**Überblick**

* **SKU-Identifier:** Wir stehen aktuell noch vor dem Problem, dass wir dem Modell keine Informationen mitgeben, die es dem Modell ermöglichen,
* **Feature Reduction / Selection:** Zumindest wenn es später um die Erstellung von Prescriptions geht, ist es für die Performance mit wenigen Daten (wenige Daten meint hierbei einen niedrigen Zeithorizont, also bei uns trotz 200k Beobachtungen lediglich etwas mehr als 50 Wochen) wichtig, dass wir mit wenigen Featuren arbeiten oder zumindest ein Verfahren wir Random Forest nutzen, welches Feature Selection betreibt. In dem Paper von Richard und Pascal wird nämlich gezeigt, dass die Lower Rate of Convergence exponentiell mit der Anzahl der Feature p abnimmt. Dies sollten wir immer im Hinterkopf behalten. Ich stelle mir dementsprechend die Frage, warum wir in allen möglichen Datensätzen des Lehrstuhls, die ich bisher gesehen habe, alles Mögliche mit Featuren vollballern. Machen wir das, weil wir auf die Selektions-Kraft von Random Forest vertrauen?
* **Semi-Observable Features:** Es gibt zahlreiche Möglichkeiten Feature zu entwerfen, die nur Observable für die aktuell kommende Woche sind. Dies betrifft in der Regel Feature, die die Tagesstruktur der Bestellungen betreffen oder einfach Features, die auf Basis unserer Wochen-Forecasts nicht sinnvoll extrapoliert werden können. Es wäre mal ganz nett zu sehen, welche Einfluss diese Feature auf einen n.ahead = 1 Forecast haben können.
  + Probleme: Keine
  + Aufwand: 1-2 Tage
  + Potential: gering bis mittel
* **Graphische Untersuchung der Feature-Impacts:** Muss nicht unbedingt graphisch geschehen. Ein kleines Framework, welches es erleichtert, auf die schnelle den Einfluss einzelner Feature auf den Forecast oder zukünftig die Prescription zu ermitteln, wäre super. Dies kann auch Aufschluss darüber geben, wo unser Modell noch hackt. Dies kann neue Verbesserungspotentiale zu Tage bringen.
  + Probleme: Keine
  + Aufwand: 1 Tag
  + Potential: mittel
* **Stock-Out Feature** bauen: Dieses Feature trackt einfach wie viele Stock-Outs in der letzten Woche insgesamt aufgetreten sind. Evtl. besteht hier ein prädikativer Wert für die nächste Woche.
  + Probleme: Schwierig nutzbar für Forecasts, die weiter als eine Woche in die Zukunft prognostizieren, denn es gibt keinen sinnvollen Weg, die Anzahl der Stock-Outs zu extrapolieren 🡺 Sobald wir mit Stock-Levels arbeiten (ob künstlich oder echt, erstmal egal), so kann dieses Feature sinnvoll eingebaut werden, denn hierauf können wir unsere Forecasts nutzen, um die Anzahl an Stock-Outs zu „bestimmen“. Wir würden hierbei identisch zur Vorgehensweise bei der Interpolation der Demand-Lags vorgehen.
  + Aufwand: gering
  + Potential: gering
* **Detrending**: Dies ist eine der wichtigsten Bausteine unseres Modells überhaupt: Wie können wir den starken Trend, der in den Daten vorliegt, sinnvoll extrapolieren?
  + Probleme: Detrending muss für langfristigen Forecast nutzbar sein
  + Aufwand: 1-2 Tage Vollzeit
  + Potential: mittel bis hoch
* **Basket-Case** Analyse: Wir analysieren, welche Produkte in der Regel gemeinsam und welche sehr selten bzw. gar nicht gemeinsam bestellt werden. Dies könnte helfen, um Interdependenzen zwischen den einzelnen SKUs besser tracken zu können.
  + Vorbereitung: Datensatz muss evtl. für diesen Zweck aufbereitet werden.
  + Probleme: Keine
  + Aufwand: 2-4 Tage Vollzeit
  + Potential: mittel bis hoch
* **Stock-Out Impact Analysis** bzw. **Comment Impact Analysis**: Diese Analyse soll dem Zweck dienen, den Effekt von Shortages genauer quantifizieren bzw. klassifizieren zu können. Zu diesem Zweck müssen die Daten aufbereitet werden und wir uns überlegen, welche Kommentare überhaupt sinnvollerweise betrachtet werden sollen. Siehe hierzu die Analyse weiter unten.
  + Vorbereitung: Datensatz sollte sinnvoll aufbereitet werden, um die Analyse erheblich zu erleichtern.
  + Probleme: Viele Kommentare treten nur sehr selten auf, wodurch diese nur schwer nutzbar sind.
  + Aufwand: Schwer einschätzbar. Aufbereitung der Daten 1-2 Tage, Analyse beliebig
  + Potential: Für Business Insides und Inventory Management hoch, für reines Forecasting gering.
* **Inkonsistente Preisänderungen**: Es lagen knapp 3000x Preisänderungen vor, wobei in der Regel unit\_selling\_price bereits den upgedateten Preis angibt. In 120 Fällen ist das nicht der Fall. Diese Fälle sind nur sehr schwer automatisiert handhabbar, denn in einige Fällen treten massive Korrekturen auf (ungefähr Faktor 10), bei welchen es sich um tatsächlich korrekte Änderungen handelt. In anderen Fällen wurden falsche Änderungen in den Kommentaren angegeben, was man durch das Einsehen der Rechnungen sieht. Evtl. sollte man hier die 120 Fälle einfach löschen.
  + Probleme: Keine
  + Aufwand: sehr gering
  + Potential: sehr gering

