Linear regression and banking stress-test

Wednesday 19th January, 2022 - 01:21

Issam Jomaa
University of Luxembourg
Email: issam.jomaa.001.student@uni.lu

This report has been produced under the supervision of:
Giacomo DI TOLLO
University of Luxembourg
Email: giacomo.ditollo@ext.uni.lu

Abstract

Our third Bachelor semester project is an attempt at legitimizing the use of machine learning as a tool to do stress tests in the banking system. In this paper, we will first present the capital adequacy ratio and why we will use it as our benchmark for banks. We will then try to accurately predict this measure by using various machine learning models. Hopefully, by the end of this paper, we will have convinced the reader that using the CAR as a measure, machine learning can effectively be used as a tool to stress test banks.

1. Plagiarism statement

This 350 words section without this first paragraph must be included in the submitted report and placed after the conclusion. This section is not counting in the total words quantity.

I declare that I am aware of the following facts:

- As a student at the University of Luxembourg I must respect the rules of intellectual honesty, in particular not to resort to plagiarism, fraud or any other method that is illegal or contrary to scientific integrity.
- My report will be checked for plagiarism and if the plagiarism check is positive, an internal procedure will be started by my tutor. I am advised to request a pre-check by my tutor to avoid any issue.
- As declared in the assessment procedure of the University of Luxembourg, plagiarism is committed whenever the source of information used in an assignment, research report, paper or otherwise published/circulated piece of work is not properly acknowledged. In other words, plagiarism is the passing off as one's own the words, ideas or work of another person, without attribution to the author. The omission of such proper acknowledgement amounts to claiming authorship for the work of another person. Plagiarism is committed regardless of the language of the original work used. Plagiarism can be deliberate or accidental. Instances of plagiarism include, but are not limited to:

1) Not putting quotation marks around a quote from another person's work
2) Pretending to paraphrase while in fact quoting
3) Citing incorrectly or incompletely
4) Failing to cite the source of a quoted or paraphrased work
5) Copying/reproducing sections of another person's work without acknowledging the source
6) Paraphrasing another person's work without acknowledging the source
7) Having another person write/author a work for oneself and submitting/publishing it (with permission, with or without compensation) in one's own name ('ghost-writing')
8) Using another person's unpublished work without attribution and permission ('stealing')
9) Presenting a piece of work as one's own that contains a high proportion of quoted/copied or paraphrased text (images, graphs, etc.), even if adequately referenced

Auto- or self-plagiarism, that is the reproduction of (portions of a) text previously written by the author without citing that text, i.e. passing previously authored text as new, may be regarded as fraud if deemed sufficiently severe.

2. Introduction

The 2008 financial crisis, was a real eye-opener in the world of finance for all the parties concerned as it showed us how fragile was the economic system and how easy it was for everything to collapse. Especially Banks that were left under-capitalized as lots of customers defaulted on their loans. This had as a consequence a severe gain in interest by the banking world in stress testing as a testing methodology to evaluate how well prepared are banks to other catastrophic events like the 2008 financial crisis. Stress testing is a computer simulation technique that tries to reproduce events like financial crises by changing variables to a bank's data and trying to predict accurately if the institution tested is resilient to those conditions. Strategies are then set in place if necessary to try and mitigate against the possible losses. Usually, there are three main types of Stress testing:

1) Historical stress testing using recorded data from past events like the crisis of 2008.
2) Hypothetical stress testing, where the data is specifically tailored for the company. For example, how an oil company would do financially if a war broke out in the middle east.
3) Simulated Stress test where the data is randomized using probabilities. They are usually offered by specialized firms for the use of other companies.

Stress testing is now the norm with various organizations like the European Banking Authority (EBA) indulging in frequent stress testing often yearly. But this Stress testing method raises a question. Is there a measure that can effectively measure how well is a bank doing and, can we predict this measure through data?. This is what our project aims to do, we will try to find a measure that is accepted and used by the scientific community when trying to benchmark banks. We will then try to find a way to properly predict it using multiple ways and see which is the most efficient one.

3. Project Description

3.1. Domains

The main idea behind this semester project is for the student to dabble for the first time in the world of Data Science, and more precisely the world of machine learning. Since Data Science is usually not a whole subject but a tool we use to try to understand and manipulate data that is created in the modern world, we need a subject to use on our data science and machine learning techniques. In this project case, we will be working on the world of finance, and to be more precise the banking area. We will be collecting the data of various banks, and try to answer questions related to stress testing banks. As such, we find ourselves with two domains. One for our scientific part, and one for our technical part.

1) The banking system.
2) Data analysis, and machine learning.

The banking system

"The modern banking industry is a network of financial institutions licensed by the state to supply banking services. The principal services offered relate to storing, transferring, extending a credit against, or managing the risks associated with holding various forms of wealth[3]. This industry is always changing in accord to new regulations, as well as other external factors like the economy and advances in information and communication technologies. The problem that banks are confronted with now is that the economy is so complex that their basic functions are at risk since we have previously mentioned that crisis can bring banks down with not enough capital which in theory should never happen. It is this complexification of the economy that leads the industry to use specialized techniques and algorithms that try to bring down this complexity through the use of computer technologies to something as simple as possible. This is what data science and machine learning are all about.

Data analysis, and machine learning.

Data science combines multiple fields, including statistics, scientific methods, artificial intelligence (AI), and data analysis, to extract value from data. This data can come from any source that can be measured. Usually, data that is being treated by data scientists is especially complex and features lots of dimensions. Analyzing data starts by preparing the data, including cleaning it and filling voids as well as fixing mistakes, and reducing noise. It is then manipulated and shaped into a form that is well suitable for use.
Machine learning could be considered as a subbranch of data science. Wherever data science tries to make data more understandable for humans, machine learning is a broad term that encompasses all algorithms and techniques that can make a machine learn

from data that it is fed to and try to make predictions out of it. So data science is immensely involved in machine learning throughout both the data that is fed to the machine and the data that is outputted.

3.2. Deliverables

Through the description of our objectives we can get a global view on what kind of deliverables we should expect having at the end of our project.

The scientific deliverable

The main scientific deliverable will be divided into multiple parts. In the first part, we will justify the usage of the capital adequacy ratio as our measure of the financial viability of a bank. Then we will detail our analysis of the multiple models of linear regression we are going to build. Our analysis will be based on multiple statistical measures to make sure we are doing a scientific and unbiased analysis. This first deliverable will include the various tables for each model we built as well as all the measures we had on it. These models will try to predict the CAR correctly and with this, we can affirm if stress-testing is viable for banks.

