Absatz- und Nachfrageanalyse in der Region Guayas

Sinem Ara-Yücel | Masterschool

Projektarbeit: Modul Time Series

GitHub Projektlink: [Sinem-B62/retail_demand_analysis]

Projektbericht: Woche 1

Berichtszeitraum: Woche 1

Ziel: Vorbereitung und Analyse der Datengrundlage als Fundament für die Modellentwicklung.

Datenimport und -aufbereitung

Alle relevanten Datensätze wurden erfolgreich importiert und für die Analyse vorbereitet. Dabei wurden notwendige Funktionen eingerichtet, um einen reibungslosen Zugriff sowie eine effiziente Verarbeitung und Datenqualität sicherzustellen.

Identifizierung der Top-3-Produktfamilien

Zur Fokussierung auf umsatzstarke Bereiche wurde eine Analyse der Produktfamilien durchgeführt. Dabei wurden die drei häufigsten Segmente – „Grocery“ (Lebensmittel), „Beverages“ (Getränke) und „Cleaning“ (Reinigungsmittel) – identifiziert. Diese bildeten die Grundlage für weitere Analysen der Verkaufsdynamik.

Datenbereinigung

Zur Sicherstellung hoher Datenqualität wurden die Datensätze sorgfältig geprüft und bereinigt:

- Fehlende Werte**:** Systematisch identifiziert und mit geeigneten Methoden behandelt, um Datenverluste und Verzerrungen zu vermeiden.
- Ausreißer: Unregelmäßigkeiten in Verkaufs- oder Retourendaten erkannt und bereinigt, um die Modellintegrität zu gewährleisten.

Feature Engineering & Explorative Datenanalyse - Region Guayas

Zur Verbesserung der Modellleistung wurden neue Merkmale (Features) aus den Rohdaten abgeleitet, darunter zeitbezogene Variablen (z. B. Wochentag, Monat) und statistische Glättungen. Diese helfen, saisonale Muster und Trends im Kaufverhalten besser zu erkennen.

Abschließend wurde eine detaillierte EDA der Verkaufsdaten für die Region Guayas durchgeführt, um Muster, Trends und regionale Unterschiede zwischen Filialen zu identifizieren und ein tieferes Verständnis der Verkaufsentwicklung zu gewinnen.

Projektbericht: Woche 2

Berichtszeitraum: Woche 2

Ziel: Implementierung eines XGBoost-Nachfragemodells durch gezieltes Feature-Engineering und die Aufteilung der Daten für das Training und die Validierung.

Laden und Vorbereiten der Daten

Zu Beginn wurden die vorbereiteten CSV- und Parquet-Dateien geladen und auf Einträge der Provinz Guayas gefiltert. Dazu wurden die Verkaufsdaten mit den Filialinformationen (df_stores) verknüpft, um eine konsistente Datengrundlage zu schaffen.

Auf Basis der Erkenntnisse aus der vorherigen Analyse wurde der Datensatz gezielt eingeschränkt:

- Top-3-Produktfamilien: Fokus auf Grocery, Beverages und Cleaning, um die relevantesten Segmente für das Modelltraining zu berücksichtigen.
- Zeitraum: Eingrenzung auf das erste Quartal 2014 (1. Januar – 31. März) für ein kompaktes, schnelles Modelltraining.

Feature Engineering

Zur Verbesserung der Modellleistung (**XGBoost**) wurden aussagekräftige Merkmale erstellt:

Zeitverzögerungen (Lags): Lag-Features (*lag_1*, *lag_7*, *lag_30*) zur Erfassung von Verkaufsdynamiken vergangener Zeiträume.

Rollierende Statistiken: Berechnung eines gleitenden Mittelwerts (*rolling_std_7*) zur Erfassung von Stabilität und Volatilität.

Metadaten: Ergänzung um Artikel- und Filialinformationen (df_items, df_stores) für zusätzliche Kontextmerkmale wie Artikelklasse oder Filialtyp.

