Title of your paper

Your Name

January 27, 2020

[1]:
```python
# Importing libraries
import pandas as pd
from sklearn.model_selection import train_test_split
import altair as alt
alt.data_transformers.enable('json')
#alt.renderers.enable('notebook')
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.feature_selection import RFE
from sklearn.linear_model import LogisticRegression
import matplotlib.pyplot as plt
from sklearn import preprocessing
import numpy as np
from sklearn.metrics import accuracy_score, plot_confusion_matrix, 
 ↪confusion_matrix, classification_report, roc_auc_score, roc_curve
from sklearn.metrics import recall_score, precision_score
from sklearn.model_selection import GridSearchCV
from imblearn.over_sampling import SMOTE
from docopt import docopt
from sklearn.feature_selection import RFECV
```

[2]:
```python
# Reading results
evaluation_matrix = pd.read_csv("../results/accuracies.csv")
evaluation_matrix_base = pd.read_csv("../results_baseline//accuracies.csv")
head = pd.read_csv("../results/head.csv")
summary=pd.read_csv("../results/num_describe.csv")
```

1 Table of Content:

[3]:
```
## testing varable in the markdown cell
```

[4]:
```
test_accuracy= round(evaluation_matrix.iloc[0][1])
```

0.7409333333333333

2 1. Summary

In this project we try to find the best features that best predict default customers using machine learning tools. `Logestic Regression` was found to achieve acceptable results on the test data provided to the trained model. The accuracy of the model on test data was about 0,74 and the recall on test data found to be 0.57. The precision for the model on the test was about 0.43 .The area under the ROC Curve for the final model is 0.71.

Due to the risk associated with wrongly labeled customers as non-defaul, the model was designed to reduce the false positive (false postive rate). This was also balanced with the overall accuracy on the training data. The model predict the following 7 features to be the most important features to predict customers default.

1. Amount of the given credit (NT dollar): it includes both the individual consumer credit and his/her family (supplementary) credit.
2. EDUCATION
3. MARRIAGE
4. AGE
5. Past monthly repayment status in September 2005
6. Past monthly repayment status in September 2005
7. Amount of previous payment (NT dollar) in September 2005

3 2. Introduction

Prediction of customers default behaviour is critically important in Risk Management by lenders. In particular, there has been a significant interest in identifying features that are associated with the highest prediction power to reduce the overall lender's credit risk. In this study, we perform a data-informed analysis to build a model that can sucssuflly capture features that predict default payment.

4 3. Methods

4.1 Data

We used credit default data collected from the Taiwanese market in 2005. The Data Set is available from UCI Machine Learning Repository Irvine, CA: University of California, School of Information and Computer Science. The data that contains 23 features from 30,000 customers. was originally publicized by Chung Hua University of Taiwan and Tamkang University of Taiwan. Features include :

- `LIMIT_BAL`: Amount of the given credit (NT dollar): it includes both the individual consumer credit and his/her family (supplementary) credit.
- `SEX`: Gender(1 = male; 2 = female).

- `EDUCATION`: Education (1 = graduate school; 2 = university; 3 = high school; 4 = others).
- `MARRIAGE`: Marital status (1 = married; 2 = single; 3 = others).

- `AGE`: Age (year).

- `PAY_1`, `PAY_2`, ..., `PAY_6`: Past monthly repayment status in September 2005, August 2005, ..., April 2005 respectively. (-1 = pay duly; 1 = payment delay for one month; 2 = payment delay for two months; . . .; 8 = payment delay for eight months; 9 = payment delay for nine months and above.)

- `BILL_AMT1`, `BILL_AMT2`, ..., `BILL_AMT6`: Amount of bill statement (NT dollar) in September 2005, August 2005, ..., April 2005 respectively.

- `PAY_AMT1`, `PAY_AMT2`, ..., `PAY_AMT6`: Amount of previous payment (NT dollar) in September 2005, August 2005, ..., April 2005 respectively.

4.2 Analysis

Immediately after importing the data it was split into traning and test data. Only 75% of the data was used to train the models and the test data was only used to obtain the test performance of the model on unseen data.

