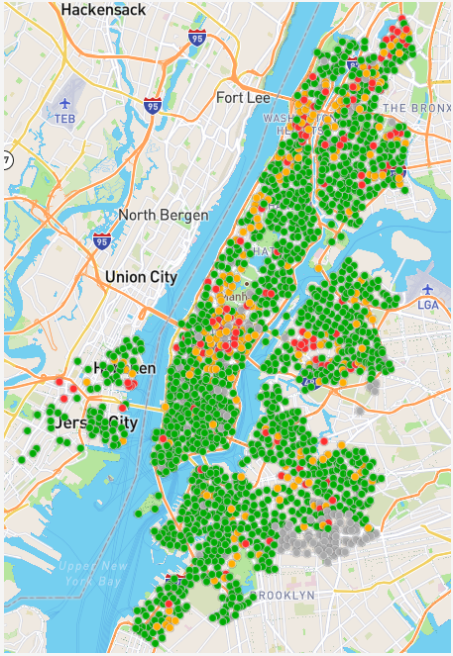
Klassifizierung von CitiBike-Kunden

Thomas Wagner

Agenda

1. Projektbeschreibung
2. Datenset und Features
3. Feature-Visualisierung
4. Modell-Training und -Auswahl
5. Kooperationsmöglichkeiten mit einer Versicherung

CitiBike



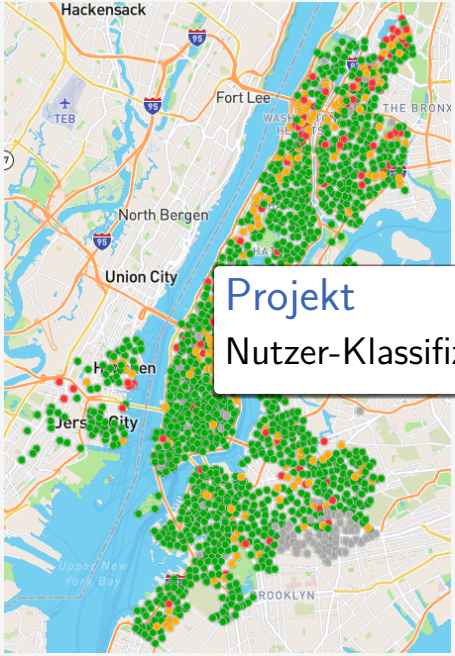
CitiBike in 2022

- über 24.000 Fahrräder
- über 1.500 Verleihstationen

Subscription-Modell

- Subscriber:
  - Jährliche Mitgliedschaft (15 $ / Monat)
- Customer:
  - 4 $ einfache Fahrt
  - 15 $ Tagespass

CitiBike



CitiBike in 2022

- über 24.000 Fahrräder
- über 1.500 Verleihstationen

Projekt

Nutzer-Klassifizierung (Customer/Subscriber)

- Jährliche Mitgliedschaft (15 $ / Monat)
- Customer:
  - 4 $ einfache Fahrt
  - 15 $ Tagespass

Datensatz

Daten von 2018

17 Millionen Fahrten, 800 Stationen, 15.000 Fahrräder

	tripduration	starttime	stoptime	start station id	start station name	start station latitude	start station longitude	end station id	end station name	end station latitude	end station longitude	bikeid	usertype	birth year	gender
0	689	2018-12-01 00:00:04.302	2018-12-01 00:11:33.846	3359	E 68 St & Madison Ave	40.769157	-73.967035	164	E 47 St & 2 Ave	40.753231	-73.970325	35033	Subscriber	1989	1
1	204	2018-12-01 00:00:05.533	2018-12-01 00:03:30.523	3504	E 123 St & Lexington Ave	40.802926	-73.937900	3490	E 116 St & 2 Ave	40.796879	-73.937261	20501	Subscriber	1966	1
2	316	2018-12-01 00:00:10.233	2018-12-01 00:05:27.203	270	Adelphi St & Myrtle Ave	40.693083	-73.971789	243	Fulton St & Rockwell Pl	40.688226	-73.979382	18386	Subscriber	1984	1
3	726	2018-12-01 00:00:21.957	2018-12-01 00:12:28.183	495	W 47 St & 10 Ave	40.762699	-73.993012	3660	W 16 St & 8 Ave	40.741022	-74.001385	27616	Subscriber	1983	1
4	397	2018-12-01 00:00:29.632	2018-12-01 00:07:07.446	473	Rivington St & Chrystie St	40.721101	-73.991925	3467	W Broadway & Spring Street	40.724947	-74.001659	35096	Subscriber	1976	1

