IHK-Projekt Dokumentation: Kundenprofilanlage für TickTockCouture Luxusuhren Werbekampagne

Datum: 20.11.2023

Gruppe1: Benjamin Binder – Constantin Naum – Kacper Benner – Klaus Reinhart

Einleitung und Ist-Analyse:

Unser erfahrenes Data-Analysten-Team von TickTockCouture hat sich das Ziel gesetzt, eine datengetriebene Werbekampagne für die neue Luxusuhrenkollektion zu entwickeln. Die Fokussierung liegt auf der präzisen Identifikation einer Zielgruppe mit einem Einkommen von mindestens 50.000 US-Dollar, basierend auf dem "Erwachsenen"-Datensatz aus dem Zensus.

Ausgangssituation und Hintergrund:

TickTockCouture, als führender Luxusuhrenhersteller, strebt an, die Vermarktung seiner neuen Kollektion durch eine hochgradig personalisierte und effektive Werbekampagne zu optimieren. Die Entscheidung für den "Erwachsenen"-Datensatz erfolgte aufgrund seiner umfassenden Informationen, die sich für eine detaillierte Zielgruppenanalyse eignen.

Backlog und Projektziele:

Im ersten Sprint haben wir, als erfahrenes Data-Analysten-Team, ein umfassendes Backlog zusammengestellt und klare Projektziele definiert:

Benutzeranforderungen:

Strukturanalyse des "Erwachsenen"-Datensatzes für eine präzise Analyse.

Identifikation von Schlüsselmerkmalen zur tiefgehenden Zielgruppenanalyse.

Klarstellung von Begriffen wie 'snlwgt' und Festlegung von Handhabungsrichtlinien für NaN-Werte.

Potenziale:

Potenzial 1: Identifikation der Top 5 Einflussfaktoren auf ein Einkommen über 50.000 US-Dollar.

Potenzial 2: Entwicklung eines klaren Backlogs für kommende Sprints.

User Stories und Hypothesen:

User Story 1:

Als Datenanalyst möchte ich den Datensatz gemäß den Bedingungen bereinigen, um eine einheitliche Datenbasis zu schaffen.

Hypothese 1:

Wenn wir die Daten bereinigen, wird die Datenqualität verbessert.

User Story 2:

Als Datenanalyst möchte ich maßgebliche Einflussfaktoren auf ein Einkommen über 50.000 US-Dollar identifizieren, um gezielte Analysen durchzuführen.

Hypothese 2:

Wenn wir maßgebliche Einflussfaktoren identifizieren, verbessern wir die Effektivität unserer Zielgruppenanalyse.

User Story 3:

Als Datenanalyst möchte ich ein interaktives PowerBI-Dashboard erstellen, um die Zielgruppenanalyse effektiv zu visualisieren.

Hypothese 3:

Wenn wir ein interaktives PowerBI-Dashboard erstellen, verbessern wir die Verständlichkeit unserer Präsentation.

Sprint 1 - Brainstorming und Datentransformation:

Im ersten Sprint haben wir intensive Brainstorming-Sitzungen abgehalten und die Daten durch sorgfältige Transformation vorbereitet:

Bereinigung des Datensatzes gemäß den festgelegten Bedingungen.

Übersetzung von englischen Datenbezeichnungen ins Deutsche für eine bessere Verständlichkeit.

Anpassung von Datentypen für eine effiziente Datennutzung.

Unsere Überlegungen zur Datentransformation umfassten auch die Klärung von Unklarheiten wie der 'snlwgt'-Spalte, wodurch eine einheitliche Datenbasis geschaffen wurde.

Sprint 2 - EDA und Machine Learning:

Im zweiten Sprint haben wir unsere umfassende Expertise in explorativer Datenanalyse (EDA) und maschinellem Lernen eingesetzt:

Identifikation der maßgeblichen Einflussfaktoren auf ein Einkommen über 50.000 US-Dollar.

Formulierung und Überprüfung von Hypothesen zur Validierung des Modells.

Konvertierung der Machine Learning-Ergebnisse für die nahtlose Integration in PowerBI.

Unsere Fokussierung auf Machine Learning umfasste auch die Evaluierung von Datenqualität und die sorgfältige Validierung unserer Modelle, um zuverlässige Ergebnisse sicherzustellen.

Sprint 3 - PowerBI und PowerPoint Präsentation:

Im dritten Sprint haben wir die gewonnenen Erkenntnisse in PowerBI implementiert und eine überzeugende PowerPoint-Präsentation erstellt:

Erstellung eines interaktiven Dashboards in PowerBI zur präzisen Zielgruppenanalyse.

Verfeinerung der PowerBI-Visualisierung für eine klare und überzeugende Präsentation.

Vorbereitung einer PowerPoint-Präsentation mit den herausragenden Erkenntnissen und Handlungsempfehlungen.

