

Projekt-Bericht und Dokumentation im Kurs
Ausgewählte Kapitel sozialer Webtechnologien (SoSe15)
Machine Learning

Claus Holland Richard Remus

25. Juli 2015

Contents

1	Einführung	3
2	Explorative Datenanalyse	5
3	Data Wrangling	6
4	Erstellung der Modelle	7
4.1	Logistic Regression	7
4.1.1	Cross Validation	7
4.2	Support Vector Machine	7
4.2.1	Grid Search	8
4.3	Random Forest	8
5	Performancevergleich der Modelle	8
6	Ergebnisse	8
7	Fazit	8

1 Einführung

Als wichtiger Teil des Wirtschaftslebens in einer Marktwirtschaft ist es die Aufgabe von Banken Kredite zu vergeben. Unternehmen können so zum Beispiel Investitionen finanzieren, wachsen und mit den hoffentlich erzielten Gewinnen ihren Kredit tilgen und wieder zurückzahlen. Doch auch Privatbürger nehmen Kredite auf, um sich etwa Wohnmobilen zu kaufen oder andere Anschaffungen zu tätigen. Wohneigentum kann wiederum mit Hypotheken belastet werden, dem Abtreten von Rechten an einer Immobilie im Gegenzug für ein Darlehen. Unternehmen haben in der Regel Kapital in Form von Anlagen oder Ähnlichem. Gerade wenn eine Privatperson aber nicht mit viel Besitz für einen Kredit haften kann, stehen Banken vor einer schwierigen Aufgabe. Sie müssen bei Abschluss eines Kreditvertrages entscheiden, wie zuverlässig der Kreditnehmer in der Zukunft sein wird. Auf Basis dieser Entscheidung bestimmen sie die Konditionen des Kredites oder entscheiden sich bei einem womöglich sehr hohen Ausfallrisiko sogar gegen eine Kreditvergabe. Damit entgeht ihnen aber womöglich ein einträgliches Geschäft. Mit den Methoden des Maschinellen Lernens stehen Werkzeuge zur Verfügung, um mit beschränktem Aufwand verhältnismäßig sichere Aussagen zu treffen, die einen potenziellen Bankkunden anhand bestimmter Eigenschaften, sogenannter Features, hinsichtlich seines Kreditzahlungs-Ausfallrisikos beurteilen helfen. Dies soll im Rahmen dieser Arbeit aufgezeigt werden. Es liegen uns 250.000 Datensätze von Bankkunden vor. Diese umfassen neben verschiedenen finanziellen und demografischen Angaben auch Werte zum Kredit-Ausfallverhalten. Es handelt sich somit um Trainingsdaten im Sinne des Maschinellen Lernens. Unsere Aufgabe besteht daher darin, Klassifikatoren zu implementieren, die anhand der gegebenen Werte am zuverlässigsten, also mit einer möglichst hohen Trefferquote, das Ausfallrisiko für die Datensätze korrekt vorhersagen. Dafür analysieren wir den Datensatz erst einmal hinsichtlich bestimmter Auffälligkeiten und auch bezüglich der vorhandenen Datenqualität, führen also die sogenannte explorative Datenanalyse durch. Aus ihr ergeben sich idealerweise bereits erste Erkenntnisse, welche der Features besser als andere geeignet erschienen, um mit dem Ergebnis “Guter Kreditnehmer” oder “Risikoreicher Kreditnehmer” zu korrelieren. Anschließend wird auf Basis des ersten Punkts von uns der Datensatz optimiert. Ermittelte Schwachstellen werden wenn möglich kompensiert und Skalierungen so gewählt, dass bildliche Darstellungen, die sogenannten Plots eine möglichst hohe Aussagekraft beinhalten. Darauf folgt die Klassifikationsphase. In ihr werden wir geeignete Klassifikationsalgorithmen ermitteln und zur Anwendung bringen. Ergebnis soll ein Klassifikator sein, der den gewünschten Anforderungen, also einer hohen Trefferquote bei den Vorhersagen entspricht. Eine Evaluation und ein Fazit schließen sich an. Der Datensatz entstammt der Web-Plattform “www.kaggle.com” und dort dem bereits

beendeten Wettbewerb “Give me some Credit” aus dem Jahr 2011. Ziel dieses Belegs ist es allerdings nicht, die Wettbewerbskriterien auf kaggle zu erfüllen. Uns geht es darum, die Mächtigkeit der Werkzeuge des Machinellen Lernens zu demonstrieren, um realwirtschaftliche Probleme zu erfassen. Da uns auf kaggle keine geeigneten Testdaten zur Verfügung stehen, in denen ebenfalls das Kreditausfallrisiko in einer Form gegeben ist, dass die erfolgreiche Anwendung eines geeigneten Klassifikators bewiesen werden kann, nutzen wir stattdessen randomisierte Datensätze aus den Trainingsdaten zur Verifizierung unserer Arbeit.

2 Explorative Datenanalyse

Die Zielklasse ist *SeriousDlqin2yrs*, welche Aussage darüber trifft, ob der Kreditnehmer innerhalb zweier Jahre nach Vertragsschluss in ernsthafte Zahlungsschwierigkeiten kam. Zunächst betrachten wir die Verteilung dieser Klasse auf den Trainingsdaten.