Aufteilung in Trainings- und Testdaten

Trainingsdaten: Januar und Februar 2014 (ca. 513.000 Zeilen) und Testdaten: März 2014 (ca. 285.000 Zeilen)

Anschließend erfolgte die Trennung in Feature-Matrizen (X) und Zielvektoren (y), wobei die Variable unit_sales als Zielgröße diente.

Projektbericht: Woche 3

Berichtszeitraum: Woche 3

Ziel: Aufbau, Training, Bewertung und Optimierung eines XGBoost-Modells zur Nachfrageprognose.

XGBoost-Modellaufbau

In dieser Phase lag der Fokus auf der Implementierung des Prognosemodells. Als Grundlage wurde ein **XGBoost-Modell** gewählt, das sich durch hohe Genauigkeit und Effizienz bei tabellarischen Daten auszeichnet. Das Modell wurde initial konfiguriert und trainiert, um eine stabile Basis für die weitere Optimierung zu schaffen.

Erstellung und Bewertung einer XGBoost-Baseline

Zunächst wurde ein **Baseline-Modell** trainiert, das als Referenzpunkt für spätere Optimierungen diente. Die Leistungsbewertung dieses Basismodells ermöglichte ein grundlegendes Verständnis seiner Vorhersagekraft und diente als Vergleichsmaßstab für alle weiteren Modellanpassungen.

Hyperparameter-Tuning zur Optimierung

Zur Verbesserung der Modellleistung wurde ein **systematisches Hyperparameter-Tuning** durchgeführt. Dabei wurden verschiedene Konfigurationen getestet, um jene Parameterkombination zu identifizieren, die die besten Prognoseergebnisse liefert.
Dieser Schritt war entscheidend, um das volle Potenzial des **XGBoost-Algorithmus** auszuschöpfen.

MLflow-Setup für das Experiment-Tracking

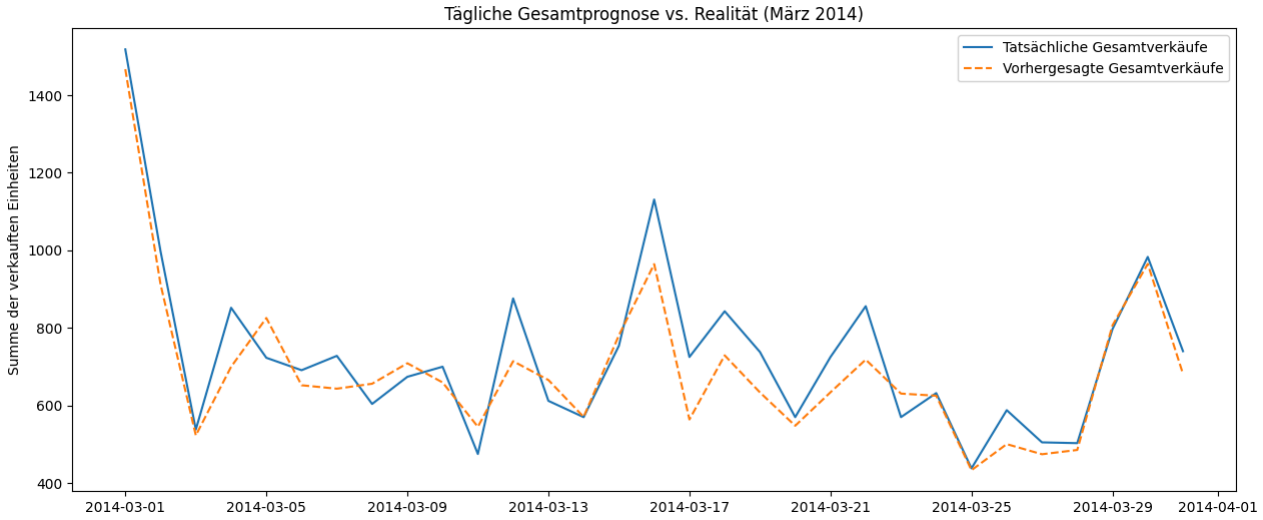
Zur Nachvollziehbarkeit und Transparenz des Entwicklungsprozesses wurde **MLflow** eingerichtet. Damit konnten alle Experimente, Parameter und Metriken (z. B. MAE, RMSE) automatisch protokolliert und verglichen werden. So ließ sich der Trainingsprozess effizient dokumentieren und die Reproduzierbarkeit der Ergebnisse sicherstellen.