Gender: 0 = unknown, 1 = male, 2 = female Tripduration: in Sekunden

Vorgehen und Tools

Vorgehen

1. Daten laden und formatieren
2. Daten säubern
3. Daten analysieren und visualisieren
4. Modelle trainieren und auswählen

Tools

- Pandas
- Matplotlib, Seaborn
- Scikit-Learn

Vorgehen und Tools

Vorgehen

1. Daten laden und formatieren
2. Daten säubern
3. Daten analysieren und visualisieren
4. Modelle trainieren und auswählen

Tools

- Pandas
- Matplotlib, Seaborn
- Scikit-Learn

Vorgehen und Tools

Vorgehen

1. Daten laden und formatieren
2. Daten säubern
3. Daten analysieren und visualisieren
4. Modelle trainieren und auswählen

Tools

- Pandas
- Matplotlib, Seaborn
- Scikit-Learn

Vorgehen und Tools

Vorgehen

1. Daten laden und formatieren
2. Daten säubern
3. Daten analysieren und visualisieren
4. Modelle trainieren und auswählen

Tools

- Pandas
- Matplotlib, Seaborn
- Scikit-Learn

Agenda

Data-Loading

Daten-Format

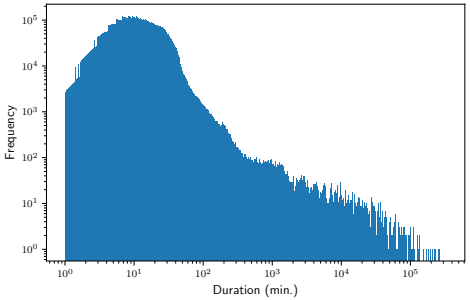
- Eine CSV-Datei pro Monat.
- Naives Einlesen benötigt $>$ 9 GB Speicher.

Daten-Konvertierung

- Pandas Category-dtype für Stationsnamen.
- Pandas Datetime-dtype für Timestamps.
- Zusammenführen von Kategorien über mehrere Dateien.
- Parquet-Format zum Erhalt von dtype-Information.

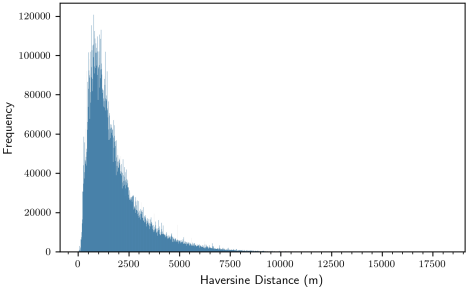
Wichtige Features

Fahrtdauer



- Typische Fahrten zwischen 2-60 min.
- Fahrten kürzer als 1 min. von CitiBike entfernt.
- 350.000 Rundfahrten (Startstation = Endstation)

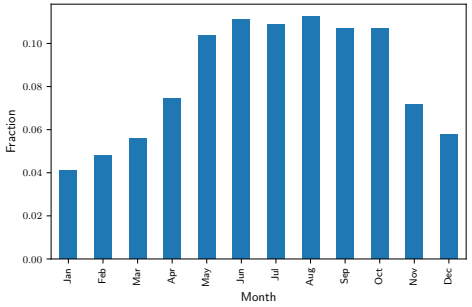
Haversine-Distanz



- Luftlinie Start- zu End-Koordianten
- Keine Routen Information
- Geschwindigkeit = Distanz / Fahrtdauer
- 0 für Rundfahrten