Unser Fokus im letzten Sprint lag darauf, nicht nur aussagekräftige Visualisierungen zu erstellen, sondern auch klare Empfehlungen abzuleiten, die direkt in die Marketingstrategie integriert werden können.

Ergebnisse und Verfeinerung:

Die implementierten Machine Learning-Ergebnisse wurden erfolgreich in PowerBI integriert, wodurch ein aussagekräftiges Dashboard entstand. Im letzten Sprint haben wir die PowerBI-Visualisierung weiter optimiert, um eine klare und überzeugende Präsentation der identifizierten Zielgruppe zu gewährleisten.

Fazit und Ausblick:

Als erfahrenes Data-Analysten-Team haben wir bedeutende Erkenntnisse zu den Einflussfaktoren für ein Einkommen über 50.000 US-Dollar gewonnen. Das PowerBI-Dashboard bietet eine benutzerfreundliche Plattform für die präzise Zielgruppenanalyse. Die PowerPoint-Präsentation liefert TickTockCouture eine solide Grundlage für eine effektive Werbekampagne, basierend auf dem identifizierten Kundenprofil. Wir empfehlen regelmäßige Datenaktualisierungen und fortlaufende Analysen, um den Veränderungen im Verhalten der Zielgruppe und den Verdienstmöglichkeiten gerecht zu werden.

Weitere Informationen zum Datensatz:

Extraktion: Barry Becker extrahierte den Datensatz aus der 1994 Census-Datenbank.

Bedingungen für Extraktion: ((AAGE>16) && (AGI>100) && (AFNLWGT>1) && (HRSWK>0))

Vorhersageaufgabe: Bestimmung, ob eine Person über 50.000 US-Dollar verdient.

Besprechung und Klärung:

Nach der Vorstellung der Ergebnisse sind einige Klärungsaufträge notwendig:

Spaltenüberschriften, insbesondere 'snlwgt', müssen weiter erläutert werden.

NaN-Werte sollen als 'nicht angegeben' kategorisiert werden.

Klärung der Eignung und Datentypen der Daten aus der Studie von 1996_5_19 für den Einsatz in Machine-Learning-Modellen.

Nächste Schritte und Verantwortlichkeiten:

Die erfolgreiche Umsetzung des Projekts "Kundenprofilanlage für TickTockCouture Luxusuhren Werbekampagne" war ein Teamerfolg, bei dem jedes Teammitglied eine spezifische Rolle übernahm. Die nächsten Schritte und Verantwortlichkeiten werden individuell zugewiesen, um sicherzustellen, dass alle Aspekte des Projekts nahtlos abgeschlossen werden.

Kacper Benner - Visualisierungen und Dashboard:

Verfeinerung und Optimierung des PowerBI-Dashboards.

Implementierung zusätzlicher interaktiver Visualisierungen für detaillierte Einblicke.

Überprüfung der Benutzerfreundlichkeit und Zugänglichkeit des Dashboards.

Benjamin Binder - KnimeFlow und Machine Learning:

Weiterentwicklung des Knime-Workflows für verbesserte Datenverarbeitung.

Feinabstimmung des Machine-Learning-Modells für höhere Präzision.

Überprüfung der Modellergebnisse und Anpassung bei Bedarf.

Constantin Naum - Report und Wissenschaftliche Arbeit:

Erstellung eines umfassenden Berichts, der die methodologischen Aspekte des Projekts erläutert.

Integration von wissenschaftlichen Erkenntnissen und Empfehlungen in den Bericht.

Sicherstellung einer klaren und präzisen Darstellung der Projektergebnisse.

Klaus Reinhart - Dokumentation und Reporting:

Ausarbeitung einer detaillierten Projektdokumentation unter Berücksichtigung aller Aspekte.

Zusammenstellung von klaren Anleitungen und Handbüchern für die zukünftige Nutzung des entwickelten Systems.

Erstellung von regelmäßigen Berichten über den Projektfortschritt für das Management.

Gemeinsame Teamsitzung und Abstimmung:

Einberufung einer gemeinsamen Teamsitzung zur Abstimmung der nächsten Schritte.

Klärung von offenen Fragen und Unsicherheiten.

Diskussion über mögliche Optimierungen und Erweiterungen des Projekts.

Rückblick und Lessons Learned:

Durchführung eines Rückblicks auf das abgeschlossene Projekt, um Erfahrungen und Erkenntnisse zu teilen.

Identifikation von Best Practices und Lessons Learned für zukünftige Projekte.

Festlegung von Verbesserungsbereichen für die kontinuierliche Weiterentwicklung der Teamleistung.

Abschließende Kundenpräsentation:

Vorbereitung einer prägnanten und überzeugenden Präsentation für den Kunden.

Vorstellung der Projektergebnisse, Methodik und Handlungsempfehlungen.

Beantwortung etwaiger Fragen und Anregungen des Kunden.

