Finding a Replicable Methodology: a Bolivian Case

Cristhian Bruno Ayllon Calderon
University of Lausanne
Lausanne, Switzerland
cristhianbruno.aylloncalderon@unil.ch

Valeria Medinaceli Molina
University of Lausanne
Lausanne, Switzerland
valeria.medinacelimolina@unil.ch

Martin Ruilova Quezada
University of Lausanne
Lausanne, Switzerland
martin.ruilovaquezada@unil.ch

***Abstract*—The Bolivian startup ecosystem is an undiscovered financial opportunity with several enhancements to make; develop an adequate tool to assess startups is part of them. Currently, there is not a complete database of the Bolivian startup market, so this research will employ a database with comparable characteristics. Therefore, this paper intends to provide a functional and replicable methodology for dealing with small databases, imbalanced binary target and higher dimensionality, by solving a machine learning classification problem. Also, it will discuss the most relevant features that can make a startup falls into a successful or failure class.**

***Index Terms*—Bolivia, startup ecosystem, startup assessment, methodology, small databases, imbalanced classes, binary target, higher dimensionality, classification.**

I. Introduction

The Bolivian startup's ecosystem, as in many countries, is characterized by highly motivated entrepreneurs, technology-driven ideas and a steep learning curve to success. In most cases, Bolivian startups have a heuristic approach, belong to the informal sector and are self-financing. A current problem that the environment faces is the lack of information about the evolution of its main components, the startups. The last and only census made was in 2019 [1], which showed that there are 152 startups related to the technology sector, condensed on the most economically dynamic cities. A crucial detail is the absence of any formal investors on its equity.

Besides, there are only a few research papers that were conducted on Bolivian startups. Most of them discuss a too precise topic, for instance, the analysis made on the entrepreneurial ecosystem in La Paz [2], a city in Bolivia, states the principal drivers of entrepreneurs on a specific period of time. Hence, because the Bolivian academia does not deliver usable insights, startup co-founders are forced to rely on foreign literature or, in the simplest, adopt a conventional approach.

The private sector in Bolivia is, mostly, conformed by informal businesses. According to the data of Fundempresa, the commercial license provider, approximately 79% of authorizations of 2019 belong to ”unipersonal” [3], the standard category among informal firms. The drivers of the latter situation are higher taxation and relentless workforce legislation. So, entrepreneurs prefer to stay informal until take-off and reach maturity.

Informality makes startups not subject to financing. That is why angel investor's programs are highly demanded to win; otherwise, startup's equity would be self-financed, which in most cases is not enough to develop and to validate an MVP (Minimum viable product). The lawmaker, aware of the last, began developing a law to promote the sector. Nonetheless, the regulation can be solidified by profoundly understanding of ecosystem's fundamentals.

Although efforts to make the startup environment flourish, the reality is that there are clear opportunities to enhance the Bolivian startup's ecosystem.

II. Research Question and Relevant Literature

Like many other developing countries, Bolivia has a flourishing startup environment with attractive growth potential and an adequate perspective to fulfill the investors' financial returns, as well as, social and environmental returns. Therefore, from a Triple Bottom Line perspective, a potential investment in one of these markets could be a crucial step further on sustainable portfolio construction. However, one of the main shortcomings of these developing markets is the lack of adequate tools and networks to raise the required resources and connect the investors with prospective investees.

Focus on tools; in Bolivia, there is not an adequate way to assess startups. Some financial institutions adapted their way to evaluate microenterprises by implementing a holistic approach to evaluate co-founders' financial health rather than the startup's ability to deliver a solution for the environment. For instance, they take into consideration civil status, current income, credit history, among others. Some concerns regarding the latter are subjectivity, there is no reliable study conducted to know what factors are crucial for the analysis, and misconceptions, a startup is not like a microenterprise.

