms on the description of the input itself (“analysis”, “solutions”, “recommendations”). Lastly, topic 4 appears to say more about the skills put forward by the student-consultants within the branch (“skill”, “expertise”, “fresh”, “insightful”).

Unsupervised Machine Learning plots

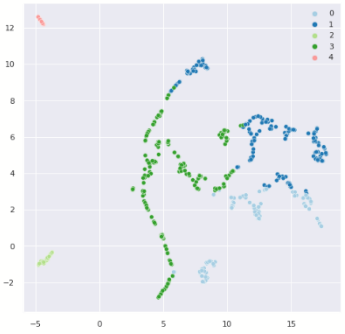
Seeking to further explore the latent patterns that group these topics, we rendered them in a more immediate visual way by incorporating standard Unsupervised Machine Learning steps and plots in our analysis. To proceed, we first tested two different algorithms for dimensionality reduction (PCA and UMAP) to “compress” the datapoints across the topics to fit into the group where they share the most significant relations; secondly, we test two clustering algorithms (KMeans and Hierarchical Clustering) seeking for the one that would provide a more accurate division of the data points across clusters.



PCA dimensionaliy reduction



UMAP and Kmeans



UMAP and Hierarchical Clustering

Summing up from the graphs above, the best combination for highlighting the latent patterns across topics was using UMAP together with Hierarchical Clustering: as shown above, this combination reached a slightly better performance, not only by highlight the clear presence of two topic clusters on the left side of the graph (Topic 4 and 5) but also in dividing diving the remaining clusters in less "noisy" and separate group clusters, especially for topic 2 and 0.

UML thus confirmed the presence of latent patterns that indeed bind together the terms of the reviews into some macro-topics: as data grows in time, so might the ability of UML to separate macro-topics more and more clearly.

Word2Vec

Besides the LSI and LDA models, we also implemented word2vec, a model that learns word embeddings (vector representations of text) to provide 180DC GLT with a practical tool to uncover features and similarities shared across branches (always based off of clients' and consultants' reviews).

We simply built the model in a way that could resemble a browser search bar: feeding the model with the chosen term one is interested in gaining further insights on, word2vec will return a series of terms sharing semantic and syntactic similarity with said term. Say one tries, like showed in the functional notebook, to learn more about what were the recurrent skills and actions that lead to a positive performance withing projects and therefore feeds the term "positive" to the model, terms such as "professional", "helpful" and "ideas" were returned, painting a general picture of what 180DC branches was able to achieve with great results, not only in terms of work outputs but also in terms of attitude .

Once again, in time, as the data grows in quantity and quality, it will be easier for word2vec to gather more significant terms and thus for those interrogating it, more significant insights.

Supervised Machine Learning (SML)

As shortly touched upon before, a supervised learning algorithm learns from labeled training data and helps predicting outcomes of new data. In our case, we will try to predict the branch score of the different branches based on feedback the clients provided to the projects they have worked on.

SML with text data

As above mentioned, on the one hand NLP has served us in gaining further insights into the 180DC client reviews and feedback, on the other hand, through NLP, we took some data cleaning and preprocessing steps that were necessary to now tackle the classification of a branch's score based on the reviews said branch has received on its projects.

Once again, our purpose in this is to confirm consistency between internal rating and external rating, ensuring that outside feedbacks are contextualized within the internal 180DC rating.

We do so by first applying two baseline-Machine Learning models, a Logistic Regression and a Random Forest Classifier, both go-to algorithms for classification tasks and imported from the *sklearn* models' library. In particular, a Random Forest Classifier creates decision trees on randomly selected data samples and gets predictions from each tree and selects the best solution by means of voting. In the following Deep Learning section, we will instead make use of neural networks non-baseline models to tackle the same classification.

The classifiers performance is measured through four main metrics: accuracy, which represents the proportion of predictions that the model classified correctly, precision which answers the question of how many, out of the instances predicted as x-label were actually x-labeled, recall, which shows how accurately a model is able to identify the relevant data and lastly, F1 Score, which measures both the preciseness and robustness of a model (Shung, 2018).