IHK-Projekt Dokumentation: KNIME-Workflow Analyse

Daten einlesen

- CSV-Reader: Daten Einlesen (Datenset_G1.csv)

Der CSV-Reader-Knoten spielt eine grundlegende Rolle, indem er die Daten aus der Datei

& quot; Datenset_G1.csv " in den KNIME-Workflow einliest. Dieser Schritt ist entscheidend, um auf das zugrunde liegende Datenset zugreifen zu können.

Erste Dateneinsicht

- Data Explorer: Erste Daten Exploration

Der Data Explorer-Knoten ermöglicht eine erste, explorative Analyse der eingelesenen Daten. Durch die Bereitstellung von statistischen Informationen und der Anzeige der ersten Datensätze gibt er einen Überblick über die Struktur und den Inhalt des Datensets. Transformation und Übersetzung

- Column Remaner: Übersetzung DE

Der Column Remaner-Knoten wird verwendet, um die Spaltennamen ins Deutsche zu übersetzen. Dies verbessert die Verständlichkeit der Daten und erleichtert die weitere Verarbeitung.

- Rule Engine: Arbeiterklasse DE
- Rule Engine: Famillienstand DE
- Rule Engine: Beruf DE
- Rule Engine: Beziehung DE
- Rule Engine: Rasse DE
- Rule Engine: Geschlecht DE
- Rule Engine: GeschlechtNR hinzugefügt in NR
- Rule Engine: EinkommenNR hinzugefügt in NR

Die Rule Engine-Knoten sind entscheidend für die Transformation der Daten. Sie ermöglichen das Anwenden von Regeln auf die verschiedenen Spalten, beispielsweise die Übersetzung von Arbeiterklasse, Familienstand, Beruf, Beziehung, Rasse und Geschlecht. Zudem werden neue Spalten, wie GeschlechtNR und EinkommenNR, hinzugefügt.

- Column Resorter: Strukturiert für Überblick

Der Column Resorter-Knoten strukturiert die Spalten, um eine bessere Übersicht und Verständlichkeit zu gewährleisten.

- Row Filter: Über 50K
- Column Filter: Unter 50K raus

Mithilfe von Row Filter und Column Filter werden die Daten in zwei Teile aufgeteilt: Datensätze mit einem Einkommen über 50K und Datensätze mit einem Einkommen unter 50K.

Daten-Export

- CSV Writer: CSV für PowerBI

Der CSV Writer-Knoten wird genutzt, um die bearbeiteten Daten in einem CSV-Format zu speichern. Dies ermöglicht die Verwendung der Daten in anderen Analysetools wie PowerBI.

Zweite Dateneinsicht

- Data Explorer: Zweite Daten Exploration

Der Data Explorer-Knoten führt eine weitere explorative Analyse durch, um sicherzustellen, dass die Transformationen die gewünschten Effekte haben und um mögliche Muster in den Daten zu entdecken.

- Date Explorer: Nur >=50K

Diese zweite Datenexploration konzentriert sich speziell auf Datensätze mit einem Einkommen über 50K, um detailliertere Einblicke in diese Gruppe zu gewinnen.

Explorative Datenanalyse (EDA)

- Linear Correlation: Einkommen
- Value Counter: Balance Testung
- EDA - Alle Daten
- Data Explorer: Platzhalter
- Linear Correlation: Einkommen
- EDA – Nur über 50K

Component Input:

- Bar Chart: Geschlecht / Alter

In diesem Abschnitt werden verschiedene EDA-Knoten verwendet, um eine umfassende explorative Datenanalyse durchzuführen. Dies schließt lineare Korrelationen, Wertzähler,

Heatmaps und Bar Charts ein, um Beziehungen zwischen den Variablen zu untersuchen und Muster in den Daten zu identifizieren.

Maschinelles Lernen

- K Nearest Neighbor (KNN)
- Parameter Optimierung: Opt.= k, Hillclimbing

Der K Nearest Neighbor (KNN)-Algorithmus wird verwendet, um Datensätze anhand der Ähnlichkeit zu k-ähnlichen Nachbarn zu klassifizieren. In diesem Schritt erfolgt die Optimierung des Parameters & quot;k & quot; durch Hillclimbing, um das bestmögliche k für die Modellleistung zu finden.

- Partitionierung: 80/20 STR Einkommen

Die Daten werden in Trainings- (80%) und Testsets (20%) aufgeteilt, um das Modell zu trainieren und zu evaluieren. Die Stratifikation erfolgt basierend auf der Einkommensklasse, um eine gleichmäßige Verteilung der Klassen in den Trainings- und Testdaten sicherzustellen.

- K Nearest Neighbor: Einkommen

Der KNN-Learner wird auf die Trainingsdaten angewendet, um das Modell zu trainieren und Muster im Zusammenhang mit dem Einkommen zu erlernen.