Hence, to develop a useful mechanism to judge startups, a necessary step is recognizing what makes them succeed or fail. We started with a rigorously search for literature related to the mentioned topic to comprehend critical industry insights. It is important to remark that success in this context is defined as a startup that raised capital and is still in operations, and vice versa; failure states as a startup went to bankruptcy or had closed its operations.

In the study ”Are ”Better” Ideas More Likely to Succeed? An Empirical Analysis of Startup Evaluation” [4], the author discusses that intensive RD startups are significantly more likely to be funded if they have positive feedback of a mentor because they have the necessary skills to classify high-quality ideas and excluding non-serious ideas.

Another clear insight is ”the average investor responds strongly to information about the founding team, but not to information about either firm traction or existing lead investors” [5]. Information, in the present context, is referred to founders’ academic and professional background. In other words, if the founding team attended a top university and worked for a renamed company in the past, raising capital probabilities would increase. The paper also shows that highly experienced investors will focus more on the last information; less experimented investors will focus on a variety of aspects.

Additionally, Lussier Corman discussed in extension that the significant factors for entrepreneurs with 0 to 10 employees are: professional advisors, planning, education, minority, staffing, parents owned a business, record keeping and financial control, capital, industry experience and economic timing [6]. However, as they stated in their study, ”management experience was not a significant variable in this prediction model. Capital is the only tested significant variable in their four models and their 0-10 model”.

Some last ideas are those stated on the ”Success and Risk Factors in the Pre-Startup Phase” paper [7]. It mentioned that the amount of time, part or full, spend by the co-founders on its venture is essential, and the founding team needs to present a solid risk manager skill, depending on the particular industry where the startup is entering.

Determinants of success can be summarized on the following categories:

- Experience
- Education
- Founders’ Skills
- Industry
- Financing

Nevertheless, the mentioned factors where studied and tested in their respective country of analysis or using information from sources that were not related to the Bolivian context. That is the central issue regarding to extrapolate these aspects into our scenario. Besides, due to the precarity of the Bolivian startup ecosystem, we do not have information available to apply a scientific approach and draw our conclusions.

A solution to discover these unknown factors and, at the same time, deliver a usable tool to assess startups is to develop a replicable methodology to be used when a Bolivian dataset is released. That methodology will consist of a Pre-Processing of the data, a feature selection and classification models application. Between the last two, we will know the most relevant aspects to analyze to have a better insight about the success or fail of startups. Then, using these features, a classification model that can predict the potential outcome of a startup performance will be constructed. This way, we can provide the Bolivian market with a more in-depth insight into the early stage of a venture. Therefore, investors will be able to allocate their resources smartly, and entrepreneurs will iterate over these features to enhance as much as circumstances will allow.

III. Scope of the project

Before to begin describing our methodology, it is necessary to remark that in this particular scenario we are using a Kaggle’s dataset, CAX_Startup_Data, due to it is the most similar to a possible Bolivian dataset, which may contain several features but it will be relatively small as the size of the ecosystem. Results are based on that input, but the methodology applied is designed for reduced databases, enabling its replicability.

IV. Methodology

To effectively convey information, the methodology will be subdivided into five subsections where reasons and arguments will be discussed, having the objective of supporting our decisions.

A. Pre-Processing

Every dataset requires a tailor-made work to be completed usable. In the case of CAX_Startup_Data, we needed to realize with what features we will be working, what modifications needed to happen and how it needed to be at the end of the process. The following sequence was developed:

1) Column transformation: A feature, ”Industry of Company”, was completed due to its relevance in the study.
2) Column removing: Some features were removed as they overlap information with others.
3) Erroneous data modification: Grammatically inconsistent data points were corrected.
4) Interval assignation: A interval was given to numerical features.
5) Missing values replacing: Features within ”NaN” values where replaced by its mode.
6) Categorization: Ordinal and nominal features were categorized.
7) Normalization: the database was scaled up between a range between 0 to 1.

Further detail about this section is explained on the ”Dataset” section.

