PREDICTING ACADEMIC SUCCESS USING DECISION TREES

Paulina Ocampo Duque	Maria José Gutiérrez	Miguel Correa	Mauricio Toro
Universidad Eafit	Universidad Eafit	Universidad Eafit	Universidad Eafit
Colombia	Colombia	Colombia	Colombia
mpocampod@eafit.edu.co	mjgutierre@eafit.edu.co	macorream@eafit.edu.co	mtorobe@eafit.edu.co

ABSTRACT

The objective of this project is to predict academic success using decision trees, this problem came up because there are many studies of academic desertion but a few of academic success, for that reason what we are looking for is to predict the total score of the students in the Saber Pro test, define if students are above average or not and know which factors influence in their results using an algorithm based on decision trees.

This study is important because it will enable to identify the students who are likely to fail and allow the teachers to provide an extra way to teach, focusing on those students and prevent them from failure. Similar to studies in India, with the difference that they consider the student´s academic performance along the years to determinate their success.

To give a solution, we proposed a decision tree using a data structure (CART algorithm) which determines if a person will have a good grade in a test. A decision tree algorithm learns from the data it analyzes and determines the probability of success or failure of the student more ease each time you pass it new data.

Keywords

Decision trees, machine learning, academic success, standardized student scores, test-score prediction.

1. INTRODUCTION

An important element in our actual society like the technology is, could be a helpful tool in education in Latin America. In this project we are going to focus on calculating the academic success the students will obtain with a total score higher than the average for his cohort, in the Saber Pro tests, taking data from different social factors and the total score of the Saber 11.

Using the digital transformation Education 4.0, successful algorithms have been used to predict academic dropout with influencing factors, however it has not been possible to predict academic success in superior education.

1.1. Problem

The problem is based on creating an algorithm using data of the test Saber 11, processing decision trees to predict if a student will get a good score in the Saber Pro test, also considering some academic and sociodemographic variables provided.

1.2 Solution

In this work, we focused on decision trees to give a solution about the results predictions in exams, because they provide great explainability, order and the most important thing is that decision trees let the person compare future statistics with specific information, for example punctuation, studies

advance which students are likely to fail, the colleges or the teachers can take the necessary actions to improve the results. and subjects. We avoid black-box methods such as neural networks, support-vector machines and random forests because they lack explainability also, they aren´t efficient and effective for these cases, like decision trees.

In this case, we want to use CART algorithm because handles both categorical and continuous attributes through Gini index to build a decision tree. Also, because CART algorithm produces binary splits making predictions straightforward.

1.2 Article structure

In what follows, in Section 2, we present related work to the problem. Later, in Section 3 we present the datasets and methods used in this research. In Section 4, we present the algorithm design. After, in Section 5, we present the results. Finally, in Section 6, we discuss the results and we propose some future work directions.

2. RELATED WORK

2.1 A Decision Tree Algorithm Pertaining to the Student Performance Analysis and Prediction

The analysis related to the prediction of student´s academic performance in higher education seems an essential requirement for the improvement in quality education.

For that reason, Mrinal Pandey and Vivek Kumar Sharma[1] did a study considered the academic performance of some students from high school to the prefinal semester of Engineering and then predict with a model, the final results for the completing the graduate degree in engineering. The Data set for the study has been collected from Manav Rachna College of engineering district Faridabad of Haryana state.

For model construction, C4.5 decision tree method has been used, which is based on gain ratio as attribute selection measure. The attribute having maximum gain ratio value is selected for splitting the node. This process continues till the complete tree is constructed. WEKA tool kit was used to select the attributes and construct the J48 decision tree algorithm, which is a java version of C4.5

The model obtained accuracy of 80.15% and 82.58% in 10-fold cross validation method and percentage method respectively. It indicates that model is good for forecasting the grades of students. This model helps to the management to identify weak students and can take appropriate decision to prevent them from failure.

2.2 Performance Prediction of Engineering Students using Decision Trees

As the number of engineering seats and colleges are increasing, the inferior students are also enrolled in engineering courses. So, the results of the universities for engineering courses are going down. If we know in

Good placement is one of the key factors that will help the college to attract students.