- Scorer: Class(kNN)+Einkommen

Der Scorer bewertet die Leistung des KNN-Modells durch den Vergleich der vorhergesagten Einkommensklasse mit den tatsächlichen Einkommen in den Testdaten.

- Parameter Optimization Loop End: max. Accuracy

Ein Optimierungszyklus endet, wenn die maximale Genauigkeit des Modells erreicht ist, basierend auf den verschiedenen Werten von "k".

- X-Partitioner: n = 10

Die Ergebnisse werden in 10 Teile (n=10) aufgeteilt, um die Robustheit des Modells zu überprüfen.

- K Nearest Neighbor: Einkommen

Der KNN-Algorithmus wird erneut auf die Daten angewendet, um zu sehen, wie gut das Modell auf verschiedenen Teilmengen der Daten generalisiert.

- X-Aggregator: Class(kNN)+Einkommen

Die Ergebnisse werden aggregiert, um einen umfassenden Überblick über die Leistung des KNN-Modells zu erhalten.

- Scorer

Der Scorer bewertet die aggregierten Ergebnisse und gibt Einblicke in die Qualität der

Modellvorhersagen.

Beschreibung des K Nearest Neighbor (KNN) Modells:

Vorteile:

- Einfach zu verstehen und zu implementieren.
- Effektiv für kleine Datensätze und nicht parametrisch.
- Kann für Klassifikation und Regression verwendet werden.

Nachteile:

- Empfindlich gegenüber Ausreißern und irrelevanten Features.
- Die Vorhersagezeit kann bei großen Datensätzen hoch sein.
- Benötigt eine geeignete Wahl des k-Werts für optimale Leistung.

Random Forest

- Parameter Optimierung: MaxLvL+NrModel, Hillclimbing

Der Random Forest-Algorithmus ist ein Ensemble-Lernalgorithmus, der auf Entscheidungsbäumen basiert. In diesem Schritt erfolgt die Optimierung der maximalen Baumtiefe und der Anzahl der Bäume durch Hillclimbing.

- Partitionierung: 80/20 STR Einkommen

Ähnlich wie bei KNN werden die Daten in Trainings- und Testsets aufgeteilt, um das Random Forest-Modell zu trainieren und zu bewerten.

- Random Forest Learner: EinkommenNR

Der Random Forest-Learner wird auf die Trainingsdaten angewendet, um das Modell zu erstellen, wobei die vorhergesagten Einkommen ohne Rundung (EinkommenNR) berücksichtigt werden.

- Random Forest Predictor

Der Predictor wendet das trainierte Random Forest-Modell auf die Testdaten an.

- Scorer: Ergebnis EinkommenNR

Der Scorer bewertet die Leistung des Random Forest-Modells unter Berücksichtigung der vorhergesagten Einkommen ohne Rundung.

- Parameter Optimization Loop End: max. Accuracy

Ein Optimierungszyklus endet, wenn die maximale Genauigkeit des Modells erreicht ist.

- X-Partitioner: n = 10

Die Ergebnisse werden erneut in 10 Teile aufgeteilt, um die Robustheit des Random Forest-Modells zu überprüfen.

- Random Forest Learner: EinkommenNR

Der Random Forest-Learner wird erneut auf die Daten angewendet, um die Generalisierung des Modells zu testen.

- Random Forest Predictor

Der Predictor wendet das trainierte Random Forest-Modell auf die Testdaten an.

- X-Aggregator: Predicted EinkommenNR

Die aggregierten Ergebnisse ermöglichen einen Gesamtüberblick über die Leistung des Random Forest-Modells.

- Scorer: Ergebnis EinkommenNR

Der Scorer bewertet die aggregierten Ergebnisse und gibt Einblicke in die Qualität der Modellvorhersagen.

Beschreibung des Random Forest Modells:

Vorteile:

- Robust gegenüber Overfitting und Ausreißern.
- Effektiv für hohe dimensionale Daten.
- Kann mit kategorialen und numerischen Daten umgehen.

Nachteile:

- Komplexität der Interpretation im Vergleich zu einzelnen Entscheidungsbäumen.
- Benötigt mehr Rechenressourcen und Zeit für das Training.

Gradient Boosted Trees

- Parameter Optimierung: LR+NrModels, Hillclimbing

Der Gradient Boosted Trees-Algorithmus ist ein weiteres Ensemble-Lernalgorithmus, der auf dem Boosting-Prinzip basiert. Die Optimierung beinhaltet die Lernrate (LR) und die Anzahl der Modelle.

- Partitionierung: 80/20 STR Einkommen

Wie zuvor werden die Daten in Trainings- und Testsets unterteilt.

- Gradients Boosted Trees Learner: EinkommenNR

Der Learner wird auf die Trainingsdaten angewendet, um das Gradient Boosted Trees-Modell zu trainieren.

- Gradients Boosted Trees Predictor

Der Predictor wendet das trainierte Modell auf die Testdaten an.