The technical deliverable

The main technical deliverable will be our various Python scripts that will ingest the raw data of banks we have and build the various model that we will use to predict the CAR. These scripts will each have a specific task and will use various techniques of data science and machine learning with the help of multiple libraries.

4. Scientific Deliverable

In order to achieve the goal that was set at the start of the project, which is to conclude if we can use stress-testing to effectively benchmark the performance of a bank in a precise scenario, we need to choose a measure that can precisely fill that role. The capital adequacy ratio is the measure that we will use in our project by which we will benchmark banks.

Capital adequacy ratio

Capital adequacy ratio or “CAR” is one of the measures which ensure the financial soundness of banks in absorbing a reasonable amount of loss. Capitals have components divided into three groups:

- Tier I capital:
  it includes paid-up capital, statutory reserves, disclosed free reserves, Perpetual Non-cumulative Preference Shares, Innovative Perpetual Debt instruments, and capital reserves representing surplus. This tier is usually referred to as the core capital and can absorb losses without the bank being required to stop working meaning that it protects depositors.
- Tier II capital:
  Tier II capital: it includes undisclosed reserves, revaluation reserves, general provisions and loss reserves, hybrid capital instruments, subordinated debt, and investment reserves account. They qualify as regulatory capital and, also protect depositors but to a lesser degree.
- Tier III capital:
  Assets that are limited to 250% of a bank’s tier I capital while being unsecured subordinated and have a minimum maturity of 2 years.
  The CAR is calculated mathematically as being the ratio of a bank’s capital in relation to its current liabilities and risk-weighted assets otherwise known as (RWA).

$$CAR = \frac{Tier I + Tier II + Tier III\ capital}{Risk\ Weighted\ Assets\ (RWA)}$$

Capital adequacy ratio formula [2]

Risk-weighted assets are the sum of a bank’s assets, weighted by risk. Banks usually have different classes of assets, such as cash, debentures, and bonds, and each class of asset is associated with a different level of risk.
Risk weighting is decided based on the likelihood of an asset to decrease in value. Asset classes that are safe, such as government debt, have a risk weighting close to 0%. Other assets backed by little or no collateral, such as a debenture, have a higher risk weighting. This is because there is a higher likelihood the bank may not be able to collect the loan.
Different risk weighting can also be applied to the same asset class. For example, if a bank has lent money to three different companies, the loans can have different risk weighting based on the ability of each company to pay back its loan.

Although the definition of CAR is invariable in itself, the various capitals and RWA may be subjected to

changes in the way they are calculated and represented in the changes.
For that reason, using international guidelines is the best way to ensure we are up to date, and using CAR as a benchmark of a bank's financial soundness makes sense.
Basel is a set of international banking regulations by the Basel Committee on Bank Supervision. The Basel III being the newest one is the one we are going to follow. It sets a new way where Tier III capital is not being taken into account and sets the minimal threshold for the CAR to be 8%.[2]

Us using CAR for this project is backed not only by the claims of international regulations but also by various scientific papers that were published in renowned publications like the International Review of Economics and Finance.
In total, we have reviewed 7 different scientific papers that all use the CAR in different uses cases to benchmark the performances of banks in stress tests. [1] [4] [6] [7] [8] [9] [14].

Now that we have a measure by which we can benchmark banks, we need to find a way by which we can predict the CAR of banks based on certain factors which is what we need in order to stress test banks. The more factors we can incorporate in our predictions the better since it will reflect better a real-world situation and make our stress testing much more realistic. At the start of the project, we were discussing doing both the neural networks and linear regressions approach, but since we wanted to compare our approach to another project that was made previously with neural networks on the same bank of raw data, we decided to only do linear regression.

Linear regression

Linear regression is the name of a machine learning algorithm used to model the relationship between two or multiple parameters by fitting a linear equation on the observed data. Usually, this is done using the Least-square regression that minimizes the sum of squares of the vertical deviation from each data point on the line. In other words, given a regression line that passes through the data, the distance of each point from it is calculated then squared before summing all of them together. The algorithm then tries to reduce this distance by playing with the constants in the equation representative of the line.

Linear regression is a fairly simple but powerful algorithm. This simplicity is what makes it faster to execute than neural networks. A typical model that we build and train on the same data will be done in a few seconds with linear regression while it might take hours for neural networks. Neural networks also might necessitate especially powerful hardware like dedicated GPU's which makes it even less practical with the current shortage of hardware at the time of writing this paper.

But this simplicity comes at a cost elsewhere, while Neural networks can give good results no matter how complex the data is as long as there is some sort of correlation between the data, linear regressions needs a more straightforward approach where it might fail as the data gets more and more complex with a correlation that is distributed between the factors which make it harder to find linearity.
That is why the use of linear regressions is usually preceded by plotting the data in scatter plots to try and find some sort of relationship between the data that can be spotted by the naked eye. In case we do see something on the plots it makes it much more reassuring and reinforces the confidence in the choice of using linear regression to try and predict data. We also used this fairly common approach.(350 words)

"A linear regression line has an equation of the form

$$Y = a + bX$$

where X is the explanatory variable and Y is the dependent variable. The slope of the line is b, and a is the intercept (the value of y when $x = 0$)." [10]

Generalized linear model

The basic linear regression predicts a certain value as a linear combination of a specific set of observed values. This means that a change in one or multiple predictors affects immediately the response variable. This type of prediction is useful in the case of simple data where the predicted variable varies a little compared to the variation of the predictors. However, for complex data that is varying over a wide range ordinary linear regression is not very effective. "Generalized linear models try to fill the gap by allowing by allowing for response variables that have arbitrary distributions (rather than simply normal distributions), and for an arbitrary function of the response variable

(the link function) to vary linearly with the predictors (rather than assuming that the response itself must vary linearly)." A GLM consists of three elements:

1) Linear predictor
2) Link function
3) Probability distribution or exponential family

The linear predictor is the linear combination of parameter b and explanatory variable x. The link function is what links the linear predicted and the probability distribution. There are many link functions and usually, they are used depending on what kind of data we are trying to predict and in which range are we expecting it to be.