[5]:
```
head
```

[5]:
```
      ID  LIMIT_BAL  SEX  EDUCATION  MARRIAGE  AGE  PAY_1  PAY_2  PAY_3  \
0  27989     210000    2          2         1   30      0      0      0
1   2701      10000    1          3         2   23      0      0      0
2  18399     210000    2          2         2   23      0      0      0
3  14563     240000    2          2         1   39      4      3      2
4  11998      90000    1          2         2   34      1      2      0

   PAY_4  ...  BILL_AMT4  BILL_AMT5  BILL_AMT6  PAY_AMT1  PAY_AMT2  PAY_AMT3  \
0      0  ...      45810      42093      36587      3000      3018      2000
1      0  ...       3615       4402       5173      2000      1500       400
2      0  ...      21032      19497       3510      5000      5000      5000
3      2  ...      48905      47993      52015         0         0      4000
4     -1  ...      20172      73512      72588         0      2000     20172

   PAY_AMT4  PAY_AMT5  PAY_AMT6  DEFAULT_NEXT_MONTH
0      1500      1500      2000                   0
1      1000      1000       500                   0
2      8000      2000      4209                   0
3         0      5000      2000                   1
4     73512      3000      4000                   0

[5 rows x 25 columns]
```

Figure 1. Head of the data used in this study.

Next, we created list for numeric and categorical features, below is the summary of the traning data. It shows that that mean, standard deviation, min, max etc. The bill amount, payment amount and credit limit ranges are roughly similar which are around 800,000. It's interesting that The medians for the bill statement amounts are around 20,000, but the medians for payment amounts are 2,000. Age ranges from 21 to 75 which is reasonable.

[6]:
```
summary
```

[6]:

	Unnamed: 0	LIMIT_BAL	AGE	BILL_AMT1	BILL_AMT2
0	count	22500.000000	22500.000000	22500.00000	22500.000000
1	mean	167229.763556	35.487022	50992.89800	48905.718978
2	std	129384.485693	9.182223	73064.68632	70748.066294
3	min	10000.000000	21.000000	-165580.00000	-69777.000000
4	25%	50000.000000	28.000000	3565.75000	2928.000000
5	50%	140000.000000	34.000000	22169.00000	20859.000000
6	75%	240000.000000	41.000000	66732.75000	63104.250000
7	max	800000.000000	75.000000	746814.00000	743970.000000

	BILL_AMT3	BILL_AMT4	BILL_AMT5	BILL_AMT6	PAY_AMT1
0	22500.000000	22500.000000	22500.000000	22500.000000	22500.000000
1	46629.685644	42932.418844	39905.282444	38385.688222	5714.377733
2	68376.985307	63802.950987	60135.853082	58733.428102	17078.235838
3	-157264.000000	-170000.000000	-81334.000000	-339603.000000	0.000000
4	2577.000000	2313.000000	1711.750000	1190.000000	990.000000
5	19889.000000	18855.500000	17875.000000	16715.000000	2100.000000
6	59532.500000	53339.500000	49743.000000	48863.500000	5006.000000
7	855086.000000	616836.000000	587067.000000	568638.000000	873552.000000

	PAY_AMT2	PAY_AMT3	PAY_AMT4	PAY_AMT5	PAY_AMT6
0	2.250000e+04	22500.000000	22500.000000	22500.000000	22500.000000
1	5.848260e+03	5132.902667	4728.448311	4725.760978	5282.126533
2	2.191690e+04	16892.473653	15430.720628	15138.455175	18506.384982
3	0.000000e+00	0.000000	0.000000	0.000000	0.000000
4	8.000000e+02	390.000000	285.750000	238.000000	119.750000
5	2.001000e+03	1800.000000	1500.000000	1500.000000	1500.000000
6	5.000000e+03	4512.000000	4000.000000	4000.000000	4000.000000
7	1.227082e+06	889043.000000	621000.000000	426529.000000	528666.000000

Figure 2. Summary the data used in this study.

To learn the association between numeric features we explored their inter-correlations which can be seen below. We can observe that some features a stronger co-linearity such as BILL-AMT1,BILL-AMT2,.. to BILL-AMT6.