The data is collected from S. G. R. Education Foundation's College of Engineering and Management. Data of 346 students of the institute is collected who appeared for the first year of engineering in the year 2009-10, 2010-11 and was collected through the enrolment form filled by the student at the time of admission.
From this data, student.arff file was created. This file was loaded into WEKA explorer. The WEKA workbench contains a collection of visualization tools and algorithms for data analysis and predictive modeling, together with graphical user interfaces for easy access to this functionality. The algorithm used for classification is J48, a java implementation of C4.5 algorithm.
The decision tree generated from student.arff with three class prediction (pass, allowed to keep terms and fail) had an accuracy of 60.46 %. That is out of 346 instances 209 instances are correctly classified.[2]

1. **Predicting students' performance using id3 and c4.5 classification algorithms**

In this project, they have analyzed the data of students in first year of engineering. This data provided includes their full name, gender, application ID, scores in entrance examinations, category and admission type. Then it was applied the ID3 and C4.5 algorithms to predict the results of these students in their first semester as precisely as possible.

In the first stage, information about students who have been admitted to the second year was collected. In the second stage, the relevant information was fed into a database. The third stage involved applying the ID3 and C4.5 algorithms on the training data to obtain decision trees and after that test the data. These stages of implementation are in Figure 1

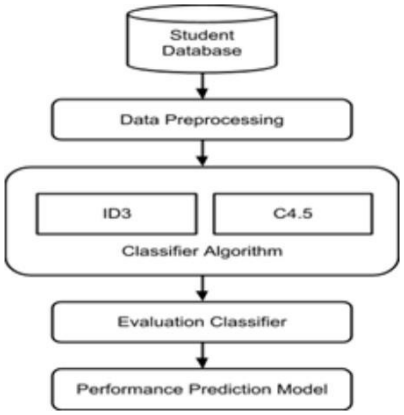


Figure 1. Processing model

Took image from:
https://arxiv.org/ftp/arxiv/papers/1310/1310.2071.pdf

They realized that the tree obtained from c4.5 algorithm had fewer nodes compared to the ID3,[6] so these newly learnt predictive patterns for predicting students were implemented in a working web application for staff members to use it.

Data Mining: A Prediction for Performance Improvement of Engineering Students using Classification

Educational data mining is used to study the data available in the educational field and bring out the hidden knowledge from it. Classification methods like decision trees can be applied on the educational data for predicting the student's performance and help to identify the weak students and help them to score better marks.

The **C4.5, ID3** and **CART** decision tree algorithms are applied on engineering student's data to predict their performance in the final exam predicting the number of students who are likely to pass, fail or promoted to next year. The marks obtained by the students are fed into the system and the results were analyzed for the next session.

The project shows us and explain us that decision trees are so popular because they produce classification rules that are easy to interpret than other classification methods. Machine learning algorithms such as the C4.5 decision tree algorithm can learn effective predictive models from the student data accumulated from the previous years. The empirical results show that we can produce short but accurate prediction list for the student by applying the predictive models to the records of incoming new students. This study will also work to identify those students which needed special.

3. MATERIALS AND METHODS

In this section, we explain how the data was collected and processed and, after, different solution alternatives considered to choose a decision-tree algorithm.

3.1 Data Collection and Processing

We collected data from the *Colombian Institute for the Promotion of Higher Education* (ICFES), which is available online at ftp.icfes.gov.co. Such data includes anonymized Saber 11 and Saber Pro results. Saber 11 scores of all Colombian high schools graduated from 2008 to 2014 and Saber Pro scores of all Colombian bachelor-degree graduates from 2012 to 2018 were obtained. There were 864,000 records for Saber 11 and records 430,000 for Saber Pro. Both Saber 11 and Saber Pro, included, not only the scores but also socio-economic data from the students, gathered by ICFES, before the test.

In the next step, both datasets were merged using the unique identifier assigned to each student. Therefore, a new dataset that included students that made both standardized tests was created. The size of this new dataset is 212,010 students. After, the binary predictor variable was defined as follows: Does the student score in Saber Pro is higher than the national average of the period?
It was found out that the datasets were not balanced. There were 95,741 students above average and 101,332 students

below average. We performed under sampling to balance the dataset to a 50%-50% ratio. After under sampling, the final dataset had 191,412 students.

Finally, to analyze the efficiency and learning rates of our implementation, we randomly created subsets of the main dataset, as shown in Table 1. The dataset was divided into 70% for training and 30% for testing. Datasets are available at https://github.com/mauriciotoro/ST0245-Eafit/tree/master/proyecto/datasets .