- Scorer

Der Scorer bewertet die Leistung des Gradient Boosted Trees-Modells.

- Parameter Optimization Loop End: max. Accuracy

Die Optimierung endet, wenn die maximale Genauigkeit erreicht ist.

- X-Partitioner:

Die Ergebnisse werden erneut in 10 Teile aufgeteilt, um die Robustheit des Modells zu überprüfen.

- Gradients Boosted Trees Learner: EinkommenNR

Der Learner wird erneut auf die Daten angewendet, um die Generalisierung des Modells zu testen.

- Gradients Boosted Trees Predictor

Der Predictor wendet das trainierte Modell auf die Testdaten an.

- X-Aggregator: Predicted EinkommenNR

Die aggregierten Ergebnisse ermöglichen einen Gesamtüberblick über die Leistung des Gradient Boosted Trees-Modells.

- Scorer

Der Scorer bewertet die aggregierten Ergebnisse und gibt Einblicke in die Qualität der Modellvorhersagen.

Beschreibung des Gradient Boosted Trees Modells:

- Vorteile:
- Robust gegenüber Overfitting und Ausreißern.
- Gute Leistung auch bei ungleichmäßigen Datenverteilungen.
- Fähigkeit, komplexe nicht-lineare Beziehungen zu modellieren.

Nachteile:

- Erfordert sorgfältige Abstimmung der Hyperparameter.
- Benötigt mehr Rechenressourcen und Zeit für das Training.

Naive Bayes

- Parameter Optimierung: threshold, Hillclimbing

Naive Bayes ist ein probabilistischer Klassifikationsalgorithmus, der auf dem Bayes-Theorem basiert. In diesem Schritt erfolgt die Optimierung des Schwellenwerts durch Hillclimbing.

- Partitionierung: 80/20 STR Einkommen

Die Daten werden in Trainings- und Testsets unterteilt.

- Naive Bayes Learner: EinkommenNR

Der Naive Bayes Learner wird auf die Trainingsdaten angewendet, um das Modell zu trainieren.

- Naive Bayes Predictor

Der Predictor wendet das trainierte Modell auf die Testdaten an.

- Scorer: Ergebnis EinkommenNR

Der Scorer bewertet die Leistung des Naive Bayes-Modells.

- Parameter Optimization Loop End: max. Accuracy

Die Optimierung endet, wenn die maximale Genauigkeit erreicht ist.

- X-Partitioner:

Die Ergebnisse werden erneut in 10 Teile aufgeteilt, um die Robustheit des Modells zu überprüfen.

- Naive Bayes Learner: EinkommenNR

Der Learner wird erneut auf die Daten angewendet, um die Generalisierung des Modells zu testen.

- Naive Bayes Predictor

Der Predictor wendet das trainierte Modell auf die Testdaten an.

- X-Aggregator: Predicted EinkommenNR

Die aggregierten Ergebnisse ermöglichen einen Gesamtüberblick über die Leistung des Naive Bayes-Modells.

- Scorer: Ergebnis EinkommenNR

Der Scorer bewertet die aggregierten Ergebnisse und gibt Einblicke in die Qualität der Modellvorhersagen.

Beschreibung des Naive Bayes Modells:

- Vorteile:
- Einfach und schnell zu implementieren.
- Effektiv für Textklassifikation und bei geringem Datenvolumen.
- Geringe Anfälligkeit gegenüber Overfitting.

Nachteile:

- Annahme von Unabhängigkeit zwischen den Merkmalen (naiv), was in einigen Fällen
- unrealistisch sein kann.
- Möglicher Informationsverlust aufgrund der Naivität der Annahmen.

Ergebnisse Transformieren

- K Nearest Neighbor
- Row Filter: Accuracy
- Column Filter: Accuracy
- Constant Value Column: Name von Modell
- RowID: erstellt RowID
- GroupBy: Mittelwert und Abweichung

Nach dem Training des K Nearest Neighbor-Modells werden verschiedene Transformationsschritte durchgeführt, um die Ergebnisse zu analysieren und aufzubereiten.

- Row Filter: Accuracy

Der Row Filter-Knoten filtert die Ergebnisse basierend auf der Genauigkeit des Modells. Dies ermöglicht es, nur die relevanten Datensätze zu behalten.

- Column Filter: Accuracy

Der Column Filter-Knoten selektiert die Genauigkeitswerte, um sie weiter zu analysieren und zu visualisieren.

- Constant Value Column: Name von Modell

Ein konstanter Wert wird hinzugefügt, um das Modell zu identifizieren, zu dem die Ergebnisse gehören.

- RowID: erstellt RowID

Die RowID wird erstellt, um jeden Datensatz eindeutig zu identifizieren.

- GroupBy: Mittelwert und Abweichung

Die Gruppierung erfolgt basierend auf dem Modellnamen, und der Mittelwert sowie die Standardabweichung der Genauigkeitswerte werden berechnet.