When comparing the scores for the aforementioned metrics of the two classifiers as reported on the provided notebook, the Random Forest Classifier has generally performed better with better scores throughout. The RF classifier reached its top performance in terms of accuracy

(almost 80%) and most importantly, although still far from being a top tier performance, beat the Logistic Regression model by 8% in terms of F1 Score metrics (70% vs 62%), making it the most sound model out of the two.

Before moving onto the SML modeling applied to categorical data, we will briefly look into the topic of Model Explainability for a better understanding of the reason why our models made a specific classification over another.

Model explainability (ELI 5)

For the way they operate, ML models are in fact often referred to as "black boxes": while we can see what input is fed into and what output derived from a model, we appear to be left in the dark in terms of the process that governs the connection between input and output (Medium, 2019). Explainability models allow us to inspect said processes so that we can better explain the logic behind why a classifier returns a specific output over another meaning that fitting an explainability model such as ELI5 to our text classification SML models, will provide further insights on what were the components within a specific review that led to a specific classification.

To proceed, we randomly selected one review out of the pool of clients' feedback in the defined test set, to use as an example. As classes "branch_score = 5" and "branch_score = 3" were the most significant on our dataset, we chose to focus on these two classes only for looking into which specific elements influenced the predictions.

As shown by the tables below, when present in a review, terms with a positive connotation such as "lot", "accomplish" or even "extremely", were among the top elements that influenced said review to be labeled as "branch_score = 5" and oppositely, among the top ones dragging it away from being labeled with the lower branch score for the Logistic Regression classifier.

y=3 top features		y=5 top features	
Weight[?]	Feature	Weight[?]	Feature
+0.179	good ideas	+0.395	<BIAS>
+0.167	founthation branding	+0.150	lot
+0.137	solutions implement	+0.136	extremely
+0.123	think beneficial	+0.124	provided
+0.111	lot team	+0.118	branding
+0.106	status extremely	+0.108	accomplish
+0.104	aware ones	+0.108	aware
… 29 more positive …		+0.106	pair
… 21 more negative …		+0.090	think
-0.113	overall	+0.083	financial
-0.113	branding	+0.081	ideas
-0.114	extremely	+0.079	pleased
-0.118	financial	… 16 more positive …	
-0.118	pleased	… 20 more negative …	
-0.127	accomplish	-0.072	team able
-0.129	ones	-0.097	good ideas
-0.131	ideas	-0.103	pair fresh
-0.139	think	-0.110	founthation health
-0.159	aware	-0.117	lot team
-0.187	lot	-0.128	solutions implement
-0.228	provided	-0.135	solutions
-0.555	<BIAS>	-0.141	aware ones

ELI5 applied to Logistic Regression

For the Random Forest Classifier, although the terms that influenced a classification over the other were less interpretable, words such "provided" and "pleased" (of positive connotation) weighted more to label the review with an higher class and the word combination “yet negative” acted as a negative weight, so that a review containing said combination would not be classified as with the higher label.

y=3 top features		y=5 top features	
Weight[?]	Feature	Weight[?]	Feature
+0.227	forward	+0.738	<BIAS>
+0.148	many good	+0.187	think
+0.146	solutions	+0.170	provided
+0.133	work team	+0.163	pleased
+0.128	good ideas	+0.154	financial
+0.126	fresh eyes	+0.129	accomplish
+0.121	work talented	+0.125	extremely
+0.102	implement	+0.125	branding
+0.096	moving forward	+0.111	pair
… 25 more positive …		+0.104	professionals
… 19 more negative …		… 23 more positive …	
-0.101	extremely	… 21 more negative …	
-0.101	health	-0.107	many ideas
-0.104	talented	-0.108	issues already
-0.113	pleased	-0.116	work talented
-0.126	pair	-0.122	moving forward
-0.138	eyes	-0.125	fresh eyes
-0.138	branding	-0.137	work team
-0.143	provided	-0.142	many good
-0.162	financial	-0.143	implement
-0.182	think	-0.248	forward
-0.846	<BIAS>	-0.389	solutions

ELI5 applied to Random Forest Classifier

Although conscious of the fact that we are dealing with a dataset that is nor particularly broad nor varied and that therefore, could not be the most suitable for this type of models and classifications, some of the examples highlighted above through ELI5, prove that there are still some modest insights that can be drawn from the above representations. As the data grows in quality and quantity, chances are that text classification will happen more accurately, making it in turn easier to clearly identify which elements influenced a specific classification over another.