	Dataset 1	Dataset 2	Dataset 3	Dataset 4	Dataset 5
Train	15,000	45,000	75,000	105,000	135,000
Test	5,000	15,000	25,000	35,000	45,000

Table 1. Number of students in each dataset used for training and testing.

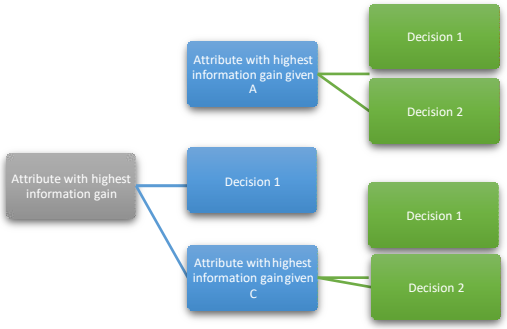
3.2 Decision-tree algorithm alternatives

The process of discovering or extracting new patterns from large data use techniques like Classification and prediction to make out important data classes and predict probable trend. The Decision Tree is an important classification method in data mining classification and induction research. These algorithms have the merits of high classifying speed, strong learning ability and simple construction.
In what follows, we present different algorithms to solve to automatically build a binary decision tree.

3.2.1 ID3 (Iterative Dichotomiser 3)

The basic idea of ID3 algorithm is to construct the decision tree by employing a top-down, greedy search through the given sets to test each attribute at every tree node. To find an optimal way to classify a learning set, what we need to do is to minimize the questions asked.

We can select the attribute with the highest information gain as the test attribute of current node, the information needed to classify the training sample subset obtained will be the smallest. That is to say, the use of this property to partition the sample set contained in current node will make all generated sample subsets reduce to a minimum.[4]



Took image from: https://www.google.com/url?sa=i&url=https%3A%2F%2Fen.wikipedia.org%2Fwiki%2FID3_algorithm&psig=AOvVaw01zQLpcZV4GPGud8BCn6m&ust=1597616547871000&source=images&cd=vfe&ved=0CAIQjRxqFwoTCNDCmcinusCFQAAAAAdAAAAABAD

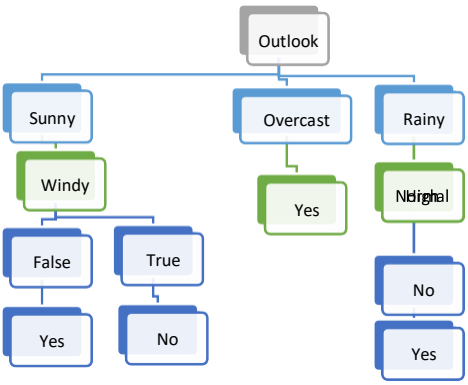
Example:

- Play Tennis

The symbolic attribute description:

Attribute	Possible values
Outlook	Sunny,overcast,rain
Temperature	Hot,mild, cool
Humidity	High, normal
Windy	True,False

Decision Tree



3.2.2 C5.0

C5.0 algorithm is an extension of C4.5 algorithm which is also extension of ID3. It is better than C4.5 on the speed, memory and the efficiency, giving more accurate and efficient results. This model works by splitting the sample based on the field that provides the maximum information gain. The classification process generates fewer rules compare to other techniques, so the proposed system has low memory usage.[5]

C5 algorithm has many features like:

- In classification technique the C5 classifier can anticipate which attributes are relevant and which are not relevant in classification.
- Error rate is low so accuracy in result set is high
- The large decision tree can be viewing as a set of rules which is easy to understand.
- The memory usage is minimum, and it also improve the accuracy.

3.2.3 CART (Classification and Regression Trees)

CART is a term introduced by Leo Breirman to refer to Decision Tree algorithms that can be used for classification or regression predictive modeling problems.

The CART algorithm provides a foundation for important algorithms like bagged decision trees, random forest and boosted decision trees.

The representation for the CART model is a binary tree which can be stored to file as a graph or a set of rules. With this model making predictions is relatively straightforward.