- **Random Forest**
- Row Filter: Accuracy
- Column Filter: Accuracy
- Constant Value Column: Name von Modell
- RowID: erstellt RowID
- GroupBy: Mittelwert und Abweichung

Für das Random Forest-Modell werden ähnliche Schritte wie beim K Nearest Neighbor-Modell durchgeführt, um die Ergebnisse zu analysieren und aufzubereiten.

Baseline

- Value Counter: Über & Unter 50k Einkommen
- Row Filter: Count
- Math Formula: Baseline Mehrheit/Gesamt
- Column Filter: Accuracy
- Constant Value Column: Name von Modell
- RowID: erstellt RowID
- Column Renamer: fehlende Spalte umbenennen

Die Baseline-Analyse wird ebenfalls aufbereitet, um sie mit den Ergebnissen der

Klassifizierungsmodelle zu vergleichen.

- Value Counter: Über & Unter 50k Einkommen

Dieser Knoten zählt erneut die Anzahl der Datensätze über und unter 50K Einkommen, um eine Vergleichsbasis zu schaffen.

- Row Filter: Count

Es wird eine Filterung nach der Anzahl der Datensätze durchgeführt, um nur die relevanten

Informationen zu behalten.

Die Baseline-Genauigkeit wird erneut berechnet, um sie mit den Genauigkeiten der Modelle zu vergleichen.

- Column Filter: Accuracy

Die Genauigkeitswerte der Baseline werden extrahiert, um sie in den Vergleich einzubeziehen.

- Constant Value Column: Name von Modell

Ein konstanter Wert wird hinzugefügt, um die Baseline eindeutig zu identifizieren

- RowID: erstellt RowID

Eine RowID wird erstellt, um die Baseline-Daten eindeutig zu identifizieren.

- Column Renamer: fehlende Spalte umbenennen

Der Column Renamer wird genutzt, um eine fehlende Spalte zu benennen und den Datensatz zuvervollständigen.

Gradient Boosted Trees

- Row Filter: Accuracy
- Column Filter: Accuracy
- Constant Value Column: Name von Modell
- RowID: erstellt RowID
- GroupBy: Mittelwert und Abweichung

Für das Gradient Boosted Trees-Modell werden ebenfalls ähnliche Schritte durchgeführt, um die Ergebnisse zu analysieren und aufzubereiten.

Naive Bayes

- Row Filter: Accuracy
- Column Filter: Accuracy
- Constant Value Column: Name von Modell
- RowID: erstellt RowID
- GroupBy: Mittelwert und Abweichung

Für das Naive Bayes-Modell werden ebenfalls ähnliche Schritte durchgeführt, um die Ergebnisse zu analysieren und aufzubereiten.

Ergebnisse Zusammenführen

- Concatenate: Modellvergleich
- Concatenate: Accuracy Vergleich
- Joiner: Full Join
- Table View: Ergebnisse

Die Ergebnisse der verschiedenen Modelle, der Baseline und anderer relevanten Daten werden nun zusammengeführt, um einen umfassenden Vergleich zu ermöglichen. Dies beinhaltet die Kombination von Modellinformationen, Genauigkeitswerten und anderen relevanten Metriken.

- Concatenate: Modellvergleich

Die Modelle werden in einem einzigen Datensatz zusammengeführt, um einen direkten Vergleich zu ermöglichen.

- Concatenate: Accuracy Vergleich

Die Genauigkeitswerte der Modelle und der Baseline werden kombiniert, um eine übersichtliche Vergleichstabelle zu erstellen.

- Joiner: Full Join

Ein Full Join wird durchgeführt, um sicherzustellen, dass alle Informationen aus den verschiedenenQuellen vollständig integriert werden.

- Table View: Ergebnisse

Die kombinierten Ergebnisse werden in einer tabellarischen Ansicht angezeigt, um eine schnelle und klare Übersicht zu ermög lichen.

- Column Filter: Vorfilterung

Um die Ansicht der kombinierten Ergebnisse zu optimieren, wird ein Column Filter-Knoten

verwendet, um nur die relevanten Spalten anzuzeigen.

Ergebnisse speichern

- CSV Writer: Modellergebnisse in CSV-Datei speichern

Die finalen, aufbereiteten Modellergebnisse werden in einer CSV-Datei gespeichert, um sie für weitere Analysen, Präsentationen oder Berichte verfügbar zu machen.

Daten einlesen

- CSV Reader: Modellergebnisse anzeigen

Die gespeicherten Modellergebnisse werden mithilfe des CSV Reader-Knotens wieder eingelesen und können nun im KNIME-Workflow analysiert oder visualisiert werden.

Ergebnisse (roh)

- Data Explorer: Modellergebnisse Übersicht

Der Data Explorer-Knoten wird genutzt, um eine Übersicht über die rohen Modellergebnisse zu erhalten und eine erste Einsicht in die gesammelten Daten zu ermöglichen.