B. Feature Selection

We decided to apply this process because we wanted to prevent overfitting as our dataset encompasses several features, reduce training time, since we will apply various models with a hyperparameter tuning, so the time needs to be optimized. Improve model performance, investors' decisions will rely on model precision. And avoid the curse of dimensionality, we required to reduce data sparsity to make our model statistically significant.

The critical goal in this process is to get the most informative and independent features as the dataset allows us; thus, we adopted a two-filter process.

In the first filter, whose primary goal was to get the most informative features, the attributes were selected based on having a strong relationship with the target variable, "status of the company" (STA). Those that had a higher correlation with the target and performed an above-average score on the selection method were desirable. 32.91% of the dataset's attributes had a weak correlation and the rest had a very weak correlation, so we decided to keep with the leading group. Then, features were scored by metric chi2 because most of the input and the output variables are categorical. Those with a score equal or superior to 10 were selected.

By taking as input features from the last filter and measuring its correlation among them, we were able to extract features with overall multicollinearity lower than the average. The last filter was used because of the variety of models that will be applied, most of them can deal with multicollinearity, tree-based models or logistic regression [8], yet some of them not, XGBoost.

C. Model Application and Description

Before applying models, it was necessary to do some additional in-between steps like splitting the data, perform a grid search and make K-fold cross-validation. The latter were oriented to overcome class imbalance and enhance models' performance.

We splited the dataset into training set, 80%, and test set, 20%, to get a more realistic result, as the model is not training with the same data that is using to classify. Also, a stratified argument was used to make both both samples have the same distribution of success and fail entries, respectively: 65% and 35%.

A grid search allowed us to tune models' hyperparameters, iterating over a list of available parameters arguments and testing each possible combination, carrying out on an optimized sequence. Models were ranked with the Macro Precision metric, which will be further analyzed. It is essential to notice that each model has different hyperparameters to be tuned, but we decided to only work with some to avoid overfitting.

Regarding the last idea, a stratified 10 - Fold cross-validation was done to test the model's metrics reliability and, at the same time, address class imbalance issue with the stratified function. The last methodology was implementing and not an oversampling approach (e.g., SMOTE) due to the higher dimensionality of CAX_Startup_Data poses [9]. Besides, a k = 10 was applied because we wanted to have a balance between bias, created by the imbalance target, and variance, produced by the small dataset. A small K implies small variance, significant bias and less computational time and, vice versa, a large k implies significant variance, small bias and more computational time [10].

The models used will be detailed below, as well as its hyperparameter tuned. We will emphasize its characteristics and suitability with our purposes.

1) K - Nearest Neighbors : is the most basic of machine learning classification algorithms. It calculates the distance between the target to be classified and every other point in the training set. Its application work on a small dataset with lower dimensionality. Also, this algorithm needs all the data be represented numerically and it is sensitive to irrelevant features. The hyperparameters tuned were the number of neighbors, low values of k in general overfit and large values often underfit, and the metric, it can be Manhattan, Euclidean or Minkowski. Both were chosen to improve the model's precision.

2) Naive Bayes: is a probabilistic classifier that makes classifications using the method of the posterior probability. We use the Multinomial Naive Bayes classifier, given that our features are discrete. The tuned parameter was Alpha, which is the smoothing parameter, as we do not want probabilities to be zero.

3) Logistic Regression: is used to model the probability when a target is binary like succeed or fail in our case. The size of the dataset will not be inconvenient for algorithm [11], but we are aware that it could drive to overfitting. That is why Penalty, C and the solver were the hyperparameters to tune. Values among the interval for C, regularization parameter, were meager, making the model more conservative.

4) Decision Tree: works by partitioning the feature dimensions into some non-overlapping regions with related decision rules. Being aware of the model bias to imbalanced datasets, we dedicated the above approaches that were discussed. It is crucial to notice that tree-based models do not generalize data in a proper way, which can cause overfitting. That was our motivation to tune Max Depth by limiting its value. Also, another way to prevent the mentioned is with Max Features, which is how many features the model will consider before each split; lower values generalize the model correctly. We know that because of complex datasets, this model can become unstable, so ensemble approaches were also implemented.