SML with numerical data

The focus of the following numerical supervised machine learning (SML) section is to bridge the gap between external and internal perception of a branch's performance only based on the numerical client feedback that was given with regards to the branch projects instead (e.g. multiple binary variables that indicate whether or not the client would repeat the project or if it was publicly disclosed for example).

Similar to the case with text data, the branch score values will be our *y* and the remaining values of the dataframe represent our *x*. After defining our *x* and *y*, we again create a train and test set using the *train_test_split* function and a test size of 0.15. This means that we will use 85 percent of our data for training the model and 15 percent for validating its performance. Like before, when trying to predict the branch score, which can take on a value between 1 and 5, we are dealing with a multiclass classification problem.

Like before, we tackle this classification by applying two different SML models, whose performance is then evaluated. The idea behind applying two models is to first find out which one performs better and then apply hyperparameter tuning to the better performing model, further improving model accuracy. Hyperparameters are parameters within a model that can be specified and adjusted to influence model performance - one example for such a hyperparameter is for example the number of estimators that are being used.

The first model we are applying to our data is once again the Logistic Regression model which achieves an overall accuracy of around 80% in correctly classifying the branch score based on numerical client feedback. The classification metrics of the logistic regression model can be observed below.

```
[[ 0  2  0  0]
 [ 0 13  0  5]
 [ 0  1  0  3]
 [ 0  3  1 50]]
              precision    recall  f1-score   support

           2       0.00      0.00      0.00         2
           3       0.68      0.72      0.70        18
           4       0.00      0.00      0.00         4
           5       0.86      0.93      0.89        54

    accuracy                           0.81        78
   macro avg       0.39      0.41      0.40        78
weighted avg       0.75      0.81      0.78        78
```

Classification Report and Confusion Matrix for SML with Logistic Regression

Although these metrics are already not too bad, we want to apply a second model to see if it performs better than the logistic regression. Especially the low recall score in the logistic regression model is something we would like to improve.

The second model we are applying is the Random Forest Classifier. For our initial model we used 200 estimators, which already resulted in an accuracy of around 94.8%. As such, the Random Forest Classifier does a better job in classifying the data than the logistic regression model.

Since the Random Forest Classifier performs better, we now want to tune its hyperparameters to improve our model even further. Namely, the parameters we want to optimize are the number of estimators that were specified as 200 in our initial model. After defining a list of potential parameter values and iterating through them using *KFold* cross validation, a resampling procedure, and *GridSearchCV*, which loops through our predefined hyperparameters.

This hyperparameter tuning then suggested an optimal amount of 10 estimators for our model. Hence, we specified the new amount of estimators for our model and fit it again to our data. The classification metrics for both instances of the *Random Forest Classifier* can be found below.

```
[[ 1  1  0  0]
 [ 0 17  0  1]
 [ 0  1  2  1]
 [ 0  0  0 54]]
              precision    recall  f1-score   support

           2       1.00      0.50      0.67         2
           3       0.89      0.94      0.92        18
           4       1.00      0.50      0.67         4
           5       0.96      1.00      0.98        54

    accuracy                           0.95        78
   macro avg       0.96      0.74      0.81        78
weighted avg       0.95      0.95      0.94        78
```

```
[[ 1  1  0  0]
 [ 0 17  0  1]
 [ 0  1  3  0]
 [ 0  0  0 54]]
              precision    recall  f1-score   support

           2       1.00      0.50      0.67         2
           3       0.89      0.94      0.92        18
           4       1.00      0.75      0.86         4
           5       0.98      1.00      0.99        54

    accuracy                           0.96        78
   macro avg       0.97      0.80      0.86        78
weighted avg       0.96      0.96      0.96        78
```

Classification Report and Confusion Matrix for SML with RandomForestClassifier (before and after hyperparameter tuning)

Comparing the two classification-metrics above, we can see that after hyperparameter tuning our random forest model was slightly improved. We can see that one more observation in class 4 was correctly classified (as seen in the confusion matrix in the top left corner) and in general the recall and f1-score for this class appear to have improved. Ultimately, we can conclude that our tuned random forest model was fairly successful in correctly classifying the branch score based on the categorical client feedback. This indicates that the perception of a branch's performance is similar for clients and 180DC internally.