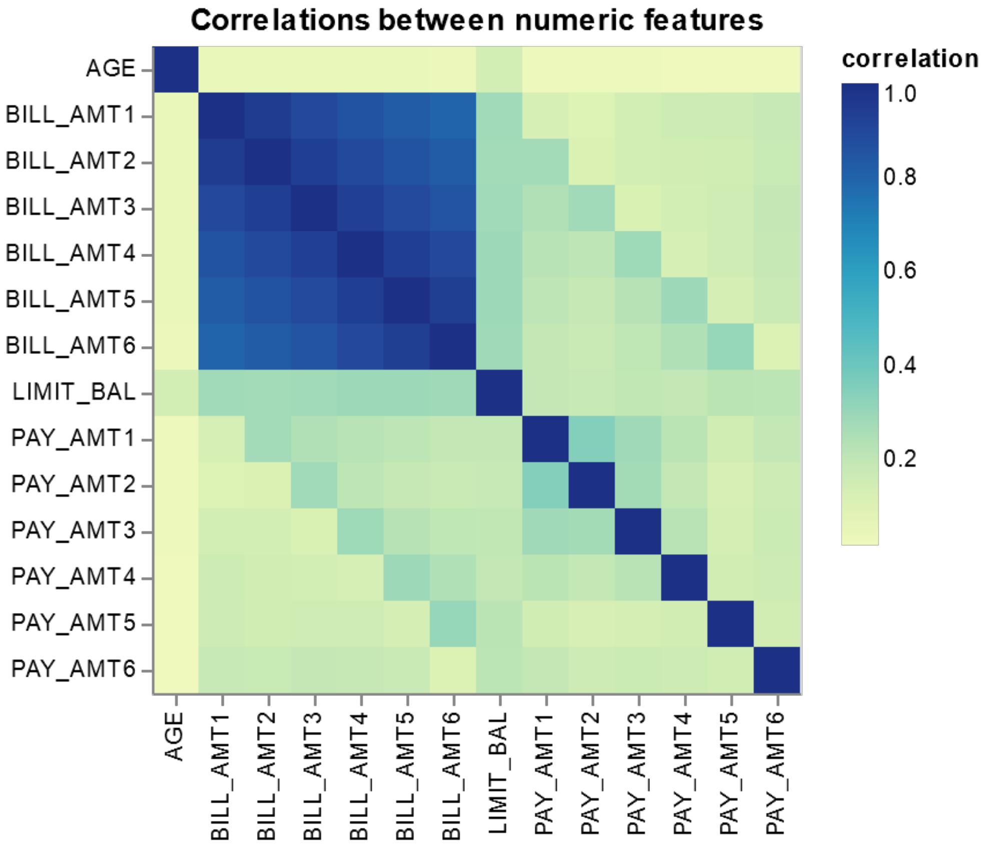

Figure 3. Inter-correlation between numeric features

We can also study the correlation between the features and the response varibale. We can see that some of the features have stronger correlation with the response varibale than others, for example LIMIT_BALANCE and Age.