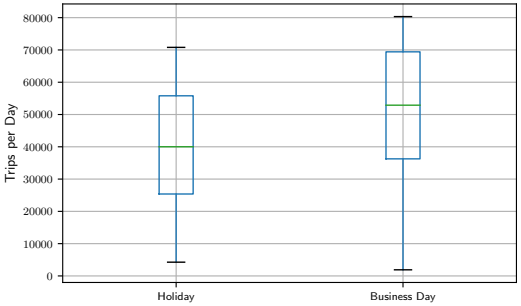
Wichtige Features

Jahreszeit



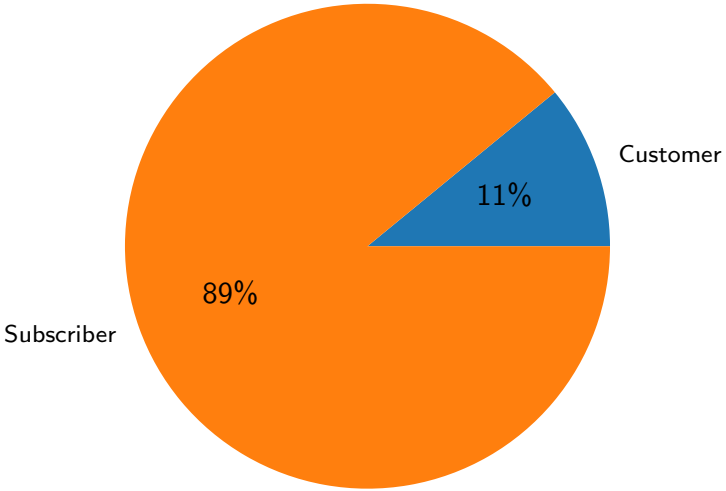
- Mehr Fahrten zwischen Mai und Oktober
- Kategorisierung in "Sommer" und "Winter"

Werktage



- 251 Werktage
- 114 Ferientage / Wochenenden

Nicht-Balancierte Daten



Data Cleaning

Entferne:

- 3000 Einträge: fehlenden Station IDs.
- 11.000 Einträge: geboren vor 1920.
- 65 Einträge: Latitude > 45 (Montreal?).
- 175 Einträge: Geschwindigkeit > 40 km/h.
- 15.000 Einträge: länger als 5 h
- 40.000 Einträge: Rundfahrt unter 2 min.

Training-Test Split

Validationset oder Kreuzvalidierung?

- Kreuzvalidierung:
  - K-facher Split des Datensets
  - K-Modelle: Training auf (K-1)-Teilen, Test auf einem Teil
  - Ermöglicht Nutzung von mehr Trainings Daten
  - Rechenzeit aufwendiger
- Hier:
  - Mehr als 17 Millionen Datenpunkte.
  - $80\% - 10\% - 10\%$ Training-Validation-Test Split.
  - SE auf der kleineren Klasse $\approx \pm 0.02\%$ (bei einer Genauigkeit von 99%).

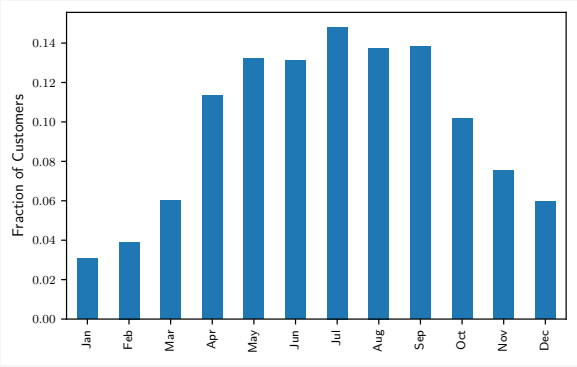
Keine Kreuzvalidierung nötig.

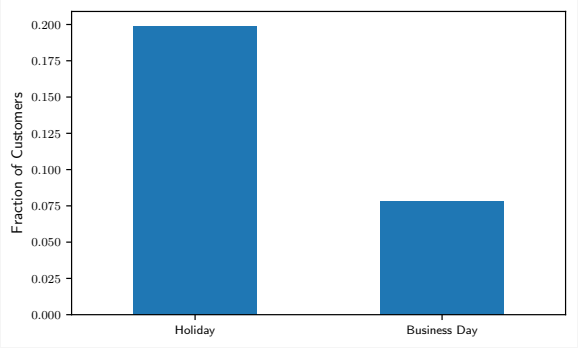
Zusammenfassung

- Bisher:
  - Betrachtung des gesamten Datensets
  - Erste Identifikation von Features
  - Keine Betrachtung von Labels
- Nächster Schritt:
  - Betrachtung der Labels (nur auf Trainingset)
  - Auswahl sinnvoller Features.