Evaluation of a model

Once we finally build a model through linear regressions, we need a way to correctly measure the correctness of this model. Is it predicting values correctly? and if not how wrong is it. Are the errors negligible and the performance of the model good, or is it terrible and should be discarded. All of these evaluations will go through different statistical measures, that are used all around the world to benchmark the performance of machine learning models.

Uncorrected sample standard deviation.
It is the measurement of dispersion in statistics in other words, how spread out the data is. The formula is:

$$\mathrm{s} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(x_i - \overline{x})^2}$$

where $\overline{x}$ is the mean, and N is the number of elements.

Mean squared errors.

It is used to determine how close a regression line is to a set of points. We calculate it by taking the distance of the points from the regression and squaring them to avoid any negative number, the squaring is also important as it helps give more weight to larger differences. We then just calculate the mean of what we found. The lower this number is the better the prediction.

$$\mathrm{MSE} = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2.$$

where n is the number of predictions, Y_i the variable being predicted, and $\hat{Y}_i$ being the actual prediction of that variable.

Root mean squares errors.

It represents the standard deviation of the prediction errors. by that, we mean that it tells us how concentrated the data is around the line that we predict. To calculate it we can take the square root of the Mean squared errors.

$$\mathrm{RMSD} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2}.$$

R Squared.
Represents how well the data fits on the regression line. More generally, it is used to analyze how the difference in one variable can be explained by other variables. In the case of regression, we can reason in percentages and say that the closer the measure is to 1 the closer the points are to the regression line up to 1 where 100% of the points are on the line. Generally, this would mean that the higher R-squared is the better the results we have but this can be false in some edge cases. It is calculated by squaring the correlation coefficient calculated with this formula

$$R^2 = 1 - \frac{SS_{\mathrm{res}}}{SS_{\mathrm{tot}}}$$

where SS_{res} is the sum of squares of residuals, and SS_{tot} is the residual sum of squares.

F-statistic.
The $F-statistic$ in a regression is a value that represents how well you improved the regression line compared to a regression line with all the coefficients = 0. if your model significantly improved the model fit then you will get a better F-statistic.But before taking into account the F-value one must first look at the P-value that is calculated at the same time as the F-statistic.

P value.
With the F statistic calculation comes the P value. Usually the P value is looked at before taking the F statistic into account. If the P value is lower than the alpha level, then we can reject the null hypothesis and we can consider the F value otherwise the F values is worthless.

T-values.
It's a value that represents the size of the difference relative to the variation in the data. The bigger T is the more difference there is and the more it is likely for us to have a null hypothesis. On the contrary, the closer to 0 t is, the less likely there is a difference.

Akaike's information criterion.
This is a mathematical method to know how well the model fits the data it was generated from based on the number of independent variables and how well the model reproduces the data. The best score is obtained when the greatest amount of variation is explained using the least amount of variables. The AIC is calculated as:

$$AIC = 2K - 2ln(L)$$

Where K is the number of independent variable used, and L the log-likelihood

4.1. Our contribution

Now that we explained everything related to our project, we can dive into our approach to the problem. We will step by step present how we worked our way to build a model that according to our benchmarks is performing well.

Now that we explained everything related to our project, we can dive into our approach to the problem. We will step by step present how we worked our way to build a model that according to our benchmarks is performing well.

We must first present the data we are working on. The data we are going to use for this project was also used in this paper. It covers a period of twelve years from 2007 to 2019. This data involves 20 different variables that cover four different quarters. The variables are:

- net loan: Net loans and leases exposure
- dep: Total deposits
- loss allow: Loss allowance to loans
- yield ea: Yield on earning assets
- fundc ea: cost of funding earning assets
- inc aa: Noninterest income to average assets
- CAR: Total risk-based capital ratio
- tot asst: Average total assets
- tot eq: Average total equity
- tot loan: Average total loans
- risk dens: Risk weight density
- GDP growth: Gross Domestic Product growth
- export growth: US real exports of goods and ser vices growth
- debt GDP: US public debt to GDP
- govex GDP: US government expenditure to GDP
- inflat: Implicit price deflator as a measure of US inflation
- HPI growth: House Price Index growth
- unemp: Unemployment rate (age 15-64)
- Yield 10Y: 10-year US sovereign bonds yields
- SP500 ret: SP 500 quarterly returns

All the data will be provided both in excel and csv with the paper. Since we are going to use this data with linear regression the first step was to try and understand the data to try to see if we could figure out anything. After trying a bunch of methods, we decided to plot the capital adequacy ratio against all the other variables to see if we could determine some pattern visible to the naked eye. Since we are using linear regression this seemed to be the most appropriate way. After all the possible combinations were plotted, we could not find any visible relationship. This indicates that the relation between the variables is more complex and not very straight forward which is a major drawback as it makes the usage of linear regression less convincing. The plots will all be included in the annex **??**.

Nonetheless we continued and built our first linear regression model using the Sci-kit learn library on Python(For technical details refer to:).

At this point in the project, we will not be using raw data anymore. We will be using pre-processing techniques on the data to make it more digestible by the machine learning models we are going to build. We will first replace the missing data in the columns by the mean of that value. Then we are going to normalize the data, which means that all the values inside the tables will be transformed and adjusted to fit in a 0 to 1 set. The formula used will be the same as used in the paper [13].

$$x_i' = log_u(|min(0, x_{min})| + x_i + 1)$$

$$u = x_{max} + 1$$

where x_{min} is the minimum value of that column, x_{max} is the maximum value of that column, x_i is the value we want to normalize, and x_i' is the normalized value.

We will apply these techniques on two different data sets one is the data we start with and the other is the same data but with the log function applied to it. We now have two different data sets that we will call raw_data and log_raw_data. We then train a basic linear regression model on these two data sets. Since we want to compare the different approaches and not get the best result for a specific model we will not split the data in training and testing sets. The results we get with the raw and log data are in the Table annex 5

We can see that the log table is performing better than the raw data table. The R^2 and adjusted R^2 are

bigger and the AIC is lower.With this, we can clearly state that applying the log function to our data was very benefic and we recommend doing it if possible as the benefit is non-negligible.