Deep Learning

Now that we have tried supervised machine learning for both text and numerical data, we are focusing on Deep Learning. However, this time we are combining text and numerical data for the branch score classification.

In general, Deep Learning is a subset of the broader family of machine learning methods based on artificial neural networks. Deep learning enables computational models that contain multiple processing layers to learn representations of data through multiple levels of abstraction. Moreover, through deep learning intricate structures in data sets can be discovered and then used to backpropagate and indicate certain internal parameters that the model should change (LeCun et al., 2015).

Within the next deep learning section, we built a stacked neural network model that has multiple input channels - one for categorical data and one for text data. Although, we are aware that with the current number of observations in the dataset a deep learning model might not perform

to the best level, we do think that as 180DC increases its amount of projects and branches such a model will become increasingly accurate.

Deep Learning pre-processing

Before we can apply a deep learning model to our data, we first need to preprocess the data accordingly.

We start by creating a copy of the *feedback* dataframe that we use for preprocessing. Within this dataframe, we then change the datatype of all numerical variables to be categorical, as this will be one of the input types for the deep learning model. Fortunately, all numerical variables were categorical in their nature already, even though some were labelled as *integer* or *float* values.

We then proceed with augmenting our text features, which is a process that can be understood as artificially generating more text data by augmenting the existing data.

Text Augmentation

Like aforementioned, deep learning models perform better when fed with high volumes of data and for the moment, the data at hand is not really classifiable as a big data kind. To overcome this obstacle and try to provide 180DC with a well-performing classification model, or at least, a proof of concept of what that model could do, we relied on text augmentation to increase our data volume.

We did so through the NLPAug package for Python and created a copy of augmented texts for all the entries within our dataframe. NLPAug works by employing BERT, which is a pre-trained NLP model that is able to identify similar terms based on their previous and following context (Raj, 2019). Through BERT, NLPAug finds similar terms for specific words within a given text sequence and substitutes ones with the others: this returns a new text sequence, which is different from the original one and yet shares practically the same meaning (which ensures that the ratings given for the original text entry still align with the returned text sequence).

We used these new text sequences as if they were a mock-up of future, potential reviews: this allowed us to double our number of entries and by adding them to our dataset, making it more fit for a deep learning model.

Neural Network Architecture

To begin with building the actual architecture of our deep learning model because we are having categorical as well as text features that we want to use as inputs to our model, we have to create multiple input channels. This is a necessary step as within the model, different processing layers are applied depending on the type of input data.

The first channel to be defined is for categorical features, whose input is run through four dense layers with different unit specifications each. Such layers feed outputs from the previous layer to all its neurons, where each of them are then each providing one output to the next layer; in neural networks, this is the most basic kind of layer.

For our text input, we need different layers; the layers that our text input iterates through are an embedding layer, then a convolutional layer, one dropout layer, a MaxPooling layer and finally a flattening layer.

The first, the embedding layer, represents words and documents using a dense vector representation. This representation then goes through the convolutional layer which applies a filter to the text input to generate a feature map which summarizes the presence of detected features in the input. As such, convolutional layers are the layers of choice when seeking feature extractions (for example, in the case of text, particular expressions or key-words). Afterwards the dropout layer randomly sets input units to 0 at a certain rate, which helps to prevent overfitting. The MaxPooling layer is then used to downsample the feature map. The flattening layer is ultimately used to flatten the output of the MaxPooling layer to a single long feature vector.

Finally, we merge the two different input channels and run them again through another dense layer and the final output layer, which is also a dense layer. In the final output layer, we specify ‘softmax’ as activation function ((best for multi-label classifications) as and add 6 units, as this is the range of values our potential target variable falls in. The final deep learning model architecture is shown below.