Agenda

Jahreszeit und Ferien



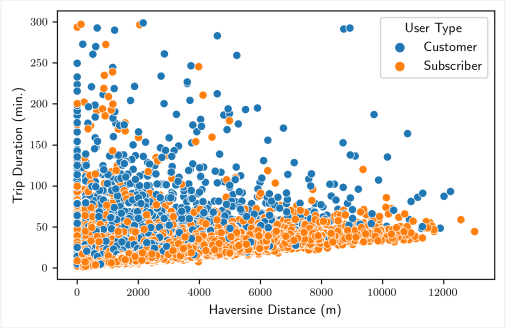


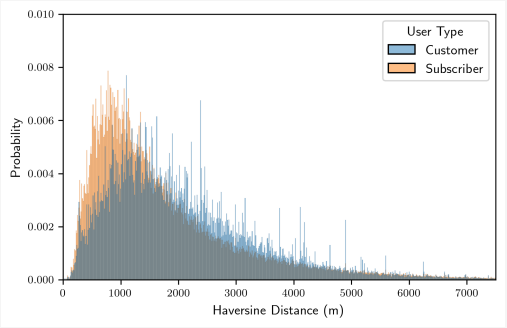
Zusammenfassung

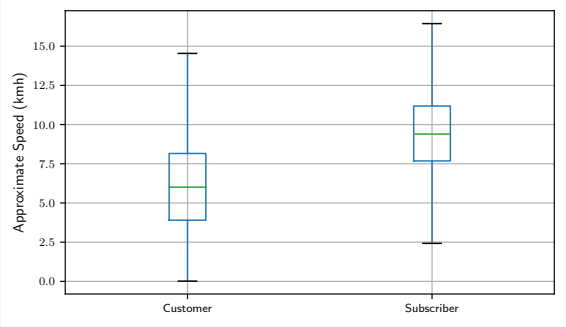
Customer fahren häufiger:

- im Sommer.
- an Feiertagen und Wochenenden.
- Kategorisierte Features.

Distanz und Geschwindigkeit





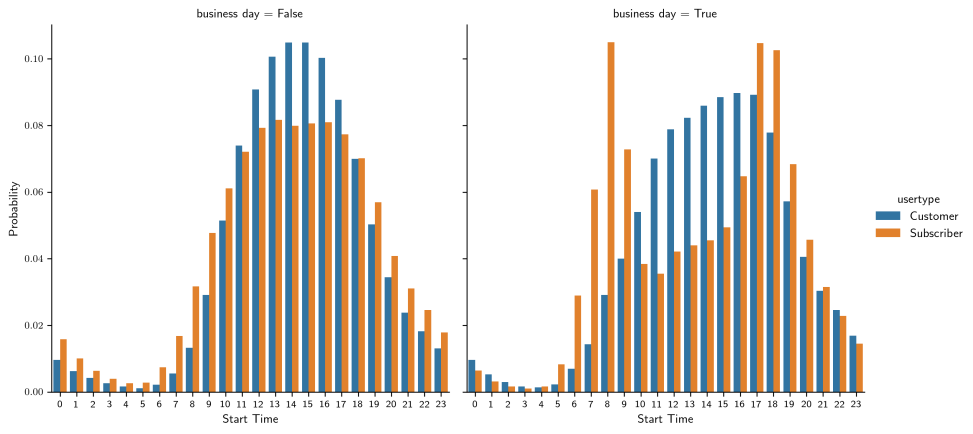


Zusammenfassung

Customer machen:

- mehr Rundfahrten.
- langsamere Fahrten.
- etwas längere Fahrten.

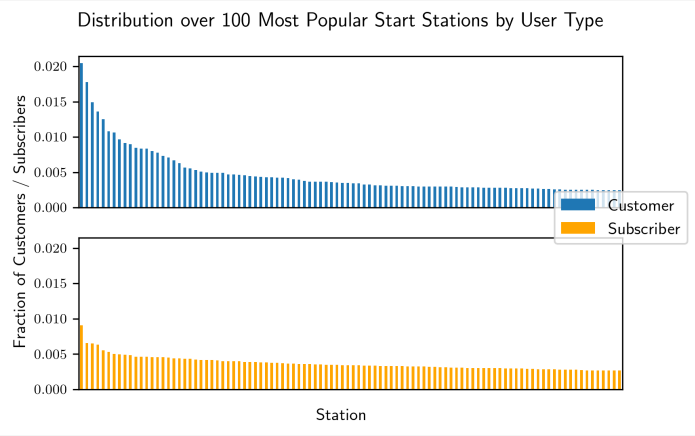
Start Zeit



Zusammenfassung

- Subscriber nutzen CitiBike auf dem Weg zur Arbeit.
- Interaktion Werktag x Tageszeit