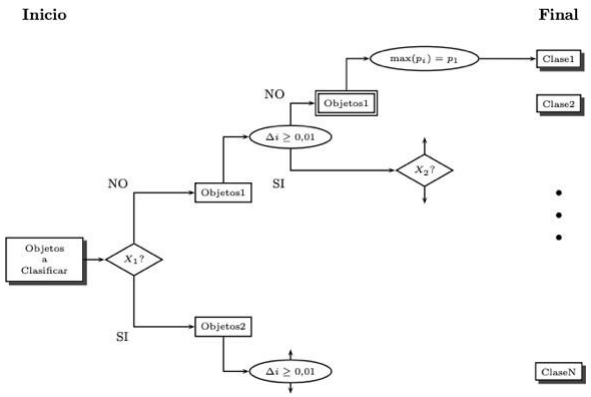
A learned binary tree is actually a partitioning of the input space. You can think of each input variable as a dimension on a p-dimensional space. The decision tree split this up into rectangles (when p=2 input variables) or some kind of hyper-rectangles with more inputs.

New data is filtered through the tree and lands in one of the rectangles and the output value for that rectangle is the prediction made by the model. This gives you some feeling for the type of decisions that a CART model is capable of making, e.g. boxy decision boundaries.

Creating a CART model involves selecting input variables and split points on those variables until a suitable tree is constructed.

The selection of which input variable to use and the specific split or cut-point is chosen using a greedy algorithm to minimize a cost function. Tree construction ends using a predefined stopping criterion, such as a minimum number of training instances assigned to each leaf node of the tree.

Example



Took image from: http://www.bdigital.unal.edu.co/671/1/42694070_2009.pdf

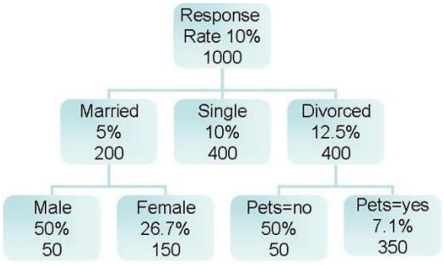
3.2.4 CHAID (CHI-SQUARE AUTOMATIC INTERACTION DETECTOR)

CHAID was a technique created by Gordon V. Kass in 1980. CHAID is a tool used to discover the relationship between variables. CHAID analysis builds a predictive model, or tree, to help determine how variables best merge to explain the outcome in the given dependent variable. In CHAID analysis, nominal, ordinal, and continuous data can be used, where continuous predictors are split into categories with approximately equal number of observations. CHAID creates all possible cross tabulations for each categorical predictor until the best outcome is achieved and no further splitting can be performed. In the CHAID technique, we can visually see the relationships between the split variables and the associated related factor within the tree. The development of the decision, or classification tree, starts with identifying the target variable or dependent variable, which would be considered the root. [9]

Decision tree components in CHAID analysis:

- **Root node:** Root node contains the dependent, or target, variable.
- **Parent's node:** The algorithm splits the target variable into two or more categories.
- **Child node:** Independent variable categories which come below the parent's categories in the CHAID analysis tree are called the child node.
- **Terminal node:** The last and less important categories of the CHAID analysis tree.

Example



Took image from: http://www.dmstat1.com/res/MarketSegmentationWithCH AID.html

4. ALGORITHM DESIGN AND IMPLEMENTATION

In this work we use the CART algorithm, which is implemented by a binary tree, because is capable of divide data into a node based on the value of a variable, allows you to decide when a branch is terminal and can no longer be split and also makes a prediction for the target variable on each terminal node.

This algorithm uses a Gini index as an impurity measure to select the attribute, and then the attribute with the greatest reduction of impurities is used to split the node records.

4.1 Data Structure

A binary decision tree technique consists of a hierarchical and sequential division of the problem in which each of these divisions or nodes graphically describes the possible decisions and therefore the results of the different combinations of decisions and events. Each event is assigned probabilities and each of the branches is determined an outcome. [10]