In our table, we can also see the P values of each predictor with the P values that are smaller than 0.05 in bold. Since the results are not good enough yet, we decide to try and take out all the predictors that had P values > 0.05. We call theses two new data sets raw_data_P, and log_raw_data_P We then retrained our model with this new data that is supposed to perform better. But it seems that it in fact did not improve our results, it even worsened the results in the case of the Log data. 5

The adjusted R^2 stagnated at 0.781 for the raw data while it deteriorated from 0.888 to 0.887 for the log data. We can conclude that this technique was not effective in the case of using it with linear regression. It is also important to note that we added a coefficient at the start of the data and that it was very important to do so. In case we did not add the coefficient the R squared reported would have been uncentered and would have been higher than what it was. During our test when we did not add the coefficient our model report an R^2 of 0.999 which was false in this case once we tested it.

Since we could not improve our model by manipulating the data, we decided to try and change the core of the model which is linear regressions with another type of linear regressions called Generalised linear models. We tried every type of model on our data and here are the best results we got. These results were achieved using the Gaussian Family and the inverse power Link. Since we already observed that the log data was better performing than the raw data we decided to only use it for this model. The results are once again in the Table annex **??**.

As we can see, the results improved tremendously The R^2 is now 0.998. The P measure is still at 0.0 while the log-likelihood went up and the AIC went further down. This result is almost perfect for that reason we decided to do another round of testing to confirm the results. To do this assessment we will test all of the models we previously built.

4.2. Assessment

We will each time divide our data set into training and validation sets. We then train the model with these sets and calculate the R^2 plus the predicted R^2 . For each model, we redo this 30 times and record each time the R^2 obtained. Afterward, we will calculate the min, max, mean, and std of the multiple R^2 recorded for each model. The predicted R^2 will indicate how well our model predicts responses for new observation, this will make sure that the very performing model is not just over fitting. Our findings are reported in the annex 5 as the summary table.

The results are in line with our previous findings. The predicted R squared is within what we expected, and our calculation shows no great variance between the multiple R squared calculated. This proves that our models are all performing as expected, it also demonstrates that our last model built on generalized linear models is outputting promising results. These results are far better than what we expected, they are also better than the results of neuronal networks trained on the same exact data. This proves that stress testing can be effectively used since we can accurately predict the capital adequacy ratio of banks based on the features we have on our data. It is important to stress out that we have done some analysis to understand which is the method do be used on our data and that we have chosen the current one based on multiple experiences and trial and errors. Please be aware that this is not a general method and it comes from a carefully design choice and data-driven considerations.

5. Tables

TABLE 1. Linear Regression normal values & Log values / GLM Log values & normal values