5) Support Vector Machine: is used to solve binary classification problems, as our target STA presents. It tries to find a hyperplane in a dimensional space that separates the classes better. Training SVM from large datasets became an issue due to the time-consuming and memory complexity [12]; a reduced dataset will not encounter such problems. We were iterating over Kernel, which takes low dimensional dataset space and transforms into a higher-dimensional space, and Gamma, a coefficient that will control how much we approach observations. Their primary purpose is making the hyperplane “fit” best, so we also need to include a regularization parameter C, presented before.

6) Random Forest: combines multiple decision trees, randomly selected from a subset of the training set, to create decisions, which will be aggregated to conclude a final choice. That combination of decision trees makes it outperforms. The algorithm also takes the most informative features to deal with structural similarities and have less correlation. This model prevents overfitting as it considers each subset's outcome, yet we still tuned the Max Depth and Max Features, explained before, for the small number of entries used. A hyperparameter did not discuss before was Criterion, which states how information gain will be measured to determine the splitting tree. The Criterion was chosen to enhanced model precision as the model will be iterating over it.

7) Extreme Gradient Boosting: also implemented with decision trees with the approach of training a new classifier using a data set in which the weighting coefficients are adjusted according to the performance of the previously trained classifier, building a succession of learning methods. Adding as a critical characteristic overfitting control with its regularization parameters (lambda, alpha), which gives more reliable performance. In addition to the mentioned, a tuned was made on Eta, learning rate (also used to prevent overfitting), Max Depth and Gamma, already explained.

8) Bagging: implements the methodology of randomly selected from a subset of the training set to a prediction and, then, aggregates them by voting or averaging. In this case, we iterate over the models before explained except the ensembles.

D. Model Evaluation

The models were evaluated based on two criteria:

- Macro Average Precision: Precision due to it is crucial to know how the model correctly detects values. Furthermore, the Macro average was applied because of the model must be evaluated by its precision to recognize successful or failed startups equally, being no sensitive to class imbalance.
- Precision-Recall curve: Displays the tradeoff between precision and recall for different thresholds. Also, the metric is appropriate for imbalanced datasets.

V. Dataset

CAX_Startup_Data is a Database that was launched for a Hackaton in Crowd Analytix Platform to create a Machine Learning Model, which predicts whether a startup will succeed or not.

The database was downloaded from Kaggle and it has been published by the end of 2019. The database consists of 472 entries and 116 features, which try to explain the current status of a Startup (for more details, go to Annex 1).

Data cleaning was an essential step because the database had several errors: typo mistakes, inconsistent capitalization, mislabeled classes, and missing values. Throughout this process, we analyzed the data and proceeded to uniform each feature according to the variable's nature.

First of all, we started by correcting structural errors, specifically uniforming ”NaN” values since having different classes could have generated problems when identifying the missing information and correcting it. Although ”Not Applicable” and ”None” can be considered as part of ”NaN” values, we did not correct them because they are possible answers to the question related to each feature.

We continue the Pre-Processing emphasizing the correction of the column ”Industry of Company” due to its relevance to the research. Some rows did not have this information; however, it was possible to infer it from the column ”Focus Functions of the Company”.

After completing the industry, we checked for irrelevant columns that could misinterpret the results of the prediction models. To accomplish this task, we took two approaches: firstly, we calculated the percentage of missing values of every feature and eliminated all the columns that had more than 35% of the entries with ”NaN” values. Secondly, we erase columns that offered similar information such as ”Country of the Company” and ”Continent of the Company” or ”Experience in Fortune 100, 500 and 1000 organizations” that have over-lapping in the information, considering that someone who has worked in Fortune 100 will also be checked in 500 and 1000.

After cleaning the dataset, it was necessary to redefine the industries while continuing to correct the typo and capital mistakes. Then, in order to minimize the presence of outliers and the dispersion of our numerical data, we decided to transform it into intervals. Subsequently, we completed the remaining missed information applying the strategy of the most frequent item, given that a significant part of our features was nominal.