Vergleich: Baseline vs. Optimiertes Modell

Nach dem Tuning wurde das **optimierte Modell** mit der ursprünglichen Baseline verglichen. Die Ergebnisse zeigten eine deutliche Verbesserung der Prognosegenauigkeit, insbesondere bei der Vorhersage der täglichen Produktnachfrage.
Das finale Modell lieferte somit verlässliche und stabile Ergebnisse und bildet die Grundlage für weiterführende Analys

Gewonnene Erkenntnisse 💡

Trends gut, Spitzen schlecht: Das Modell erkennt allgemeine Wochentrends zuverlässig, unterschätzt reale Verkaufsspitzen aber massiv. (Belegt durch Zeitreihen- und Streudiagramm). Es ist zu konservativ und sollte nur als Baseline für den Mindestbedarf genutzt werden.

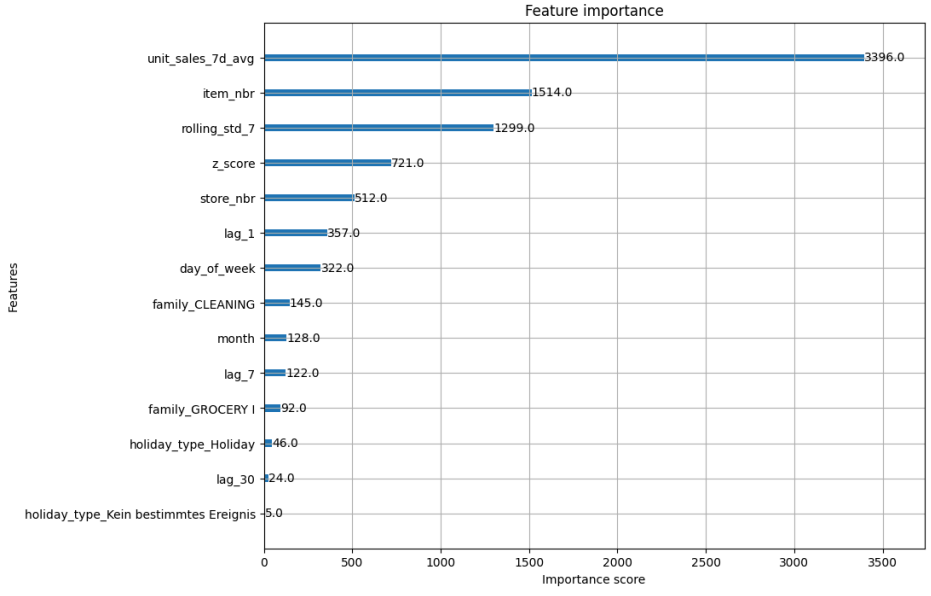


Metriken bestätigen Schwäche: Die Metriken (MAPE: 30.34%) bestätigen dies. Obwohl der durchschnittliche Fehler (MAE: 0.22) niedrig ist (weil das Modell Tage mit 0 Verkäufen gut trifft), ist der prozentuale Fehler an den wichtigen, umsatzstarken Tagen sehr hoch.

Der **7-Tage-Durchschnitt der Verkäufe** ist der stärkste Indikator für zukünftige Nachfrage. → Das Modell erkennt ein starkes **Zeitreihenmuster** (Nachfrage hängt stark von den letzten Tagen ab).