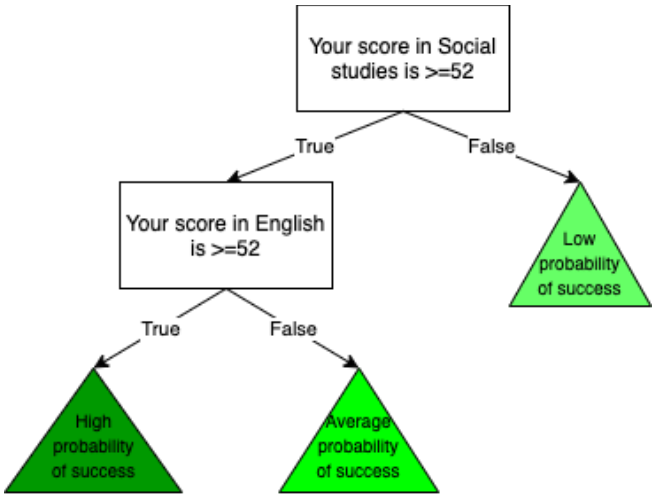


Figure 1: decision nodes indicates a decision to be made, represented by a square. Each branch indicates a possible outcome or action, represented by lines. Probabilistic nodes show multiple uncertain outcomes, represented with triangles.

4.2 Algorithms

Creating a CART model algorithm involves selecting input variables and split points on those variables until a suitable tree is constructed.

The selection of which input variable to use and the specific split or cut-point is chosen using an algorithm to minimize a cost function. Tree construction ends using a predefined stopping criterion, such as a minimum number of training instances assigned to each leaf node of the tree.

4.2.1 Training the model

The builidng of the binary decision tree based on the algorithm starts at the root node, which includes the entire population in the learning dataset. Starting with this node, the CART algorithm finds the best possible variable to split the node into two child nodes. To find the best variable, the software checks all possible split variables.

This is process stops when:

- There is only one observation on each of the child nodes.
- All observations within each child node have the identical distribution of predictor variables, making splitting impossible.
- The user has set an external limit on the number of levels in the maximum tree. [11]

To generate simplest secuence of trees, the pruning method is used, which is based on a complexity parameter.
First, a branch of the t-node of a T tree is made up of it and all its descendants, then pruning the branch in t consists of removing all descendants from node t and finally, weaker branches are removed with error criteria and tree complexity.
The maximum tree will always fit the learning dataset more accurately than any other tree. The performance of the maximum tree in the original learning dataset, generally greatly overestimates the tree's performance in a separate set of data obtained from a similar population. [12]

4.2.2 Testing algorithm

The algorithm sorts the new data doing the same process as in the beginning, putting the data in the tree, going down from the root node to a leaf.

4.3 Complexity analysis of the algorithms

First, M columns were walked through to make all the calculations necessary to find the condition that best divided the tree. Then, N rows were walked to evaluate all possible
values taken by each column. And finally, to
calculate the Gini index with each condition tested, they had to go through N rows, in the worst case, to find the best.
the
N2*M terms that appear in the complexity calculation in time is because these operations were nested.
The term 2M in the worst case is because we use a binary tree which the subdivisions depend on the number of columns that the tree had.

We realized the complexity of tree validation
in a way that the conditions cross the N rows and M columns once, leaning of the decision tree that was already created, to know which would be the result of each individuals present in the data of validation.

Algorithm	Time Complexity
Train the decision tree	$O(N^2*M^2*2^M)$
Test the decision tree	O(N*M)

Table 2: Time Complexity of the training and testing algorithms. Where N represents the number or rows, and M represents the columns.

The complexity in memory, at the time of training model turned out to be N*M, since in the worst of cases a pair of matrixes were created from each node of the size
unified from the original matrix. This for each node, which finally takes the number of 2M,
because it is a binary tree with M subdivisions in the worst-case scenario.
The complexity in memory for the validation of the model is a constant complexity, since at the moment of validate is already created the whole binary tree and the elements necessary for the validation or obtaining of
results from new entries.

Algorithm	Memory Complexity
Train the decision tree	$O(N*M*2^M)$
Test the decision tree	O(1)

Table 3: Memory Complexity of the training and testing algorithms. Where N represents the number or rows, and M represents the columns.

4.4 Design criteria of the algorithm

We designed the CART algorithm with the Gini impurity measuring the degree of impurity between nodes to make a binary tree, because this algorithm is the best in terms of classification due to offers a good complexity in time O(n). Also, because missing values don´t influence in the model making the tree more accurate and easier to understand

5. RESULTS

5.1 Model evaluation

In this section, we present some metrics to evaluate the model. Accuracy is the ratio of number of correct predictions to the total number of input samples. Precision is the ratio of successful students identified correctly by the model to successful students identified by the model. Finally, Recall is the ratio of successful students identified correctly by the model to successful students in the dataset.