**Details**

**Detrending**

Es ist etwas überraschend, dass das Modell, welches mit Detrending arbeitet, nicht besser funktioniert als die Modelle ohne Detrending. Vielleicht ändert sich das bei einem größer werdenden Zeithorizont der Forecasts, weil dort der Effekt des Trends stärker zum Tragen kommt. Dies sollten wir definitiv testen. Es wäre zu erwarten, dass alle Modelle ohne Detrending erheblich schlechter performen bei einem 3-Monats-Forecast, denn xgBoost bspw. interpoliert nur in-sample Outcomes für seine Predictions.

Der Trend sollte auf den ersten Blick mit der Anzahl der Pharmacies selbst und deren durchschnittlichen Kauf-Verhalten zusammenhängen wie „durchschnittliche Anzahl an Bestellungen pro Monat“ oder „durchschnittlicher Umsatz pro Pharmacy pro Monat“. Die Entwicklung dieser Parameter sollten wir uns nochmals in Relation zum aggregierten Demand genauer anschauen. Vielleicht liegt ein anderer Art Trend vor als wir erwarten würden.

**Erste Erkenntnisse:** Hinweis: Ich habe hierbei den Zusammenhang zwischen dem wöchentlichen Demand an Einzel-Tabletten usw. betrachtet aggregiert über alle Produkte. Diese Größe macht meiner Meinung nach am meisten Sinn, was sich vor allem dadurch nochmals bestätigt hat, dass der durchschnittliche Preis pro Unit einer ganzen Woche über die Zeit überraschend stabil ist. Der durchschnittliche Preis pro Unit bzw. Umsatz pro Unit schwankt hierbei um maximal 5-6%, was nahelegt, dass der Produkt-Mix, welcher bestellt wird, relativ stabil ist. Gleichzeitig ist aber schon ein kleiner Trend im bezahlten Preis zu beobachten, welcher vor allem zum Ende der Daten hin noch größer wird, wo der durchschnittliche Preis um knapp 10% anzieht. Der konstante Trend könnte durch Inflation erklärbar sein, wohingegen der Preisanstieg am Ende wohl Corona-bedingt sein könnte.

Ein lineares Modell zwischen diesem aggregierten Demand und der Anzahl der Pharmacies (entweder pro Monat oder seit Beginn) erzielt ein R^2 von über 0,85, was natürlich sehr sehr viel ist. Was hierbei nicht ganz unwichtig ist, ist die Tatsache, dass die Größe der Residuen von der Größe des Demands selbst abhängt (ähnlich bei einer multiplikativen Saisonalität). Dies widerspricht der Grundannahme der Linearen Regression, denn die Errors sind nicht stationär. Nichtsdestotrotz handelt es sich hierbei um einen grundsoliden Weg, um in diesem Fall ein Detrending durchzuführen.

**Comment/Stock-Out Impact**

Was genau müssen wir tun? Wir wollen bspw. herausfinden, ob ein oder mehrere Stock-Outs einer einzelnen Pharmacy das Bestell-Verhalten dieser Pharmacy mittel- bzw. langfristig beeinflussen. Dies kann man noch erweitern auf weitere vorliegende Spezialfälle, die die Comments in den Daten hergeben.

Sinnvolle Kommentare:

* **Stock-Out:** Sowohl in comments als auch item\_comment. Insgesamt knapp 18000x
* **NEAR EXPIRY:** Sowohl in comments als auch item\_comment. Knapp über 1000x
* **Price/Discount Change:** Das selbsterstellte Feature „priceChanged“ kann hierfür genutzt werden, welches knapp 3000x TRUE ist. Allerdings liegen die originalen Preise in der Regel nicht vor. In den Fällen, in denen sowohl neuer als auch alter Preis vorliegt (120x), ist dieser oftmals nicht korrekt (siehe oben).

Sollte mit „discChanged“ analog betrachtet werden.

* **Sent Wrong Product:** Sowohl in comments als auch item\_comment. Knapp 140x. Kommt auch als „WRONG\_ITEM“ vor
* **Below Minimum Invoice:** Sowohl in comments als auch item\_comments. Knapp über 1200x.

Die Aufbereitung des Datensatzes sollte sich nicht allzu stark vom originalen Datensatz unterscheiden. Für jede Pharmacy tracken wir jede Order inkl. der ausgewählten Kommentare. Felix meinte irgendwas in die Richtung, am Ende noch ein kleines Modell hierauf zu trainieren. Was genau hat er hier gemeint?