Layer (type)	Output Shape	Param #	Connected to
input_2 (InputLayer)	[(None, 500)]	0	
input_1 (InputLayer)	[(None, 16)]	0	
embedding (Embedding)	(None, 500, 36)	180000	input_2[0][0]
dense (Dense)	(None, 256)	4352	input_1[0][0]
conv1d (Conv1D)	(None, 493, 32)	9248	embedding[0][0]
dense_1 (Dense)	(None, 128)	32896	dense[0][0]
dropout (Dropout)	(None, 493, 32)	0	conv1d[0][0]
dense_2 (Dense)	(None, 64)	8256	dense_1[0][0]
max_pooling1d (MaxPooling1D)	(None, 246, 32)	0	dropout[0][0]
dense_3 (Dense)	(None, 32)	2080	dense_2[0][0]
flatten (Flatten)	(None, 7872)	0	max_pooling1d[0][0]
concatenate (Concatenate)	(None, 7904)	0	dense_3[0][0] flatten[0][0]
dense_4 (Dense)	(None, 20)	158100	concatenate[0][0]
dense_5 (Dense)	(None, 6)	126	dense_4[0][0]

Total params: 395,058
Trainable params: 395,058
Non-trainable params: 0

Neural network architecture of multi-channel input model

Once this model is compiled, it was fitted to our data and the model thus evaluated. The output below shows that our model achieves an overall accuracy of around 76%, which is not particularly good but neither a terrible performance. Considering the low amount of data that was fed into the model, we consider 76% fairly good. We can see that the validation accuracy is steadily increasing with each epoch, while the validation loss is constantly decreasing. This behavior suggests that our model is not overfitting the data.

```
Epoch 1/3
10/10 [==============================] - 2s 92ms/step - loss: 1.2824 - accuracy: 0.4992 - val_loss: 0.7766 - val_accuracy: 0.7404
Epoch 2/3
10/10 [==============================] - 1s 63ms/step - loss: 0.9749 - accuracy: 0.5517 - val_loss: 0.7401 - val_accuracy: 0.7548
Epoch 3/3
10/10 [==============================] - 1s 64ms/step - loss: 0.7350 - accuracy: 0.6803 - val_loss: 0.6487 - val_accuracy: 0.7596
```

Multichannel input model loss and accuracy through epochs

However, the above results are to be taken with a grain of salt. As mentioned earlier, the data at hand is limited and with only a thousand observations, deep learning in general is a rather hard task. Nevertheless, we do believe that the model has the potential to classify more accurately once there are more observations that can be fed into the model.

Comparison with non-multichannel type of network

After employing a multiple input channel Deep Learning model, a non-multichannel type of neural network will be built, to inspect how performs in comparison to the previous one. For this model we will therefore merge our categorical and text features together, and give only one input to the model.

Since this model is just conceived as a comparison basis to our multiple input channel, we will build a rather simple network architecture and use the same specifications for our model compilation as with the stacked model before. Again, the model architecture and performance can be observed below.

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
dense_6 (Dense)              (None, 256)               132352
_________________________________________________________________
dense_7 (Dense)              (None, 128)               32896
_________________________________________________________________
dense_8 (Dense)              (None, 64)                8256
_________________________________________________________________
dense_9 (Dense)              (None, 32)                2080
_________________________________________________________________
dense_10 (Dense)             (None, 6)                 198
=================================================================
Total params: 175,782
Trainable params: 175,782
Non-trainable params: 0
```

Neural network architecture of one channel input model

Although the two Deep Learning models differ in complexity, the results still enabled us to see differences between multiple channels and a single one.