Predictors	Coefficient	P > |t|	Coefficient	P > |t|	Coefficient	P > |t|	Coefficient	P > |t|
net_loan	-0.3700	**0.000**	-0.2830	**0.004**	0.2989	**0.034**	1.0930	**0.000**
net_loan_1	0.1401	0.278	0.4206	**0.001**	-0.8497	**0.000**	-0.9808	**0.003**
net_loan_2	-0.1966	0.104	-0.1034	0.402	0.0202	0.918	0.4529	0.178
net_loan_3	-0.1074	0.174	-0.2000	**0.000**	0.3372	**0.000**	0.2248	0.294
loss_allow	0.0370	**0.008**	-0.0196	**0.001**	0.0336	**0.000**	-0.1540	**0.000**
loss_allow_1	0.0741	**0.000**	0.0258	**0.000**	-0.0143	0.189	-0.2247	**0.000**
loss_allow_2	0.0095	0.560	0.0071	0.296	-0.0025	0.809	0.0163	0.710
loss_allow_3	-0.0594	**0.000**	0.0008	0.872	-0.0014	0.861	0.0685	**0.032**
dep	-0.2771	**0.001**	-0.1092	0.145	-0.7510	**0.000**	0.0373	0.863
dep_1	0.3940	**0.000**	0.1370	0.057	0.3579	**0.002**	-0.8630	**0.000**
dep_2	-0.0763	0.289	0.0136	0.819	-0.1556	0.097	0.0445	0.782
dep_3	-0.0319	0.531	0.1037	**0.016**	-0.1334	0.050	0.1105	0.388
yield_ea	0.1153	**0.000**	0.0198	0.123	0.0174	0.381	-0.4456	**0.000**
yield_ea_1	-0.1082	**0.000**	0.0416	**0.026**	-0.1363	**0.000**	0.3150	**0.000**
yield_ea_2	-0.0697	**0.001**	-0.0448	**0.007**	0.1077	**0.000**	0.2730	**0.000**
yield_ea_3	0.0086	0.586	-0.0098	0.470	0.0026	0.911	0.1535	**0.000**
fundc_ea	-0.0374	**0.001**	-0.0067	0.355	0.0231	**0.023**	0.0517	0.109
fundc_ea_1	0.0295	0.058	-0.0084	0.447	-0.0238	0.143	-0.1390	**0.002**
fundc_ea_2	0.0089	0.504	0.0289	**0.006**	-0.0370	**0.019**	0.0660	0.074
fundc_ea_3	0.0404	**0.000**	-0.0276	**0.001**	0.0422	**0.001**	-0.1630	**0.000**
inc_aa	0.0619	**0.002**	-0.0092	**0.008**	0.0117	**0.034**	-0.1679	**0.003**
inc_aa_1	-0.0195	0.622	-0.0153	**0.000**	0.0153	0.009	-0.1439	0.154
inc_aa_2	-0.1840	**0.000**	-0.0107	**0.019**	0.0127	**0.080**	0.6313	**0.000**
inc_aa_3	0.0404	0.272	0.0047	0.264	-0.0096	0.152	-0.1741	0.078
CAR	0.5791	**0.000**	0.6392	**0.000**	-1.6998	**0.000**	-1.1639	**0.000**
CAR_1	0.1504	**0.000**	0.0662	**0.000**	-0.0640	**0.031**	-0.2285	**0.003**
CAR_2	0.0182	0.536	0.0618	**0.000**	-0.0949	**0.001**	-0.0831	0.327
CAR_3	-0.0212	0.460	0.0730	**0.000**	-0.1088	**0.000**	0.1561	0.063
tot_asst	0.154	0.105	-1.0849	**0.000**	0.2239	0.388	3.7361	**0.000**
tot_asst_1	-0.0119	0.951	-0.3915	**0.043**	0.2363	0.471	-0.8209	0.133
tot_asst_2	0.6072	**0.000**	0.6024	**0.000**	-0.4659	**0.087**	-2.5751	**0.000**
tot_asst_3	-0.4036	**0.000**	0.2316	**0.015**	-0.5260	**0.002**	1.6650	**0.000**
tot_eq	0.0625	0.415	1.1335	**0.000**	-0.1986	0.115	-0.8797	**0.000**
tot_eq_1	0.0528	0.544	-0.1111	0.158	0.0106	0.942	-0.0515	0.841
tot_eq_2	-0.0537	0.529	-0.4686	**0.000**	0.8980	**0.000**	0.1319	0.601
tot_eq_3	0.0466	0.475	-0.0759	0.183	0.1519	0.150	-0.2300	0.234
tot_loan	-0.1819	0.213	-0.5194	**0.000**	0.8462	**0.000**	0.1483	0.687
tot_loan_1	0.0877	0.605	0.2609	0.129	-0.0164	0.952	0.1111	0.809
tot_loan_2	0.2861	**0.026**	0.0560	0.641	-0.1204	0.534	-0.2161	0.557
tot_loan_3	0.1728	**0.028**	0.2466	**0.000**	-0.4079	**0.000**	-0.5288	**0.013**
risk_dens	-0.1145	**0.000**	-0.2207	**0.000**	-0.2333	**0.000**	0.3778	**0.000**
risk_dens_1	0.0529	0.095	-0.1025	**0.000**	0.1913	**0.000**	-0.0035	0.969
risk_dens_2	0.0244	0.423	0.0934	**0.000**	-0.1534	**0.000**	-0.0437	0.615
risk_dens_3	0.0051	0.848	0.0772	**0.000**	-0.1155	**0.000**	-0.0144	0.852
GDP_growth	0.0208	**0.002**	0.0323	**0.000**	-0.0778	**0.000**	-0.0816	**0.000**
GDP_growth_1	-0.0129	0.674	-0.0215	**0.000**	0.0560	**0.000**	-0.5260	**0.000**
GDP_growth_2	-0.1545	**0.000**	0.0011	0.598	-0.0074	0.095	-0.0520	0.087
GDP_growth_3	0.0574	**0.000**	0.0266	**0.001**	-0.0459	**0.001**	0.0330	0.178
export_growth	-0.0615	**0.000**	-0.0059	0.056	-0.0145	**0.000**	0.0544	**0.000**
export_growth_1	-0.5562	**0.000**	-0.0099	**0.000**	0.0293	**0.000**	1.6390	**0.000**
export_growth_2	0.0753	**0.000**	0.0040	**0.002**	-0.0048	**0.033**	0.1587	**0.000**
export_growth_3	-0.0889	**0.009**	-0.0124	**0.000**	0.0439	**0.000**	0.8163	**0.000**
debt_GDP	-0.5377	0.601	2.8613	**0.001**	-13.0942	**0.000**	-0.7957	0.745
debt_GDP_1	-26.0942	**0.000**	2.3876	0.150	-17.4408	**0.000**	52.9073	**0.000**
debt_GDP_2	-4.2981	**0.000**	-6.4670	**0.000**	18.1417	**0.000**	8.4754	**0.000**
debt_GDP_3	24.5871	**0.000**	6.6201	**0.000**	-5.0297	**0.000**	-42.3461	**0.000**
govex_GDP	-0.2847	0.586	-0.0490	0.180	-0.1498	**0.000**	15.8558	**0.000**
govex_GDP_1	9.3549	**0.000**	-0.0091	0.299	0.1252	**0.000**	-15.0544	**0.000**
govex_GDP_2	2.0223	**0.000**	-0.0521	**0.000**	0.1371	**0.000**	-1.0264	**0.013**
govex_GDP_3	-0.2562	0.318	0.1095	**0.000**	-0.2363	**0.000**	-3.0311	**0.000**
inflat	79.0444	**0.000**	76.9373	**0.000**	-123.4500	**0.000**	-20.2406	**0.000**
inflat_1	-32.4249	**0.000**	-39.6103	**0.000**	41.4886	**0.000**	51.3091	**0.000**
inflat_2	22.5975	**0.000**	22.4580	**0.000**	-24.7965	**0.000**	-62.3405	**0.000**
inflat_3	-23.2567	**0.000**	-53.8651	**0.000**	92.5075	**0.000**	65.7227	**0.000**
HPI_growth	-17.4524	**0.000**	-19.7581	**0.000**	22.9509	**0.000**	-12.9498	**0.000**
HPI_growth_1	-9.1599	**0.000**	15.7663	**0.000**	-30.3343	**0.000**	18.1973	**0.001**
HPI_growth_2	-8.6975	**0.000**	-8.3684	**0.000**	14.8865	**0.000**	13.6532	**0.000**
HPI_growth_3	-7.4829	**0.000**	2.6759	**0.021**	-8.4043	**0.000**	-1.0464	0.486
unemp	-0.3945	**0.003**	0.3664	**0.002**	-0.5675	**0.000**	1.2399	**0.004**
unemp_1	-1.5341	**0.000**	-0.5918	**0.000**	0.1355	0.319	2.2035	**0.000**
unemp_2	0.5839	**0.039**	-0.0897	0.345	-0.2996	**0.012**	-6.1846	**0.000**
unemp_3	-5.4510	**0.000**	-0.9426	**0.000**	0.8815	**0.000**	10.2602	**0.000**
yield_10Y	-0.0706	**0.027**	-0.1148	**0.000**	0.2577	**0.000**	-0.2472	**0.000**
yield_10Y_1	-0.3063	**0.000**	-0.0223	**0.000**	-0.0052	0.437	0.0047	0.929
yield_10Y_2	-0.0576	0.218	0.1283	**0.000**	-0.1755	**0.000**	0.3904	**0.000**
yield_10Y_3	0.1493	**0.000**	-0.1705	**0.000**	0.4514	**0.000**	-0.9556	**0.000**
S&P500_ret	-0.0219	**0.002**	-0.0201	**0.000**	0.0443	**0.000**	-0.0081	**0**.583
S&P500_ret_1	-0.2444	**0.000**	0.0215	**0.000**	-0.0454	**0.000**	0.6729	0.000
S&P500_ret_2	-0.1606	**0.000**	-0.0243	**0.000**	0.0827	**0.000**	0.2942	**0.000**
S&P500_ret_3	-0.1023	**0.049**	0.0184	**0.000**	-0.0269	**0.000**	0.6804	**0.000**
Statistics: R^2	0.782		0.888		0.9998		0.9164	
$Adjusted R^2$	0.781		0.888	F-statistic	0.888		0.9164	
AIC	-93030		-134500		-136127.9159		-85528.7546	
$Pvalue$	0.00		0.00		0.00		0.00	

