# Beat the Bookie – TripleA-DWH

**Andre Dörr, Trevisto AG**

***Sportwetten sind in den letzten Jahren zu einem Milliardengeschäft geworden. Es vergeht kaum eine Werbepause während einer Sportübertragung, in der kein TV-Spot eines Wettanbieters zu sehen ist. Doch ist es überhaupt möglich, mit Sportwetten Geld zu verdienen?***

Dieser Artikel zeigt am Beispiel der Vorhersage von Fußball-Ergebnissen, wie auf Basis des Architektur-Konzepts „TripleA-DWH" (Advanced-Agile-Analytical-DWH) ein Predictive System aufgebaut werden kann. Die Grundidee des Konzepts beruht auf der Verschmelzung von Data Vault 2.0 und Predictive Analytics.

Mit normalem Fachwissen ist es nicht möglich, systematisch Geld mit Sportwetten zu verdienen. Die Quoten der Buchmacher sind sehr genau und werden schnell angepasst, da heutzutage so gut wie jede Information im Internet verfügbar ist. Um gegenüber dem Buchmacher im Vorteil zu sein, muss man sich mit Statistiken und Vorhersagemodellen beschäftigen. Ab diesem Zeitpunkt bewegt man sich auf dem Gebiet von Predictive Analytics.

Predictive Analytics bedeutet unter anderem, unterschiedliche Vorhersagemodelle zu entwickeln, zu testen und auszuführen. Idealerweise wird dies durch eine gute System-Architektur unterstützt. Data Vault 2.0 löst aktuell klassische DWH-Architekturen (Inmon, Kimball) ab, da es einige Nachteile dieser Ansätze ausgleichen kann. Verbindet man nun die beiden Themengebiete Data Vault 2.0 und Predictive Analytics, ergibt sich ein Architektur-Konzept, das man als „TripleA DWH" (Advanced Agile Analytical) bezeichnen kann. Im Folgenden wird zunächst erläutert, wie der Aufbau dieser Architektur aussieht und worin genau die Vorteile liegen. Darauf aufbauend wird anhand eines Beispiels für ein Vorhersagemodell von Fußball-Ergebnissen die Arbeitsweise mit solch einer Architektur aufgezeigt.

## Architektur-Konzept

Die komplette Architektur basiert auf der Data-Vault-2.0-Referenz-Architektur und sie besteht aus vier Schichten *(siehe Abbildung 1)*. Der Stage Layer übernimmt die gleichen Funktionen wie bei klassischen DWH-Architekturen. Er dient als temporärer Zwischenspeicher innerhalb des Systems, bevor die Daten in die nächsten Schichten verarbei-

tet werden. An dieser Stelle können zum Beispiel bereits Datentyp-Überprüfungen durchgeführt werden. Danach werden die Daten in den Raw-Data-Layer übertragen. Diese Schicht wird zum dauerhaften, integrierten und historisierten Speichern aller Rohdaten verwendet. Damit bildet sie den „Single Point of Facts". Aufbauend darauf werden die Rohdaten im Analytical Layer mit weiteren Informationen – den Features und Ergebnissen der Vorhersagemodelle – angereichert. Bei den Features handelt es sich um die Variablen und Prädiktoren, die für eine Vorhersage benötigt werden. Im Information Layer werden die Daten für abnehmende Systeme aufbereitet. Die Ergebnisse verschiedener Modelle können beispielsweise in Berichten miteinander kombiniert werden, oder es werden beispielsweise dimensionale Datenmodelle für BI-Tools zur Verfügung gestellt.

## Agile

Für den Raw Data Layer und den Analytical Layer werden die Datenmodellierungstechniken von Data Vault 2.0 genutzt. Ein entscheidendes Charakteristikum der Data-Vault-Modellierung ist die Trennung der Daten in Objekte („Hubs"), Beziehungen („Links") und Kontexte („Satelliten"). Dadurch besitzt Data Vault Eigenschaften, die eine agile Entwicklung innerhalb eines DWH ermöglichen:

- *Pattern Based Loading*
  Data Vault kennt nur drei verschiedene Typen von Tabellen: Hubs, Links und Satelliten. Jeder Typ besitzt die gleiche Grundstruktur. Die Ladeverfahren aller Typen sind standardisiert. Diese Standardisierung ermöglicht eine automatisierte Generierung sowohl der Data-Vault-Strukturen als auch der benötigten ELT-Prozesse zur Beladung eines Datenmodells.