```
Epoch 1/3
12/12 [==============================] - 1s 26ms/step - loss: 1.2796 - accuracy: 0.5599 - val_loss: 0.9753 - val_accuracy: 0.7115
Epoch 2/3
12/12 [==============================] - 0s 9ms/step - loss: 0.7169 - accuracy: 0.7160 - val_loss: 0.8132 - val_accuracy: 0.7244
Epoch 3/3
12/12 [==============================] - 0s 9ms/step - loss: 0.6454 - accuracy: 0.7232 - val_loss: 0.8501 - val_accuracy: 0.6859
```

One channel input model loss and accuracy through epochs

Generally, it seems like the stacked model is performing better than the one input channel model, mainly because the validation accuracy and loss appear more consistent in the multiple input channel model.

Discussion

This very last section of our stakeholder report is meant to discuss the methodologies we have chosen, what we could have done differently and what could be interesting potential developments for this analysis in the future.

In general, we executed all our analysis to our best knowledge and applied the methods we deemed most promising for the tasks we wanted to solve.

With regards to the network analysis, we could have also created a network of branches instead of projects but believed that a network of branches would provide less information, mainly because we did not have any data on interactions between branches, which would result in very little node attributes per branch. For the sake of this project and with the dataset at hand, designing a project network, however, allowed to draw much more detailed conclusions with regards to what defined a project's success or failure. We do advise the future 180DC Data Science & Analytics – Business unit to try and also collect data indicating interactions between branches (e.g. what actions have been taken to make the branch successful), as it would allow to create a more efficient overview in terms of Network Analysis of 180DC branches globally.

With regards to Natural Language Processing, what could have been done was to proceed in a more sectional way, meaning we could have looked at the text per column instead of merging it together. Moreover, we could have done so by looking into client types or engagement type focus, investigating for macro topics, trends or common characteristics, first for project reviews dealing with Strategy as focus, then for reviews whose projects engaged with a client in the field of Education or one from Social Entrepreneurship, etc. However, with the current amount of data, results of said type of analysis would have been misleading or very inconclusive at best.

With regards to Supervised Machine Learning, although we thought about an approach that would differentiate between two possible branch performances (either positive or negative), thus creating binary classification model for a more straightforward output, we deemed it ill-suited, because to favor more immediate and clear outcomes, we would have lost track of the different nuances that currently characterize performance across branches.

Another potential challenge for the Data Science & Analytics - Business unit to look into could be to create a new type of scoring, based on text review's sentiments, which will then be complemented with the other metrics in generating a final score that truly accounts for consistency

between internal and external perceptions. For the sake of this project, we decided not to do so, as such a new score would imply a completely reworked way branches are being rated within 180DC.

Moreover, another potentially interesting SML model might try to predict a project's final outcome based on feedback that was given up to a certain point. However, that would require 180DC to collect client feedback while the project is still running and not already finished. Although this implies an extra effort, we believe it could help in keeping the clients engaged and bears the potential for the project team to readjust their efforts based on this in-between feedback.

For the Deep Learning model, we decided to use a multi input channel model. While we do think this model is well suited for classification tasks based on client feedback, it might also be interesting to build a Deep Learning model that is able to predict other branch or project related metrics, such as partnership growth or client retention for example (assuming the necessary data is readily available).

Overall, we found the data at hand to be a bit scarce, especially after deleting all missing values which was necessary to proceed with a sound analysis. This made it increasingly hard for a lot of techniques within the data science spectrum to perform to their best.

As 180DC is planning to introduce a said Data Science & Analytics - Business unit to their organization, we would recommend starting to capture branch and project information more consistently and in more detail. Ultimately, this would create a solid "data-foundation" for the upcoming unit to work more efficiently and to produce more valuable results for the organization as a whole.

Again, we do think that especially having data on the actions that branches took as a response to certain circumstances could be extremely valuable - not least as a benchmarking base for future projects and branches. We look forward to seeing how 180DC evolves as a company and wish them all the best for all of their future endeavors.

Conclusion

In the present analysis we conducted for 180DC, we focused on providing insights regarding branch and project characteristics and tried to match the client feedback to the internal branch scoring.