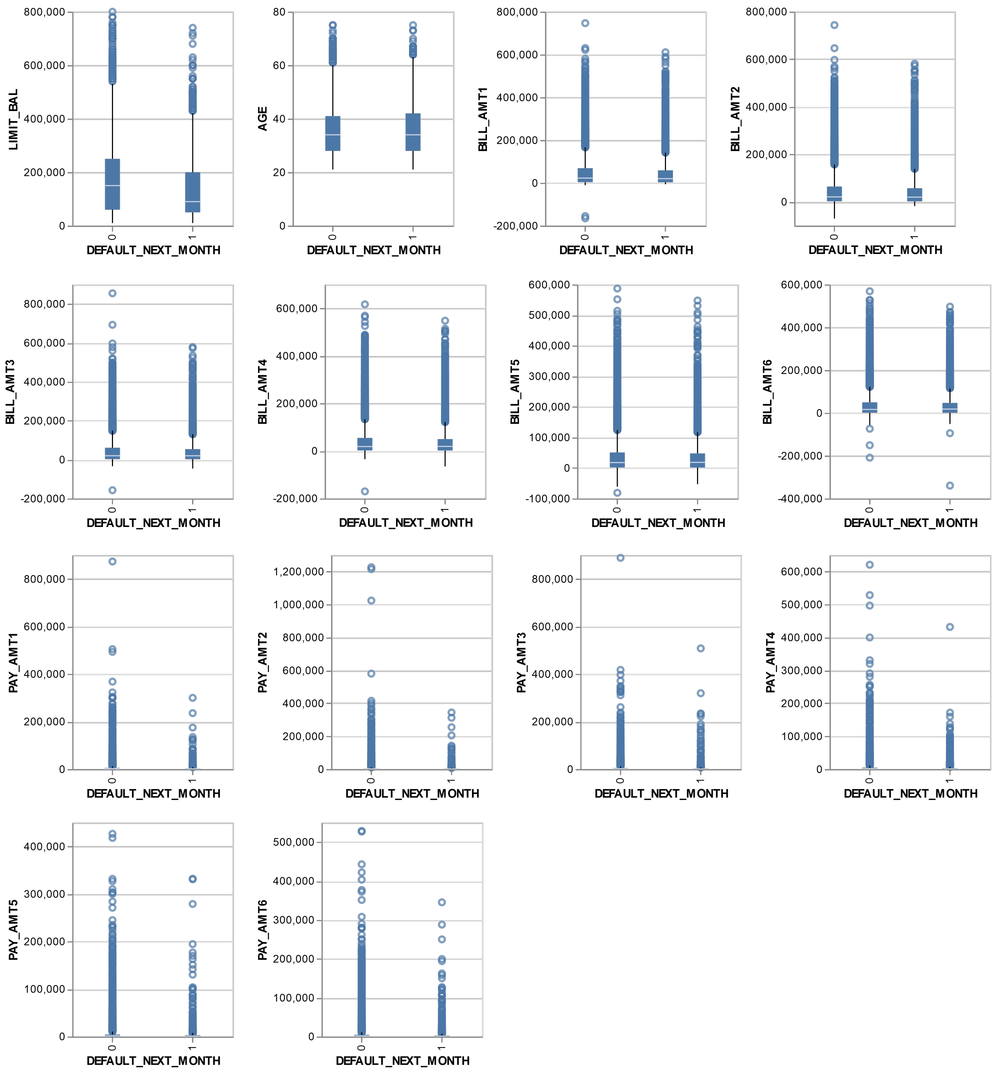

Figure 4. Correlation between numeric features and response

Figure 4 also shows that many of the features have a heavy tail distribution. To mitigate this issue we applied SMOTE (Synthetic Minority Oversampling Technique) on the response variable to create a balanced data set to fit the model. Furthermore, we implemented `RobustScaler` to scale predictors

5 4. Results

We selected logistic regression model(`LogisticRegression`) and `RFE`(recursive feature elimination) as our model since it is more robust given that the dataset has many of the features are not normally distributed. One additional advantage of (`LogisticRegression`) that is much interpretable than more complex models

We started the analysis by applying a robust scalar on the training data-set.Following that we build a model with the full set of features as our base-case model. The confusion matrix, evaluation matrix and ROC results were obtained to set the a bench-mark for for comparison purposes. `RFE` was then used to identify the most useful predictors and consequently we dropped those columns that are deemed as less useful. Eventually 7 features were used to train the model.

The hyperparameters `C` was tunned in the range from -4 to 20 using 5-fold cross-validation and the model was then fitted with the best hyperparameter. Let us now look at the result by glancing into the confusion matrix