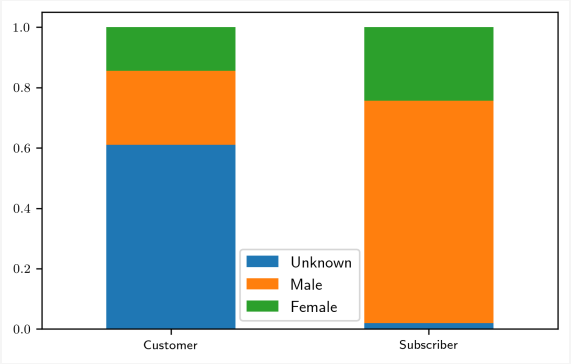
Stationen

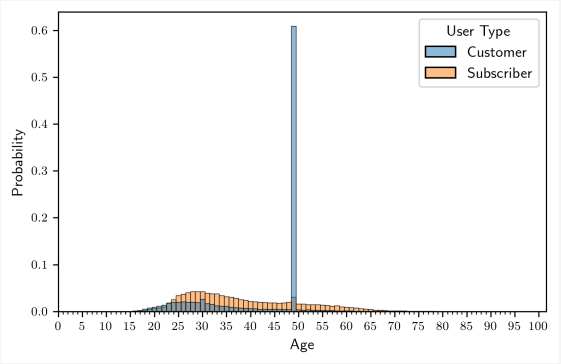


Zusammenfassung

- Customer konzentrieren sich stärker um wenige Stationen.
- Die 20 beliebtesten Stationen für Customer / Subscriber haben nur 2 gemeinsam.
- Beliebteste Station Customer: Central Park Subscriber: Pershing Square

Geschlecht und Alter





Zusammenfassung

- Geschlecht von Customern ist typischerweise unbekannt.
- Artefakte des Registrierungsprozesses.

Zusammenfassung

Customer oft Touristen.

Sinnvolle Features:

- Sommer/Winter
- Tageszeit
- Werktag
- Distanz
- Geschwindigkeit
- Rundfahrt
- Station

Irreführende Features:

- Geschlecht
- Alter

Agenda

1. Projektbeschreibung
2. Datenset und Features
3. Feature-Visualisierung
4. Modell-Training und -Auswahl
5. Kooperationsmöglichkeiten mit einer Versicherung

Nicht-Balancierte Daten

Evaluierung

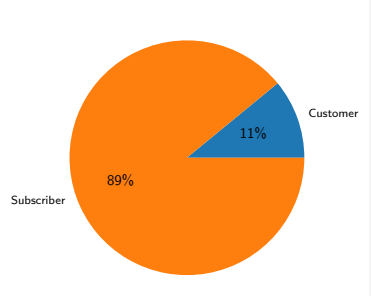
- Baseline-Genauigkeit 89%
- Confusion Matrix
- Matthews Correlation Coefficient (MCC)
  - Korreliert vorhergesagte und echte Klasse.
  - Zwischen -1 und 1, Baseline 0
  - Alle Felder der Confusion Matrix

Confusion Matrix

Echt	Vorhergesagt	
	Subscriber	Customer
Subscriber	TN	FN
Customer	FP	TP

Class-Balancing

- Gewichtung mit inversem Anteil
- Implizite Replizierung der seltenen Klasse



Logistic Regression

Modell

- Lineares Modell für Log-Odds:

$$\log(\frac{p}{1-p}) = X\beta$$

- Lineare Decision-Boundary $x \cdot \beta = 0$.
- Feature-Codierung sehr relevant.
- Weight-Decay Regularisierung $\|\beta\|_2^2 \rightarrow$ Feature-Skalierung relevant.
- Interaktionen müssen von Hand eingebaut werden.
- Baseline für andere Modelle.

Logistic Regression

Design

Features: Fahrtdauer, Sommer, Werktag, Distanz, Rundfahrt, Geschwindigkeit

- Station:
  - Ordinal?
  - Kategorisiert?
  - Frequenz codiert?
- Tageszeit:
  - Ordinal?
  - Kategorisiert?
  - Zyklisch codiert?

Interaktionen: Tageszeit x Werktag, Start- x Endstation
Skalierung: Min-Max Skalierung in das Intervall $[0, 1]$.
Training: Class-Balancing?