Unterschiedliche Produkte haben stark unterschiedliche Absatzmuster – das Modell hat gelernt, dass der Artikeltyp selbst einen großen Einfluss hat.

Die **Volatilität (Schwankung)** der Verkäufe in der letzten Woche ist relevant – bei stabilen oder unregelmäßigen Verkäufen verhält sich die Nachfrage unterschiedlich.



Handlungsempfehlungen 👩‍💻

Operative Sofortmaßnahmen

1. Prognose als Baseline + Sicherheitsbestand nutzen: Da das Modell Spitzen unterschätzt, nutzen Sie die Prognose nur als Mindestbedarf (Baseline). Kombinieren Sie dies mit einem dynamischen Sicherheitsbestand, der sich an der Volatilität (rolling_std_7) orientiert.

2. Problem-Artikel manuell steuern: Artikel mit hohem Einfluss (item_nbr) und hoher Schwankung müssen aus der Automatik genommen und manuell vom Kategoriemanagement überwacht werden.

Strategische Modellverbesserung

1. Wahre Treiber finden (Zahltage & Marketing): Die holiday_type-Features sind irrelevant. Die Erfassung von Zahltagen (Paydays) und detaillierten Marketing-Aktionen (statt nur "onpromotion") hat jetzt höchste Priorität, um die Spitzen zu erklären.

2. Datenaktualität sicherstellen: Das wichtigste Feature ist der unit_sales_7d_avg (7-Tage-Schnitt). Die tägliche Aktualität und Zuverlässigkeit der Daten-Pipeline ist geschäftskritisch für die Prognosegenauigkeit.

Persönliche Selbstreflexion zum Projekt - Nachfrageprognose für Guayas 💫

Das Projekt stellte aus meiner Sicht eine wertvolle Vorbereitung auf zukünftige berufsrelevante Data-Science-Projekte dar. Der chronologische Aufbau der Aufgaben ermöglichte einen klaren Überblick über die einzelnen Arbeitsschritte im gesamten Zeitreihenanalyse-Projekt. Besonders hilfreich empfand ich die praxisnahen Kontexte, die strukturierte Aufgabenplanung sowie die verständliche Erläuterung der jeweiligen Wochenaufgaben im Rahmen der Weiterbildung bei **Masterschool**.

Während der Projektarbeit traten jedoch auch einige Herausforderungen auf, die einen höheren Zeitaufwand erforderten als ursprünglich vorgesehen. Insbesondere die Empfehlung, zwei Modelle (LSTM und XGBoost) zu implementieren, erwies sich als anspruchsvoll. Beim Arbeiten in Google Colab kam es häufig zu technischen Problemen und Systemabstürzen, wodurch sich die Entwicklungszeit deutlich verlängerte.

Auch die Integration zusätzlicher Features gestaltete sich zeitintensiver als erwartet, da umfangreiche Recherchen und Tests erforderlich waren. Nach wiederholten Stabilitätsproblemen im Colab-Umfeld habe ich mich schließlich bewusst entschieden, mich auf das **XGBoost-Modell** zu konzentrieren, um konsistente und reproduzierbare Ergebnisse zu erzielen.

Die Implementierung einer **Web-App** war ursprünglich nur für das LSTM-Modell vorgesehen, sodass ich für das XGBoost-Modell weitgehend selbstständig Lösungen erarbeiten musste. Dies erforderte zusätzliche Einarbeitung und Eigeninitiative. Trotz dieser technischen und organisatorischen Herausforderungen konnte ich wertvolle praktische Erfahrungen sammeln – insbesondere im Umgang mit begrenzten Ressourcen (RAM-Nutzung) und im selbständigen Troubleshooting.

Insgesamt war das Projekt **praxisnah, lehrreich und fordernd**. Es hat mir nicht nur geholfen, mein technisches Wissen zu vertiefen, sondern auch meine Problemlösungsfähigkeit, Ausdauer und Selbstorganisation zu stärken.