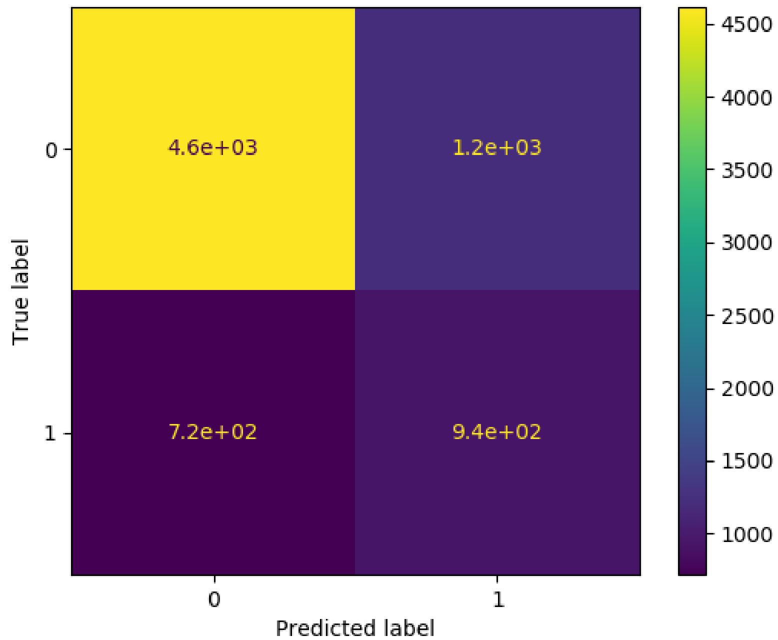

Figure 5. Confusion matrix of the fitted model with 7 features

We can see that the best model which uses 7 features tends to correctly predict the customer that defaulted out-performing the base-case model which uses all the features. This is critically important in risk management. We can see that `4600` predictions were made that correctly classified a non-default as a non-default. This is about 600 cases better than the base-case model. There was also `1200` predictions that were made that incorrectly classified a non-default as a default, actually about 700 cases worse than the best-case model.On the other hand the model was able to predict `940` cases of defaulted customers that actually defaulted which is about 160 cases worse than base-case model. The best-case model was also able to produce `720` predictions were made that incorrectly classified a defaulted customers as a defaulted customers.

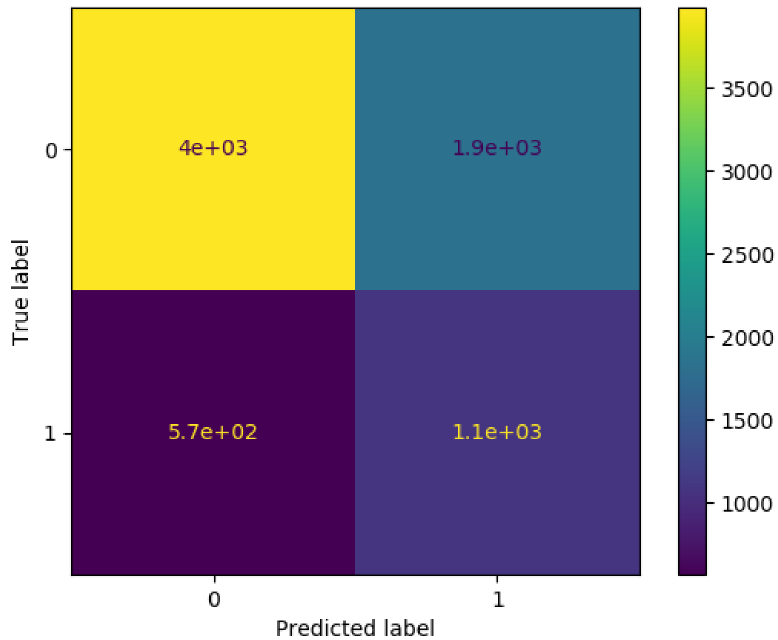

Figure 6. Confusion matrix of the fitted model with all 23 features

In terms of accuracy the results are shown below, we can see that the accuracy of the model on test data was about 0.74 and the recall on test data found to be 0.56 . The precision for the model on the test was about 0.43 .The area under the ROC Curve for the final model is 0.70.

[7]:
```
evaluation_matrix
```

[7]:

	Unnamed: 0	result
0	test accuracy	0.740933
1	train accuracy	0.741733
2	test recall	0.567372
3	test precision	0.433518
4	auc score	0.707454

This is also a good improvement over the base model which use all the 23 features as can see below. In particular we can see that the best-case model performs better in terms of test accuracy and test-precision.

[8]:
```
evaluation_matrix_base
```

[8]:

	Unnamed: 0	result
0	test accuracy	0.676267
1	train accuracy	0.677244
2	test recall	0.656798
3	test precision	0.368850
4	auc score	0.721871

ROC was plotted to to measure the model's discriminative ability. We can see that the model perform fairly good.