TABLE 2. Linear Regression normal values with predictors that had a P |t| ≤ 0.05

Predictors	Coefficient	P > |t|
const	-5.7291	**0.001**
net_loan	-0.4810	**0.000**
loss_allow	0.0389	**0.005**
loss_allow_1	0.0796	**0.000**
loss_allow_3	-0.0526	**0.000**
dep	-0.3325	**0.000**
dep_1	0.3551	**0.000**
yield_ea	0.1014	**0.000**
yield_ea_1	-0.0911	**0.000**
yield_ea_2	-0.0592	**0.000**
fundc_ea	-0.0100	0.079
fundc_ea_3	0.0541	**0.000**
inc_aa	0.0590	**0.000**
inc_aa_2	-0.1569	**0.000**
CAR	0.6317	**0.000**
CAR_1	0.1364	**0.000**
tot_asst_2	0.4075	**0.000**
tot_asst_3	-0.3561	**0.000**
tot_loan_2	0.2426	**0.000**
tot_loan_3	0.0679	**0.012**
risk_dens	-0.0120	0.255
GDP_growth	0.0310	**0.000**
GDP_growth_2	-0.1364	**0.000**
GDP_growth_3	0.0676	**0.000**
export_growth	-0.0605	**0.000**
export_growth_1	-0.5440	**0.000**
export_growth_2	0.0609	**0.000**
export_growth_3	-0.1015	**0.000**
debt_GDP_1	-25.1583	**0.000**
debt_GDP_2	-4.0818	**0.000**
debt_GDP_3	23.8067	**0.000**
govex_GDP_1	8.9331	**0.000**
govex_GDP_2	1.6444	**0.000**
inflat	79.4701	**0.000**
inflat_1	-35.8591	**0.000**
inflat_2	24.8466	**0.000**
inflat_3	-20.3880	**0.000**
HPI_growth	-14.2707	**0.000**
HPI_growth_1	-10.7524	**0.000**
HPI_growth_2	-6.6812	**0.000**
HPI_growth_3	-8.8400	**0.000**
unemp	0.0984	0.306
unemp_1	-1.2217	**0.000**
unemp_2	0.6253	**0.000**
unemp_3	-5.4806	**0.000**
yield_10Y	-0.0857	**0.000**
yield_10Y_1	-0.2777	**0.000**
yield_10Y_3	0.1320	**0.000**
S&P500_ret	-0.0245	**0.000**
S&P500_ret_1	-0.2310	**0.000**
S&P500_ret_2	-0.1417	**0.000**
S&P500_ret_3	-0.0952	**0.000**
Statistics:		
R^2	0.781	
$AdjustedR^2$	0.781	
AIC	-92970	
$Pvalue$	0.00	

TABLE 3. Linear Regression log values with predictors that had a P |t| ≤ 0.05

Predictors	Coefficient	P > |t|
const	12.1814	**0.000**
net_loan	-0.4181	**0.000**
net_loan_1	0.5390	**0.000**
net_loan_3	-0.2231	**0.000**
loss_allow	-0.0162	**0.004**
loss_allow_1	0.0298	**0.000**
dep_3	0.1374	**0.000**
yield_ea_1	0.0601	**0.000**
yield_ea_2	-0.0525	**0.000**
fundc_ea_2	0.0147	**0.035**
fundc_ea_3	-0.0298	**0.000**
inc_aa	-0.0081	**0.018**
inc_aa_1	-0.0148	**0.000**
inc_aa_2	-0.0086	**0.041**
CAR	0.6496	**0.000**
CAR_1	0.0503	**0.000**
CAR_2	0.0697	**0.000**
CAR_3	0.0647	**0.000**
tot_asst	-1.2304	**0.000**
tot_asst_1	-0.2757	**0.025**
tot_asst_2	0.7091	**0.000**
tot_asst_3	0.1280	**0.035**
tot_eq	1.0911	**0.000**
tot_eq_2	-0.5911	**0.000**
tot_loan	-0.2904	**0.011**
tot_loan_3	0.2806	**0.000**
risk_dens	-0.2102	**0.000**
risk_dens_1	-0.1221	**0.000**
risk_dens_2	0.1028	**0.000**
risk_dens_3	0.0721	**0.000**
GDP_growth	0.0303	**0.000**
GDP_growth_1	-0.0332	**0.000**
GDP_growth_3	-0.0043	0.338
export_growth_1	-0.0055	**0.000**
export_growth_2	-0.0010	0.261
export_growth_3	-0.0048	**0.000**
debt_GDP	3.0391	**0.000**
debt_GDP_2	-5.8687	**0.000**
debt_GDP_3	5.9395	**0.000**
govex_GDP_2	-0.0484	**0.000**
govex_GDP_3	0.1162	**0.000**
inflat	66.9787	**0.000**
inflat_1	-43.2423	**0.000**
inflat_2	10.3727	**0.000**
inflat_3	-32.1644	**0.000**
HPI_growth	-8.9211	**0.000**
HPI_growth_1	3.3588	**0.003**
HPI_growth_2	9.1115	**0.000**
HPI_growth_3	-0.2829	0.440
unemp	0.0753	0.202
unemp_1	-0.7511	**0.000**
unemp_3	-1.3173	**0.000**
yield_10Y	-0.1446	**0.000**
yield_10Y_1	-0.0238	**0.000**
yield_10Y_2	0.0776	**0.000**
yield_10Y_3	-0.1798	**0.000**
S&P500_ret	-0.0136	**0.000**
S&P500_ret_1	0.0226	**0.000**
S&P500_ret_2	-0.0215	**0.000**
S&P500_ret_3	0.0170	**0.000**
Statistics:		
R^2	0.887	
$AdjustedR^2$	0.887	
AIC	-134400	
$Pvalue$	0.00	