Finally, we categorized and normalized the features (except the ”Dependent Company-Status”) in a range from 0 to 1 due to the usage of some classification models that do not accept negative values. Also, we changed the headers to make them more identifiable when running the models.

This work was implemented through a Pre-Processing algorithm and the processed database comprised 472 startups with 101 features, which was the primary input for feature selection.

VI. Implementation of the code

A. Downloading the data

The implementation of the code started with the downloading of the dataset through Kaggle’s API. To perform this task, first, it was necessary to generate the credentials from our Kaggle’s profile and place them in the “.kaggle” folder. Then, we established the direction from where the database will be downloaded and subsequently stored. (See Fig. 1)

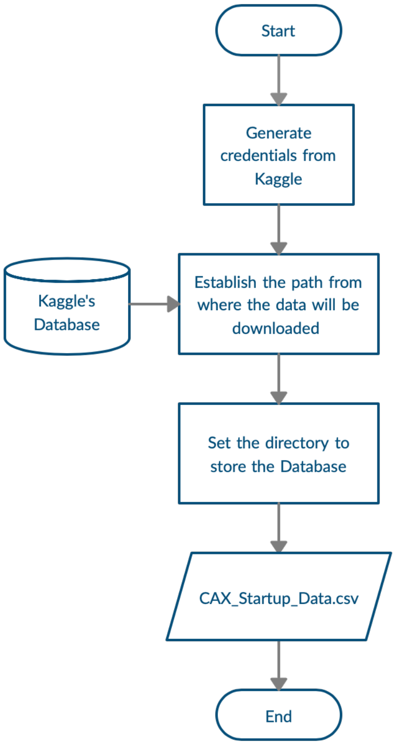
Start
Generate credentials from Kaggle
Kaggle's Database
Establish the path from where the data will be downloaded
Set the directory to store the Database
CAX_Startup_Data.csv
End

Fig. 1. API’s Flowchart

B. Pre-Processing

For the Pre-processing implementation, we used as input the csv file generated when downloading the data. After data cleansing: as explained above, we were able to fill the missing data applying the SimpleImputer, which is a special package designed to complete missing values in this case, using the strategy parameter as the most frequent value of each column.

Then, we used the astype(‘categories’).cat.codes package from Pandas library to categorize data; by default, it is sorted alphabetically. Finally, we normalized the data using the package MinMaxScaler.

C. Data Analysis

In order to analyze the data and have some insights of its behavior, we decided to generate some graphs using the library Seaborn. It has some specialized graphs for categorical variables, which is our case.

D. Feature Selection

As mentioned above, to choose the adequate features, we went through two methodologies: K-best and correlation methodology. To apply them, we used SelectKbest package and corrwith, respectively. Finally, we calculated the correlation between the features pre-selected and chose those that had less intercorrelation. (See Fig. 2)

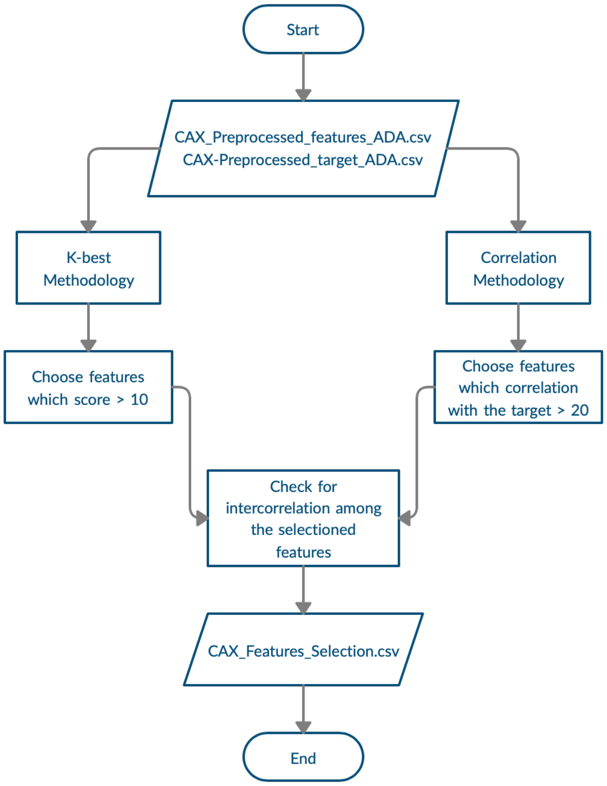
Start
CAX_Preprocessed_features_ADA.csv
CAX-Preprocessed_target_ADA.csv
K-best Methodology
Correlation Methodology
Choose features which score > 10
Choose features which correlation with the target > 20
Check for intercorrelation among the selectioned features
CAX_Features_Selection.csv
End