Logistic Regression

Baseline-Genauigkeit: 89,1 %

Ohne Class-Balancing

- Genauigkeit: 90,4 %
- Confusion:

Echt	Vorhergesagt	
	Subscriber	Customer
Subscriber	$98,2\%$	$1,8\%$
Customer	$72,7\%$	$27,3\%$

- MCC: $0,38$

Mit Class-Balancing

- Genauigkeit: 79,5 %
- Confusion:

Echt	Vorhergesagt	
	Subscriber	Customer
Subscriber	$79,6\%$	$20,4\%$
Customer	$21,7\%$	$78,3\%$

- MCC: $0,41$

Wichtige Features

Fahrtdauer, Distanz, Geschwindigkeit, Stationen

Decision-Tree und Random-Forest

Decision Tree

- Serie von Splits des Datensets.
- Nicht-Lineares Modell.
- Findet Interaktionen.
- Tendiert zu Overfitting.

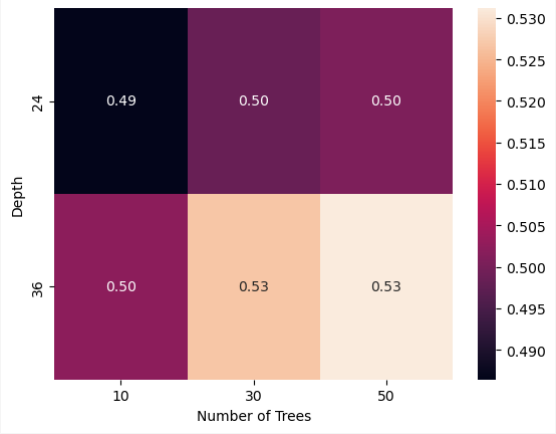
Random-Forest

- Bagging mehrerer Decision-Trees.
- Bootstrap-Sampling der Training-Daten.
- Zufällige Maskierung von Features in jedem Split.
- Kontrolliert Overfitting.

Decision-Tree

Design

- Feature-Codierung und -Skalierung weniger relevant.
- Class-Balancing
- Mehrere Hyperparameter, z.B.: Maximale Tiefe, Anzahl Bäume



Finale Auswertung

Performance-Vorhersage

Training: Auf Training + Validation Set
Evaluation: Auf Test Set

Forest

Insgesamt bestes Modell.

- Genauigkeit: 91,8 %
- Confusion:

Echt	Vorhergesagt	
	Subscriber	Customer
Subscriber	$97,2\%$	$2,8\%$
Customer	$51,2\%$	$48,7\%$

- MCC: $0,53$

Decision-Tree

Hohe Genauigkeit auf Customers.

- Genauigkeit: 82,9 %
- Confusion:

Echt	Vorhergesagt	
	Subscriber	Customer
Subscriber	$84,2\%$	$15,8\%$
Customer	$28,1\%$	$71,9\%$

- MCC: $0,42$

Zusammenfassung

- Modell Wahl hängt vom Anwendungsfall ab.
- Insgesamt bestes Modell (MCC):
  Großer Random-Forest
- Klares Overfitting vor allem auf Customers, Verbesserung durch mehr Trees oder Pruning?
- Hohe Genauigkeit auf Customers:
  Kleinerer Decision-Tree/Logistic Regression
- Hohe Genauigkeit auf Customers z.B. relevant im Marketing.

Agenda

1. Projektbeschreibung
2. Datenset und Features
3. Feature-Visualisierung
4. Modell-Training und -Auswahl
5. Kooperationsmöglichkeiten mit einer Versicherung

Kooperation mit einer Versicherung

Analyse der Unfallstatistik des NYPD

- Radfahrer haben 4-5 mal weniger Unfälle pro Fahrt als Taxis.
- Die Rate an Verletzungen pro Fahrt ist ebenfalls etwas niedriger.
- Bei einem Fahrradunfall wird in ca. 70% der Fälle der Radfahrer verletzt.
- Bei ca. 40% der Unfälle mit Fahrrädern hatte der Fahrradfahrer einen Beitrag.
- Bei ca. 2% der Unfälle mit Fahrrädern hatte das Fahrrad einen Defekt.

Kooperationsmöglichkeiten

CitiBike übernimmt bei einem Unfall keine Kosten, außer bei defekten Rädern.

- Unfallversicherung für Nutzer (häufige Personenschäden).
  - Viele Fahrten allerdings auf dem Arbeitsweg.
- Haftpflichtversicherung für Nutzer (häufige Mitschuld).
- Versicherung des Verleihers im Fall defekter Räder (selten).
- Rabatte bei Versicherungen für Subscriber.

Decision Tree and Random Forest