Ergebnisse aufbereiten

- Round Double: Auf 2 Stellen
- Column Filter: Nicht gerundete Werte
- Column Manager: Accuracy
- Column Filter: Gefilterte Accuracy
- Cell Splitter: RowID (Dopplungen)
- Column Filter: RowID’s aussortiert
- Column Renamer: RowID zu Modell
- Column Resorter: Modell
- Timestamp Generator: Zeitstempe

Verschiedene Schritte werden durchgeführt, um die rohen Modellergebnisse aufzubereiten, zu filtern und zu strukturieren. Dies beinhaltet das Runden von numerischen Werten, das Filtern von nicht gerundeten Werten, die Verwaltung von Genauigkeitswerten, das Entfernen von Duplikaten durch den Cell Splitter, das Aussortieren von RowID's, die Umbenennung von Spalten und die Strukturierung nach Modellen.

Ergebnisse Final

- CSV Writer: Modellergebnisse speichern

Die finalen, aufbereiteten Modellergebnisse werden in einer CSV-Datei gespeichert. Dies ermöglicht eine einfache Weitergabe der Ergebnisse, Integration in andere Analysewerkzeuge oder die

Erstellung von Berichten.

Kurze Erklärung der vier Modelle und ihre Auswahl:

Im Rahmen der Einkommensklassifikation haben wir uns für die Anwendung von vier Machine Learning (ML)-Modellen entschieden:

- **K Nearest Neighbor (kNN)**
- **Random Forest, Gradient Boosted Trees**
- **Naive Bayes**

Jedes Modell bietet spezifische Vorzüge, die auf die Anforderungen unserer Aufgabe zugeschnitten sind.

K Nearest Neighbor (kNN):

Beschreibung:

KNN ist ein einfacher, aber leistungsfähiger Algorithmus, der auf der

Idee basiert, dass ähnliche Datensätze in der Regel die gleiche Klassifikation haben.

Er eignet sich besonders für die Identifizierung von Mustern in den Daten.

Begründung:

Aufgrund der Annahme, dass ähnliche Einkommensklassen nahe

beieinander liegen, kann kNN gut geeignet sein, um lokale Muster zu erkennen und

präzise Vorhersagen zu machen.

Random Forest:

Beschreibung:
Random Forest ist ein Ensemble-Lernalgorithmus, der auf Entscheidungsbäumen basiert. Er erstellt mehrere Bäume und kombiniert ihre Vorhersagen, was zu einer robusten und genauen Klassifikation führt.

Begründung:
Random Forest eignet sich gut für komplexe Datenstrukturen und bietet eine hohe Genauigkeit. Durch die Kombination mehrerer Bäume wird Overfitting reduziert.

Gradient Boosted Trees:

Beschreibung: Gradiant Boosted Trees ist ein weiterer Ensemble-Lernalgorithmus, der Entscheidungsbäume sequenziell erstellt. Jeder Baum korrigiert die Fehler des vorherigen Baumes, was zu einem starken Modell führt.

Begründung:
Dieser Algorithmus ist effektiv bei der Handhabung komplexer Beziehungen in den Daten. Er bietet eine hohe Genauigkeit und ist besonders nützlich, wenn präzise Vorhersagen erforderlich sind.

Naive Bayes:

Beschreibung:
Naive Bayes basiert auf dem Bayes'schen Theorem und geht von der naiven Annahme aus, dass Merkmale unabhängig voneinander sind. Dieser Ansatz eignet sich gut für einfache Klassifizierungsaufgaben.

Begründung:

Naive Bayes ist schnell zu implementieren und funktioniert gut

mit großen Datensätzen. Es ist besonders effizient, wenn die

Unabhängigkeitsannahme für die Merkmale erfüllt ist.

Entscheidungsgrundlage:

Die Auswahl der Modelle erfolgte basierend auf der Komplexität der Daten, der gewünschten Genauigkeit und der Interpretierbarkeit der Ergebnisse. KNN wurde gewählt, um lokale Muster zu erfassen, Random Forest für komplexe Strukturen, Gradient Boosted Trees für präzise Vorhersagen und Naive Bayes für schnelle und effiziente Klassifizierung.

Insgesamt bieten diese Modelle eine ausgewogene und leistungsfähige Basis für die

Einkommensklassifikation, wobei jedes Modell seine Stärken je nach Datenstruktur und

Anforderungen der Aufgabe ausspielt.

IHK-Projekt Dokumentation: Datentransformation und PowerBI-Dashboard

1. Datentransformation in PowerBI:

Die vorab bereinigten Daten werden im CSV-Format in das PowerBI-Programm eingelesen und zur weiteren Transformation/Bearbeitung weitergeleitet. In diesem Prozess werden englische Datenbezeichnungen ins Deutsche übersetzt, ungünstige Wert-Bezeichnungen gekürzt oder modifiziert, und Datentypen einiger Spalten optimiert, um eine effiziente Datennutzung sicherzustellen.