TABLE 4. Summary of model tests

Data Used	Model	statistic	values
Raw Data	Linear Regression	Predicted R-squared	0.7797282979179458
		R-squared	0.782
		Adj R-squared	0.781
		P	0.0
		min	0.7665523144965829
		max	0.8028269158795125
		mean	0.7797391441771296
		std	0.009001134401114164
Log Data	Linear Regression	Predicted R-squared	0.885335827544602
		R-squared	0.888
		Adj R-squared	0.888
		P	0.0
		min	0.8726676527808044
		max	0.8935438631990753
		mean	0.8830270857724957
		std	0.00639179288745294
Raw Data R	Linear Regression	Predicted R-squared	0.7798553134019632
		R-squared	0.781
		Adj R-squared	0.781
		min	0.7671878471544182
		max	0.7966316666100917
		mean	0.7809500984248432
		std	0.008082050648597891
Log Data R	Linear Regression	Predicted R-squared	0.8853759364364571
		R-squared	0.887
		Adj R-squared	0.887
		P	0.0
		min	0.8666525271842045
		max	0.9063607799133397
		mean	0.8858658002176351
		std	0.010627251527007624
Raw Data	GLM	Predicted R-squared	0.7107356132584344
		Pseudo R-Squared	0.9164
		P	0.0
		min	0.6925739371955923
		max	0.7203831321332419
		mean	0.7088660697266058
		std	0.0074578889592919985
Log Data	GLM	Predicted R-squared	0.892491461096736
		Pseudo R-squared	0.9998
		P	0.0
		min	0.8767593635112512
		max	0.9051090793612955
		mean	0.8932184201144348
		std	0.007959196969854601

6. Technical Deliverable

6.1. Technical Background

Python

"Python is a high-level, interpreted, interactive and object-oriented scripting language" [11]. Compared to other programming languages, Python stands out as being highly readable thanks to it using more English words compared to other programming languages that would often use more punctuation as syntax. Python also features dynamically allocated memory which is especially useful for applications like ours since we will need variables of length that could attain thousand's of Bits. Python also has a wide variety of libraries that are easy to use. That's why it is one of the programming languages we chose to use as it will be a nice first contact with the world of programming.

Pandas

Pandas is a python library that is used to work with datasets. It provides high-performance data manipulation and analysis that are very useful for machine learning projects and data analysis projects in general where we manipulate data all the time. The data frame object that it revolves around proved itself to be fast and efficient.

NumPy

Numpy is a Python library that is used to work with multi-dimensional arrays called ndarray instead of the usual Python lists that are slow and inefficient. Using mathematical operations on arrays is relatively simple and fast which is why it is usually used as a link between pandas and machine learning as the data needs to be prepared to be fed to the neural network.

scikit-learn

scikit-learn is one of the biggest machine learning Python libraries. Developed by the French Institute for Research in Computer Science and Automation). It contains a wide range of functions that facilitates the use and the testing of various machine learning model.

Statsmodels

According to their own website, "statsmodel is a Python module that provides classes and functions for the estimation of many different statistical models, as well as for conducting statistical tests, and statistical data exploration. An extensive list of result statistics are available for each estimator. The results are tested against existing statistical packages to ensure that they are correct"[12]

6.2. Requirements

Our technical deliverables are a series of scripts that we produced to accomplish all we described in the scientific part. The scripts should include a script to clean the data, build our plots, build our model, and finally test the said models and output the tables that we previously showcased. The scripts should be written in Python. The code should be as clean as possible and should need as little interaction with a potential user. The code should also be properly commented out so that it is easier to build on top of it for future researchers.

6.3. Design and production

In this section, we will talk about each file, what they were designed to accomplish and how we did it by adding some code snippets.

The first file is the "dataset.py" file, this file was taken from the BSP of a previous student that worked on the same dataset and we built other functions on top of it.

Data

This file contains all the data functions related, from loading to preprocessing techniques. The functions that start with "get" like $get\_full\_xy\_data()$ or $get\_full\_log\_xy\_data()$ are all loading functions that load the data from the corresponding CSV file and returns it in a NumPy array usually two, one for the predictors and one for the predicted parameter. This way we can directly feed the data for the models. Another two functions in this file that we will use are the preprocessing functions. the functions $replace\_missing$ fill all the gaps in the data with the mean value of that column while using the technique we showcased earlier in the scientific part, the function $normalize$ normalizes the data. Finally, the

get_training_and_test_set shuffles the data and outputs a training set and a validation set with 70% of the data going to the training set and the rest for the validation one.

Linear regression

Using all these functions, we will talk about the *linear_regression.py* file, where we build the first four linear models. All these four models are built on the same principle they only differ in the data they ingest and as such, they are separated into four different functions.

Each function takes three parameters. The first parameter is the name of the file where the model should be saved, the model will be saved in the pickle format. The second parameter is the text file where the table summary of the model will be saved. Finally, the third parameter is the number of times we should change the training and validation sets and test the model.

Each function will train a linear regression model on the data it is associated with, the training will be done as many times as the user inputs to find the best possible model. The best model will be saved and its performance table will also be saved. *linear_regression* will return a linear regression model built on the raw data. *linear_regression_P* will return the same type of model but with the raw data P that we mentioned in the scientific part. *linear_regression_log* will used the log data, and *linear_regression_log_P* will use the log data P.

to load and preprocess the data we use these lines of code:

```
    import dataset
    x_data, y_data = dataset.get_full_xy_data()
    x_data = np.nan_to_num(x_data)
    y_data = np.nan_to_num(y_data)
    x_data, y_data = dataset.replace_missing
     (x_data, y_data)
    x_data, y_data = dataset.normalize(x_data,
        y_data)
```

the only thing that changes between each function is what function in used to load the data.

```
    for _ in range (x):
        training_x, training_y, validation_x,
            validation_y =
            dataset.get_training_and_test_set
            (x_data, y_data)

        linear = linear_model.LinearRegression()
        linear.fit(x_data, y_data)
        acc = linear.score(x_data, y_data)
        print(acc)

        if acc > best:
            best = acc
            with open(name_of_save_file, "wb")
                as f:
                pickle.dump(linear, f)
```

This part of the code will be training and testing the model based on the data that *get_training_and_test_set* will output. We will then record the score of the model in the best variable. This process will be repeated x number of times, and each time a new model has a better result than the recorded one, it will be saved until we get the best model in that amount of iteration.

this line creates the linear regression model thanks to the scikit-learn library

```
    linear = linear_model.LinearRegression()
```

then we use the fit function to train it on the training data.

```
    linear.fit(x_data, y_data)
```

To score the model we use the score function that compares the prediction of the model to what it was supposed to output. this function returns the percentage of accuracy.

```
    acc = linear.score(x_data, y_data)
```

Generalized linear models

To build our generalized linear models, we start with the same block of code that loads and preprocess the data. The only difference is that we add a constant to our *x_data* this will be useful later to calculate the P-value of the F statistic as it will act as the intercept. Here is how we do it using the Statsmodels library.

```
    x_data = sm.add_constant(x_data)
```

These two lines of code:

```
    gaussian_model = sm.GLM(y_data, x_data,
family=sm.families.Gaussian(link=inverse_power()))
    gaussian_results = gaussian_model.fit()
```

will build and train our generalized linear model. In the first line, we set up the family which will be Gaussian, as well as the link that is going to be the inverse power. In the second line, we just use the fit() function that will train the model on the data that we provided in the previous line while building the

model.We can replace the family and link with any option available in the Statsmodels library.