5.1.1 Evaluation on training datasets

In what follows, we present the evaluation metrics for the training datasets in Table 3.

	Dataset 1	***Dataset 2***	***...Dataset n***
Accuracy	0.78	0.75	0.78
Precision	0.76	0.72	0.72
Recall	0.79	0.82	0.8

Table 3. Model evaluation on the training datasets.

5.1.2 Evaluation on test datasets

In what follows, we present the evaluation metrics for the test datasets in Table 4.

	Dataset 1	***Dataset 2***	***...Dataset 3***
Accuracy	0.77	0.78	0.78
Precision	0.76	0.72	0.72
Recall	0.78	0.81	0.81

Table 4. Model evaluation on the test datasets.

5.2 Execution times

	Dataset 1	***Dataset 2***	***...Dataset n***
Training time	36 s	180 s	350 s
Testing time	7 s	55 s	102 s

Table 5: Execution time of the *CART* algorithm for different datasets.

5.3 Memory consumption

We present memory consumption of the binary decision tree, for different datasets, in Table 6.

	Dataset 1	***Dataset 2***	***...Dataset n***
Memory consumption	66 MB	320 MB	447 MB

Table 6: Memory consumption of the binary decision tree for different datasets.

6. DISCUSSION OF THE RESULTS

According with the table number 3 and 4, we can say that the precision, recall and accuracy are in the percentage to what was expected so it is appropriate for this problem. We can improve our memory consumption because the results were a little high, time consumption was fast considering the amount of data but also in the future we can improve it.
This algorithm will help to identify the students that won´t have a good result in the saber Pro test.

6.1 Future work

In the future we want to make our code more optimal and recursive, and with that the complexity in terms of time and memory will be better too, because now there are some repeated or useless process.

ACKNOWLEDGEMENTS

We want to thank Simón Marín, Computer Science and Engineering student at EAFIT University and Data Structures and Algorithms instructor who explained and supported us with the concepts related with the project and the course.

REFERENCES

[1] Pandey, M. and Kumar Sharma, V. A Decision Tree Algorithm Pertaining to the Student Performance Analysis and Prediction. *International Journal of Computer Applications 61*, 13 (2013), 1-5.

[2] Kabra, R. R., and Bichkar, R. S. Performance prediction of engineering students using decision trees."*International Journal of computer applications* 36,11 (2011),8-12.

[3] Jin, C., De-Lin, L., & Fen-Xiang, M. (2009, July). An improved ID3 decision tree algorithm. In 2009 4th International Conference on Computer Science & Education (pp. 127-130). IEEE.

[4] Brijain, M., Patel, R., Kushik, M., & Rana, K. (2014). A survey on decision tree algorithm for classification.

[5] Pandya, R., & Pandya, J. (2015). C5. 0 algorithm to improved decision tree with feature selection and reduced error pruning. International Journal of Computer Applications, 117(16), 18-21.

[6] Adhatrao, K., Gaykar, A., Dhawan, A., Jha, R., & Honrao, V. (2013). Predicting students' performance

using ID3 and C4. 5 classification algorithms. arXiv preprint arXiv:1310.2071.
[7] Yadav, S. K., & Pal, S. (2012). Data mining: A prediction for performance improvement of engineering students using classification. arXiv preprint arXiv:1203.3832.
[8] Brownlee, J. Classification and Regression Trees for Machine Learning. *Machine Learning Mastery*, 2016. https://machinelearningmastery.com/classification-and-regression-trees-for-machine-learning/.
[9] CHAID - Statistics Solutions. *Statistics Solutions*, 2020. https://www.statisticssolutions.com/non-parametric-analysis-chaid/#:~:text=CHAID%20analysis%20splits%20the%20target,data%20to%20be%20normally%20distributed.
[10] D. & T. Kotsiantis, Sotiris & Koumanakos, E & Tzelepis, "Forecasting Fraudulent Financial Statements using Data Mining," vol. 1, no. 12, pp. 844–849, 2007
[11] Roger J. Lewis, M. P. (2000). An Introduction to Classification and Regression Tree (CART) Analysis. San Francisco,California.
[12] Roman Timofeev, D. W. (2004). Classification and Regression Trees (CART) Theory and Applications. Berlin: Humboldt University, Berlin.