Die Datentransformation stellt den ersten Schritt in der Arbeit mit PowerBI dar, da sie zur Vereinheitlichung der Daten und ihrer kompakten Darstellung dient.

2. Dashboard-Seite: TickTockCulture – Customer Board

Das Customer Board veranschaulicht Unternehmensdaten mit Fokus auf ein zielerreichendes Kundenprofil mit einem Einkommen über $50.000. Nutzer können sich an relevanten Faktoren und Daten orientieren, um ein für sie relevantes Kundenprofil zu erstellen und so ihre Umsätze weltweit zu steigern.

3. Gruppiertes Säulendiagramm:

Auf der linken Seite des Dashboards befindet sich ein gruppiertes Säulendiagramm mit den Variablen "Bildung" auf der X-Achse und "Anzahl von Alter" auf der Y-Achse. Die Bildungsvariable ist mit einem TopN-Filter versehen, der nur die obersten fünf Kategorien zeigt. Die Legende enthält die Variable "Einkommen" mit einem Filter, der nur Daten über $50.000 anzeigt.

Dieses Visual wurde gewählt, um eine Schlüsselvariable anschaulich darzustellen und einen genauen Überblick zu bieten. Es ist einfach zu verstehen und zu interpretieren.

4. Karten-Visual 1:

In der Mitte oben gibt es ein Karten-Visual, das die Anzahl der Personen mit einem Einkommen über $50.000 zeigt. Hier wird ebenfalls die Variable "Einkommen" verwendet und grafisch dargestellt. Dieses Visual bietet einen einfachen Überblick und ermöglicht einen präzisen Einblick in die Menge der ausgewählten Personen.

5. Kreisdiagramm:

Direkt darunter befindet sich ein Kreisdiagramm, das die prozentuale Verteilung der Geschlechter mit einem Einkommen über $50.000 darstellt. Die gleichen Filtereinstellungen wie beim Säulendiagramm werden verwendet. Diese Diagrammart eignet sich besonders gut für Verteilungsdarstellungen in einem verständlichen Format.

6. Ringdiagramm:

Weiter unten gibt es ein Ringdiagramm, das die Altersgruppen im Verhältnis zum Einkommen über $50.000 zeigt. Auch hier wurden die oben genannten Filtereinstellungen verwendet. Dieses Visual wurde gewählt, um die gleichen Gründe wie zuvor zu erfüllen.

7. Filter und Wiederherstellung:

Im rechten Teil des Dashboards befinden sich verschiedene Filtereinstellungen, ein weiteres Karten-Visual und ein Button zur Wiederherstellung der Ursprungsfassung des Dashboards.

8. Karten-Visual 2:

In diesem Visual wird das durchschnittliche Alter der Personen aus dem Datensatz berechnet und angezeigt. Die Variable "Alter" wird verwendet und der Mittelwert grafisch dargestellt. Auch hier dient das Visual der Klarheit und einem umfassenden Überblick über die verfügbaren Daten.

9. Filter-Visual (Familienstand, Bildung, Arbeitsklasse, Geburtsland):

Es gibt separate Filter-Visuals für die kategorischen Werte der Variablen "Familienstand", "Bildung", "Arbeitsklasse" und "Geburtsland". Diese dienen dazu, die restlichen Visuals nach ihren Werten zu filtern und dem Betrachter einen klaren Überblick über den Zusammenhang zwischen diesen Variablen und den restlichen zu geben.

10. Button (Datenschnitte löschen):

Dieser Knopf ermöglicht das Zurücksetzen der genutzten Filtereinstellungen, um die ursprüngliche Form des Dashboards leicht wiederherzustellen. Dies ist besonders praktisch, um bei Verlust des Überblicks über die Filternutzung einen einfachen und unkomplizierten Wiederherstellungsprozess zu initiieren.

11. Button (Wichtige Einflussfaktoren):

Über diesen interaktiven Knopf gelangt der Nutzer auf die zweite Seite des Dashboards, wo er eine Übersicht über die wichtigsten Einflussfaktoren und weitere interaktive Funktionen findet.

12. Wichtige-Einflussfaktoren-Seite:

Diese Seite bietet eine erweiterte Analyse des Kundenprofils und interessante Einblicke in die Zusammenhänge verschiedener Variablen und deren Einfluss auf die Einkommensvariable. Hier kann der Nutzer interaktiv stöbern und sich passende Säulendiagramme erstellen lassen, die den Einfluss der Variablen auf das Einkommen genau erklären.

13. Buttons (Zurück & Datenschnitt löschen):

Auf dieser Seite befinden sich zwei interaktive Knöpfe, die dem Nutzer die Rückkehr zur Dashboard-Seite oder die Wiederherstellung der Ursprungsfassung des Datenschnitts ermöglichen.