We first explored the network of top-rated projects. Although it was found that most types of engagement focus were not a characteristic linking to successful projects, we found that on

average the best rated projects appear to be focusing on a client's strategy. The most central part within the network was the English language due to >80% of the projects within the data that were conducted in English. Network Analysis furthermore suggested an increase in client satisfaction with the projects over time, as it was found that more recent years were more central to the network of top-rated projects.

Through Natural Language Processing (NLP), we first could observe hints of an alignment between outside reviews and internal ranking, specifically through text explorations (word counts and the word cloud). As a positive sentiment was shared across reviews very prominently and considering that in the dataset at hand, the maximum branch score of five characterizes the vast majority of elements, this was regarded a general sign of an outside-inside ranking alignment.

As far as finding latent patterns across the text, by applying the LSI model first and with the visual help of UML plots, we could observe some blurry macro-topics starting to emerge: in particular, topic 0, 3 and 4 that were dealing with the work and input coming from 180DC consultants, as well as the skills they presented ("skill, expertise, fresh, insightful").

Through Word2Vec, we also provided an accessible tool to search and get a fast overview on specific features and skills as reported in the Clients' feedback.

We continued our analysis by using the NLP-processed text data as well as the categorical data from the clients' reviews as inputs to multiple supervised machine learning (SML) models. The goal of our SML part was to see if we can accurately classify the different branch scores based on client feedback, which in itself solely contributes to the project reviews, but not the branch score. If we assume the branch score to reflect the 180DC Global satisfaction with the respective branch, 180DC Global satisfaction can be correctly classified by just using client feedback. Were that to be the case, it could be concluded that the perception and satisfaction with branches are similar for clients and the Global Leadership Team (GLT) of 180DC.

While our models could have performed better when using text data, we saw that with applying the models to numerical client feedback, especially after tuning their hyperparameters, we were able to reach an accuracy to over 96%. This result suggests that indeed there is a strong overlap in branch satisfaction between the clients and 180DC Global, indicating that branches are perceived similar externally and internally.

Ultimately, for the Deep Learning section of our analysis, although aware that with the amount of observations we had, deep learning will have likely performed poorly, we chose to apply

two deep learning models to our data, one with multiple input channels and one without, trying to provide at least a proof-of-concept of what deep learning models can do, especially as the data quantity increases.

Nevertheless, even with a limited amount of observations, our deep learning model proved able to correctly learn and classify almost 76% of our validation data. While these results do not appear great at first sight, they are not as bad as expected for the reason aforementioned. Again, as the dataset grows, the accuracy and value of such a deep learning model will also increase.

Furthermore, this "multi-channel input" model can be seen as a base model which can be extended based on the goals of the future Data Analytics Business Unit - e.g. as a recommendation tool as to what branches should work on in the future to achieve a better branch score. After all, the outcomes of our deep learning model already suggest the similar perception of the branch performance between clients and the global 180DC brand.

Finally, we hope that the different data science methods and the overall analyses within this paper contribute to a better understanding of clients and branches within 180DC and that it provides a first knowledge base and initial point of departure for their upcoming data science team. We once again want to thank 180DC for allowing us to conduct this project with them and are looking forward to how our insights will be utilized to further grow and develop the organization.

Bibliography

180 Degrees Consulting. (n.d.a). Homepage. Retrieved from https://180dc.org/

180 Degrees Consulting. (n.d.b). Why We Exist. Retrieved from https://180dc.org/about/why-we-exist/

Hevey, D. (2018). Network analysis: a brief overview and tutorial. *Advanced Methods in Health Psychology and Behavioral Medicine*. Retrieved from https://www.tandfonline.com/doi/full/10.1080/21642850.2018.1521283

LeCun, Y., Bengio, Y. and Hinton, G. (2015). Deep Learning. *Nature*. Retrieved from https://www.nature.com/articles/nature14539

Medium. (2019, December 18). Show Me The Black Box. Retrieved from https://medium.com/towards-artificial-intelligence/show-me-the-black-box-3495dd6ff52c

Raj, B. S. (2019, October 12). Understanding BERT: Is it a Game Changer in NLP?. *Towards Data Science*. Retrieved from https://towardsdatascience.com/understanding-bert-is-it-a-game-changer-in-nlp-7cca943cf3ad

Shung, K. P. (2018, March 15). Accuracy, Precision, Recall or F1?. *Towards Data Science*. Retrieved from https://towardsdatascience.com/accuracy-precision-recall-or-f1-331fb37c5cb9

Xu, J. (2018, May 25). Topic Modeling with LSA, PLSA, LDA & lda2Vec. *Medium*. Retrieved from https://medium.com/nanonets/topic-modeling-with-lsa-psla-lda-and-lda2vec-555ff65b0b05