Fig. 2. Feature Selection Flowchart

E. Models Implementation

Looking for the best model, we started applying train_test_split to divide our data between training and test samples. Subsequently, we applied GridSearchCV to find the optimal values for the hyperparameters related to each model, taking care of not running into overfitting problems. In some cases, the ranges where the hyperparameter was searched were reduced to a small interval for computational efficiency. Once the optimal hyperparameters were found, the respective model was applied and its precision was calculated so they could be evaluated and compared. (See Fig. 3)

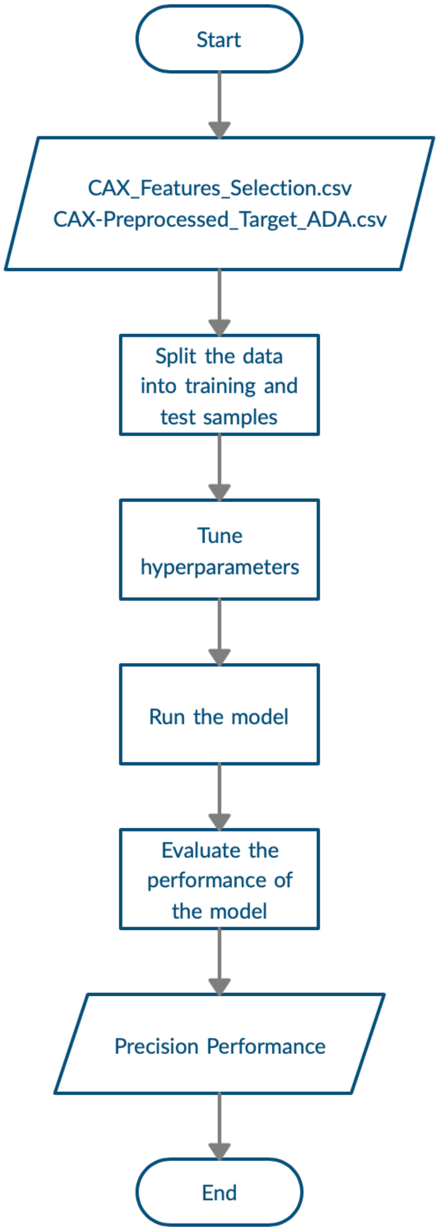
Start
CAX_Features_Selection.csv
CAX-Preprocessed_Target_ADA.csv
Split the data into training and test samples
Tune hyperparameters
Run the model
Evaluate the performance of the model
Precision Performance
End

Fig. 3. Model Application Flowchart

VII. Code Maintenance

The outcome of this project's code implementation is to obtain a methodology that, through a predictive model, can establish the potential of a startup to succeed or not, that in this pilot stage is entirely based on the example database. Therefore, the proper maintenance and update of this code are paramount for the next step of the development of our proposed solution of the shortcomings identified.

The codebase for the prediction model is updated in a Github public repository. It will allow to share the current findings with other interested developers and to keep the projects' code stored adequately for further advances. All the code is appropriately commented and documented to ease its use by an outside party.