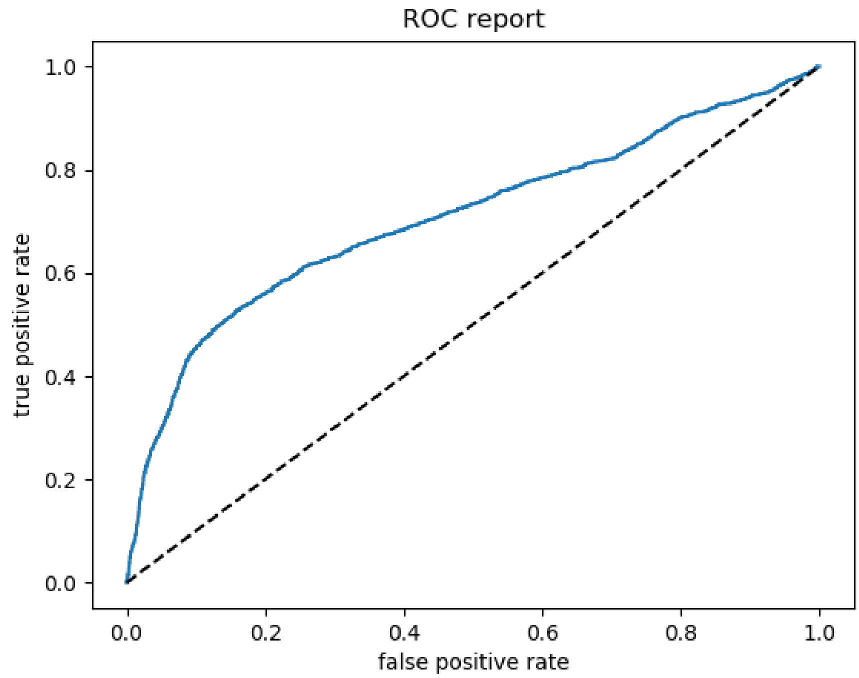

Figure 7. ROC curve for the fitted model with 7 features

6 5. Conclusions

We were able to successfully use `LogisticRegression` model to find the most important features that predict customer default. The model acheives an acceptable level of accuracy on the testing data, better tunning of hyper paramters may result a higher accuracy. Overall, we selected the best-case model to extract the most important features as it is more accurate. The precision of the best-case model is 0.43. In comparison, the base-case model only scores 0.36. While the recall of best-case model decreased from 0.656 to 0.567, AUC score only slightly dropped. Since the best-case model is more accurate, we expect the following 7 features to have the highest predictive power among all the features

1. Amount of the given credit (NT dollar): it includes both the individual consumer credit and his/her family (supplementary) credit.
2. EDUCATION
3. MARRIAGE
4. AGE
5. Past monthly repayment status in September 2005
6. Past monthly repayment status in September 2005
7. Amount of previous payment (NT dollar) in September 2005

[1] [2]

[1] Dheeru Dua and Casey Graff. UCI machine learning repository, 2017. UCI Machine Learning Repository Irvine, CA: University of California, School of Information and Computer Science

[2] Guido Van Rossum and Fred L. Drake. Python 3 Reference Manual. CreateSpace, Scotts Valley, CA, 2009

[3] Wickham, H. 2017. tidyverse: Easily Install and Load the 'Tidyverse'. R package version 1.2.1. https://CRAN.R-project.org/package=tidyverse

[4] Wickham H (2011). "testthat: Get Started with Testing." The R Journal, 3, 5–10. https://journal.r-project.org/archive/2011-1/RJournal_2011-1_Wickham.pdf.

[5] McKinney, W. (2012). Python for data analysis: Data wrangling with Pandas, NumPy, and IPython. " O'Reilly Media, Inc.".

[6] Nielsen, F. Å. (2014). Python programming—Scripting.

[7] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., … & Vanderplas, J. (2011). Scikit-learn: Machine learning in Python. Journal of machine learning research, 12(Oct), 2825-2830.

[8] VanderPlas, J., Granger, B., Heer, J., Moritz, D., Wongsuphasawat, K., Satyanarayan, A., … & Sievert, S. (2018). Altair: Interactive statistical visualizations for python. Journal of open source software, 3(32), 1057.

[9] Percival, H. (2014). Test-driven development with Python: obey the testing goat: using Django, Selenium, and JavaScript. " O'Reilly Media, Inc.".

[10] Lemaître, G., Nogueira, F., & Aridas, C. K. (2017). Imbalanced-learn: A python toolbox to tackle the curse of imbalanced datasets in machine learning. The Journal of Machine Learning Research, 18(1), 559-563.

[?] **[?]**

References

[1] Eric Jones, Travis Oliphant, Pearu Peterson, et al. SciPy: Open source scientific tools for Python, 2001–. [Online; accessed Date].

[2] Wes McKinney et al. Data structures for statistical computing in python. In *Proceedings of the 9th Python in Science Conference*, volume 445, pages 51–56. Austin, TX, 2010.