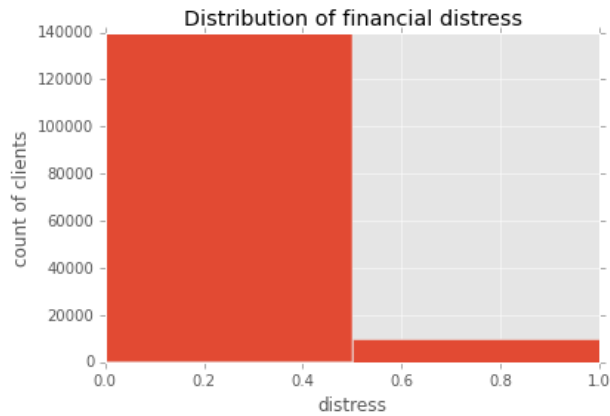


Figure 1: Verteilung von SeriousDlqin2yrs

3 Data Wrangling

Die Beschaffenheit des Datensatzes hat uns vor ein paar Probleme gestellt. Zunächst lag für die als *cs-test.csv* betitelten Daten keine Klassifikation vor, lediglich eine Beispieleinreichung mit Wahrscheinlichkeiten (*sampleEntry.csv*). Die Daten aus *cs-test.csv* sollten also zur Erstellung einer Einreichung für die Leaderboards dienen. Dementsprechend mussten wir für Training und Validierung die 150000 Datensätze aus *cs-training.csv* verwenden.

Von diesen 150000 Datensätzen verfügten 29731 über NaN-Einträge. Für die Behandlung der betroffenen Zeilen haben wir zwei verschiedene Herangehensweisen gewählt:

- Data Cleaning: die NaN-Werte wurden durch den Spaltenmittelwert ersetzt. Dadurch soll verhindert werden, dass die fehlerhaften Zeilen keine starken Strömungen in den jeweiligen Kategorien geben.
- Data Cropping: Die Zeilen mit NaN-Werten wurden gelöscht.

Darüber hinaus haben wir unrealistische oder fehlerhaft erscheinende Extremwerte (Outliers) entfernt. Der erlaubte Wertebereich wurde hierbei durch ein Vielfaches der Standardabweichung bestimmt. Der jeweilige Faktor wurde für jede Klasse händisch ermittelt.

Aufgrund des starken Ungleichgewichts in der Zielklasse, haben wir ausserdem Oversampling und Undersampling angewendet. Beide Verfahren dienen dazu, einen möglichst ausgeglichenen Trainingsdatensatz zu erhalten.

- Undersampling: Aus dem Datensatz werden zufällig aus beiden Klassifikationen gleichviele Daten gewählt und als Trainingsdaten verwendet. Der Undersampling-Trainingsdatensatz hatte dementsprechend geringen Umfang.
- Oversampling: Es werden sooft Daten der unterrepräsentierten Klasse kopiert und angefügt, bis ein Gleichgewicht eintritt. Der Oversampling-Datensatz war also in unserem Fall geringfügig größer.

Für das Over-/Undersampling haben wir die bereits im Data Cleaning behandelten Daten gewählt und einen möglichen großen Satz an Trainingsdaten zu erhalten.

4 Erstellung der Modelle

Im Folgenden beschreiben wir die Modelle, welche wir für diese Aufgabe entwickelt haben. Die Ergebnisse wurden gleichermaßen nach Genauigkeit und ROC-AUC-Score bewertet.

4.1 Logistic Regression

Zunächst haben wir eine einfache Logistische Regression angewendet.

Trainingsdaten	Genauigkeit	Area-Under-Curve
Cleaned	0,9332	0.69885151 (+/-0.00110)
Cropped	0,9332	0,6589
Undersampled	0,9332	0,6589
Oversampled	0,9332	0,6589

4.1.1 Cross Validation

Trainingsdaten	Genauigkeit	Area-Under-Curve
Cleaned	0,9338	0,6589
Cropped	0,9332	0,6589
Undersampled	0,9332	0,6589
Oversampled	0,9332	0,6589

4.2 Support Vector Machine

Das Modellieren mit der Support Vector Machine, in unserem Falle ein *Support Vector Classifier*, war äußerst zeitaufwändig. Zudem bietet die SVM viel Spielraum für die Wahl der Hyperparameter, wie dem *Kernel*, dem Penalty-Parameter C und dem Kernel-Koeffizienten γ . Wir haben zunächst die Standardparameter von SciKitLearn beibehalten. Ausserdem mussten wir eine Feature-Skalierung durchführen, da die SVM in aus SciKitLearn einen zentrierten Datensatz erwartet. Dies ließ sich aber mit dem zur Verfügung gestellten `StandardScaler` leicht bewerkstelligen.

4.2.1 Grid Search

Trainingsdaten	Genauigkeit	Area-Under-Curve
Cleaned	0,9332	0.69885151 (+/-0.00110)
Cropped	0,9332	0,6589
Undersampled	0,9332	0,6589
Oversampled	0,9332	0,6589

4.3 Random Forest

Trainingsdaten	Genauigkeit	Area-Under-Curve
Cleaned	0,9332	0.69885151 (+/-0.00110)
Cropped	0,9332	0,6589
Undersampled	0,9332	0,6589
Oversampled	0,9332	0,6589

5 Performancevergleich der Modelle

6 Ergebnisse

7 Fazit