The adaptability and replicability of this work are essential to accomplish the global objective. Therefore, the project's code will be updated to adjust the structure to the incoming information, once the database for the Bolivian startup's ecosystem is available.

VIII. Results

After implementing the code, we obtained the following results. During data analysis we found some interesting discoveries, first, it seems that the level of education could be a good factor to analyze when trying to know the potential of a startup, given that according to the graph there are more successful startups when the founder has a Master or PhD degree.

Then, it can be seen that the number of advisors is not important to determine the potential of a startup the same as having work for one of the top companies. Finally, we identified that neither being part of an incubation process nor the number of investors in seed have significant impact when determining the potential of a startup. (See Fig. 4)

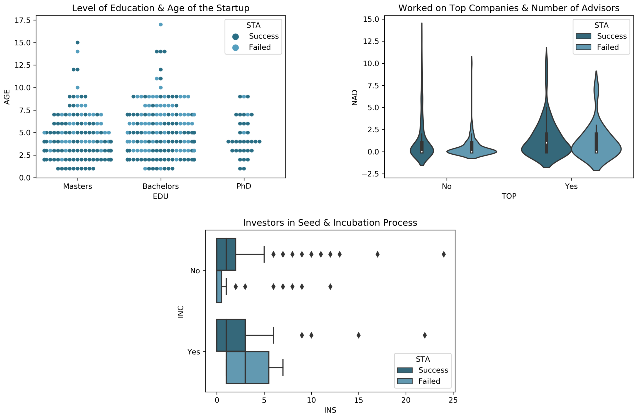

Fig. 4. Data Analysis

As mentioned before, to find the features that had more explicative power concerning the target, we performed two methodologies. In the k-best method, the features were selected based on the chi2 score, while the other methodology took as score the correlation with the status of the startup. In both cases, our decision was based on a threshold, as you can see in Fig. 5.

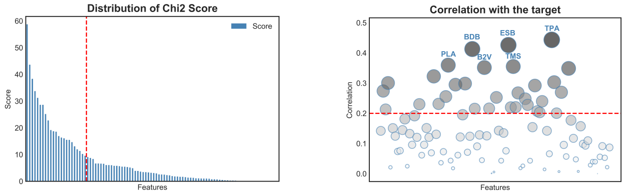

Fig. 5. Feature Selection

From above, we obtained a pre-selection of the features, and checking for multicollinearity we were able to establish the twelve most informative and independent aspects to take into account when analyzing a startup. These factors were compared with those found in the literature, and we discover that 25% are the same: Industry of the Company (IND), Average Years of Experience of Founders (AYE) and Survival through recession (SUR). These results can be summarized in the table I.

TABLE I
FEATURE SELECTION

K-best Selected	**Correlation**	**Features Selected**	**Theoretical Features**
TPA BDB	TPA ESB	AYE	REP
B2V PLA	BDB PLA	BIG	RED
LTR SIZ	TMS B2V	BAR	TOP
PAB CAV	SUR CAV	IND	TIE
AWA ESB	SOL SIZ	DWF	IND
E10 CCA	PAB PAR	PST	NAD
INC PPP	LTR IAS	CPB	PAS
AYE SOL	DIS RES	REC	SAS
CDA PST	CPB EDU	SUR	AYE
MLB EDU	DWF AWA	CDA	EDU
REC BIG	CDA AVC	INC	SUR
	PST REC	MLB	
	E10 AYE		
	CCA IND		
	PPP BAR		
	INC		

When performing the models, we calculated its precision, so we could choose the highest value among them, the results can be seen in table II.

Analyzing the precision we found the best model is Random Forest, the value of the tuned hyperparameters are:

- criterion: Gini
- max depth: 4
- max features: 2

The result we found seems logical given that Random Forest is an embedded model of several uncorrelated Decision Trees, through a third filter when selecting the features to use. The outcome is reliable as the overfitting control approaches were deployed.