Finally, we can display the results table of our model by using the summary() function like this.

```
gaussian_results.summary()
```

To calculate the F statistic and get the P-value, we follow this method [5]. We first create an identity matrix based on the length of the matrix of the coefficients of the fitted model.We also must take out the coef line.

```
A = np.identity(len(gaussian_results.params))
A = A[1:,:]
```

We then can just show the F statistic as well as the P value of it by using the f_test function. The functions apply to the model we already built and take the matrix we just created as the parameter.

```
gaussian_results.f_test(A)
```

Other files

Other files that we will include with our paper but that we will not discuss in detail are:
$Final\_linear\_regression.py$ is the script that we used to produce part of the Summary table where we test the different models we had. Instructions on how to use it should be available inside the file as comments.
Another file that we used to calculate the predicted R^2 that we also used in the final summary table was $predicted\_R\_squared.py$.
$search\_car\_by\_bank\_ID.py$ is a script that we used at the start of the project to search and plot the car of a bank by its ID. This helped us in the first stage when we were trying to visualize the data and make sense of it.
$plot.py$ is the script that we used to produce all the plots in the Annex. This file reads two separate CSV files. One with the data of the banks with a capital adequacy ratio above or equal to eight and the other with the ones less than eight. The CSV files must both contain only two columns, one with the CAR and the other with the predictor you are trying to plot against it. Finally we will include the CSV files of the raw data as well as the Log data.

6.4. Assessment

All the files are working as intended and we are correctly building the models with promising results. We can nonetheless give an assessment on certain specific points that are important.

- **Modulable**

  One of our requirements at the start of the project was to have a modulable project with well-defined scripts that were not too long. This point has been effectively taken into account throughout the project, and we now have multiple scripts with each a well-defined task.

- **Documentation**

  Another requirement that we had for our technical deliverable was documentation. Our code is well documented both through our description on this paper and also with commented-out lines of code that describe the use for each script.

- **versatility**

  This point is fulfilled but not completely, as some of our scripts especially the ones to plot the data are completely tailored to our data set. Nonetheless, the majority, and especially our model-building scripts should be able to work out of the box with any type of data set.

7. Acknowledgment

I would first like to thank Prof. Giacomo DI TOLLO for being my pat and supporting me throughout the entirety of my 3rd-semester bachelor project at the University of Luxembourg. He answered all my questions throughout the project, and I learned a lot about data science and machine learning.

8. Conclusion

During this project, we tried to see if stress-testing banks was something possible with today's technology and how would that be done. We have proved the usefulness of the Capital adequacy ratio in that matter, and found some promising results as to how we could potentially try to predict this measure using real life data gathered from different banks throughout the world. Our machine learning models had a very good accuracy, and further steps to assess the legitimacy of our claims were taken. With this, we can conclude that the project was a success and a future project in

continuation with this one would be to effectively try our machine learning model on another set of data gathered from other banks and potentially try and stress-test these banks.

References

[1] Viral V Acharya, Diane Pierret, and Sascha Steffen. "Introducing the "Leverage Ratio" in assessing the capital adequacy of European banks". In: *ZEW Discussion Und Working Paper* 49.621 (2016), pp. 460–482.

[2] Moody's Analytics. *Regulation Guide: An Introduction*. https://www.moodysanalytics.com/-/media/whitepaper/2011/11-01-03-regulation-guide-introduction.pdf.

[3] *Banking Industry*. URL: https://www.encyclopedia.com/social-sciences/applied-and-social-sciences-magazines/banking-industry (visited on 12/30/2021).

[4] Leila Bateni, Hamidreza Vakilifard, and Farshid Asghari. "The influential factors on capital adequacy ratio in Iranian banks". In: *International Journal of Economics and Finance* 6.11 (2014), pp. 108–116.

[5] *F-statistic: Understanding model significance using python*. URL: https://medium.com/analytics-vidhya/f-statistic-understanding-model-significance-using-python-c1371980b796 (visited on 06/01/2022).

[6] Nicolás Gambetta, María Antonia García-Benau, and Ana Zorio-Grima. "Stress test impact and bank risk profile: Evidence from macro stress testing in Europe". In: *International Review of Economics & Finance* 61 (2019), pp. 347–354.

[7] Andreas Hadjixenophontos and Christos Christodoulou-Volos. "Financial crisis and capital adequacy ratio: A case study for Cypriot commercial banks". In: *Journal of Applied Finance and Banking* 8.3 (2018), pp. 87–109.

[8] M Kabir Hassan, Omer Unsal, and Hikmet Emre Tamer. "Risk management and capital adequacy in Turkish participation and conventional banks: A comparative stress testing analysis". In: *Borsa Istanbul Review* 16.2 (2016), pp. 72–81.

[9] Ali Jamali. "Modeling effects of banking regulations and supervisory practices on capital adequacy state transition in developing countries". In: *Journal of Financial Regulation and Compliance* (2019).

[10] $linear_regression$. URL: http://www.stat.yale.edu/Courses/1997-98/101/linreg.htm (visited on 12/31/2021).

[11] *Python_overview*. URL: https://www.tutorialspoint.com/python3/python_overview.htm (visited on 12/30/2021).

[12] *statsmodels*. URL: https://www.statsmodels.org/stable/index.html (visited on 12/30/2021).

[13] Krystyna Tsaryk. "Artificial Neural Networks and Deep Learning for stress testing a banking system". B.S. thesis. Università Ca'Foscari Venezia, 2020.

[14] DeLisle Worrell. "Stressing to breaking point: Interpreting stress test results". In: (2008).