TABLE II
PRECISION SCORES

Model	**Macro Precision (%)**
K-Nearest Neighbors	87.0923
Naive Bayes	78.3560
Logistic Regression	82.9423
Decision Tree	82.6654
Support Vector Machine	83.4656
Random Forest	88.0796
Extreme Gradient Boosting	87.5755
Bagging	81.6779

A more graphical way to explain these results can be through the precision-recall curve where we can see that the curve that has the highest peak (has the lowest tradeoff between recall and precision)in the right up corner is the one that corresponds to the Random Forest model (See Fig. 6).

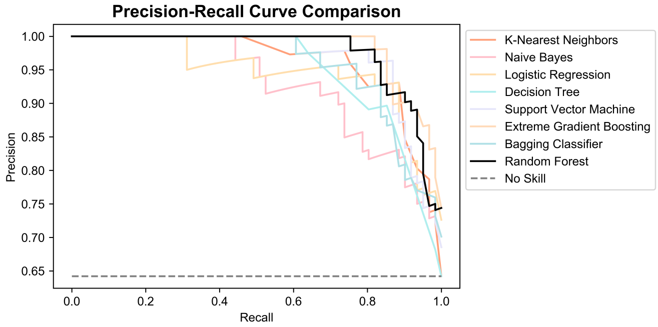

Fig. 6. Precision-Recall curve

IX. Conclusions

There is an undiscovered potential in the Bolivian startup ecosystem. It requires out-of-the-box thinking approaches to find a solution to enhance our market. However, by developing a methodology to assess Bolivian startups is a concrete step. We are aware that the applied concepts and models existed a long time ago, yet, for the Andean country, that step lands on innovation, providing opportunities to further discussion, feedback and, most important, deeper research.

We established a methodology to pre-process a dataset, making available to display statistical insights of it. Then, by applying a two-filter approach, we selected the most informative features that investors and entrepreneurs must be aware of. Models were selected and hyperparameters were tuned thinking on overcome obstacles regarding the dataset nature. Last, we deliver our outcomes in a comprehensive and explicative way.

Finally, it is essential to note that this time a pilot database was employed due to the Bolivian startup market cannot provide one, yet it will have comparable characteristics, reflecting sizes and features as the pilot. It will take an integrated collaboration of the entire startup ecosystem to produce such a database, but once it is released, the developed methodology will be applied, producing more exciting conclusions.

References

[1] Mapeo TIC Bolivia,
`https://mapeoticbolivia.org/homepage`

[2] Ecosistema del emprendedor Paceño,
`https://bit.ly/3d70HlH`

[3] Fundempresa,
`https://bit.ly/2A0uIF3`

[4] Are ”Better” Ideas More Likely to Succeed? An Empirical Analysis of Startup Evaluation, *Scott, E., Shu, P. Lubynsky, R. (2016)*

[5] Attracting Early Stage Investors: Evidence from a Randomized Field Experiment, *Bernstein, S., Korteweg, A. Laws, K. (2014)*

[6] A Business Success Versus Failure Prediction Model for Entrepreneurs with 0-10 Employees, *Lussier, R. Corman, J. (1996)*

[7] Success and Risk Factors in the Pre-Startup Phase, *Van Gelderen, M., Thurik, R. Bosma, N. (2005)*

[8] The Precise Effect of Multicollinearity on Classification Prediction. 40. 5-10, *Lieberman, Mary Morris, John. (2014)*

[9] ASMOTE for high-dimensional class-imbalanced data. BMC Bioinformatics 14, 106., *Blagus, R., Lusa, L. (2013)*

[10] Principles and Theory for Data Mining and Machine Learning. 594, *Clarke, B., Fokoue, E., Zhang H., (2009)*

[11] Sample Size and Robustness of Inferences from Logistic Regression in the Presence of Nonlinearity and Multicollinearity, *Bergtold, J., Yeager, E., Featherstone, A., (2011)*

[12] Selecting training sets for support vector machines: a review. Artif Intell Rev 52, 857–900, *Nalepa, J., Kawulok, M., (2019)*