- *Zero Impact*
  Data Vault verfolgt einen harten Zero-Impact-Ansatz. Dies bedeutet, dass die Anbindung neuer Datenquellen oder die Erweiterungen bestehender Datenquellen keinen Einfluss auf existierende Strukturen und Prozesse haben. Neue Vorhersagemodelle können beispielweise dem Analytical Layer hinzugefügt werden, indem sie als neuer Kontext mit dem entsprechenden Objekt verbunden werden. Dies sorgt nicht nur dafür, dass Erweiterungen eines bestehenden Modells extrem flexibel sind, sondern auch dafür, dass sogar Regressionstests entfallen können.

- *Sandbox Prototyping*
  Die Entwicklung und Optimierung von Vorhersagemodellen ist eine komplexe Aufgabe, die in der Regel mithilfe von separaten Tools (etwa R-Studio) durchgeführt wird. Die Einführung eines Sandbox Prototyping bietet hier die Möglichkeit, dieses Vorgehen in einen agilen Prozess zu überführen. Dabei wird wieder die Flexibilität eines Data-Vault-Datenmodells genutzt. Dem Endanwender wird eine vom restlichen Verarbeitungsprozess gekapselte Sandbox innerhalb der Datenbank zur Verfügung gestellt. Features und Ergebnisse von Vorhersagemodelle, die sich in der Entwicklung befinden, lassen sich als neuer Kontext (Satelliten) innerhalb dieser Sandbox abspeichern. Eine Integration in die bestehenden Daten findet dabei automatisch statt. So können bestehende und neue Vorhersagemodelle miteinander verglichen werden. Erweist sich ein Vorhersagemodell als effektiv, wird es im Anschluss in den Standardverarbeitungsprozess integriert.

## Analytical

Viele Datenbankhersteller (wie Oracle, Microsoft, Exasol) haben mittlerweile das Potenzial und die Relevanz von statistischen Programmiersprachen erkannt und bieten an, die Option „R Code" innerhalb der Datenbank auszuführen. Dies bedeutet architektonische Vorteile für den Aufbau eines Predictive Systems.

Bisher wurden die Berechnungen der Vorhersagemodelle meist auf externen Servern durchgeführt. Dazu war es nötig, alle erforderlichen Features zunächst zu exportieren und die Ergebnisse der Vorhersage danach wieder zu importieren. Diese Schritte können nun komplett entfallen. Die Ausführung der Vorhersagemodelle kann direkt auf den Daten innerhalb der Datenbank erfolgen. Dies verringert die Komplexität der Architektur und der Verarbeitungsprozesse eines Predictive Systems.

## Advanced

Wie bereits erwähnt, bildet Data Vault 2.0 die Grundlage dieses Architektur-Konzepts. Dessen Datenmodellierungsmethoden beinhalten eine entscheidende Verbesserung gegenüber der Version 1.0: Es werden Hash-Keys statt künstlicher Schlüssel für die eindeutige Identifizierung eines Objekts genutzt. Die Identifizierung durch Hash-Keys erlaubt es, ein Data-Vault-Modell über mehrere Technologien hinweg zu modellieren und zu implementieren. Dadurch besteht die Möglichkeit, in der Gesamt-Architektur ein relationales DBMS und NoSQL-Technologie zu kombinieren *(siehe Abbildung 2)*.

Damit ergeben sich Potenziale für verschiedenartige Use Cases. Im einfachsten Fall könnte eine NoSQL-Technologie als reiner Stage Layer genutzt werden. So lässt sich eine einfachere Schnittstellen-Anbindung durch das Nutzen von Schema-on-Read umsetzen. Eine weitere Möglichkeit ist das effektive Auswerten von unstrukturierten Daten. Dabei werden die unstrukturierten Informationen in einer NoSQL-Technologie gespeichert. Die

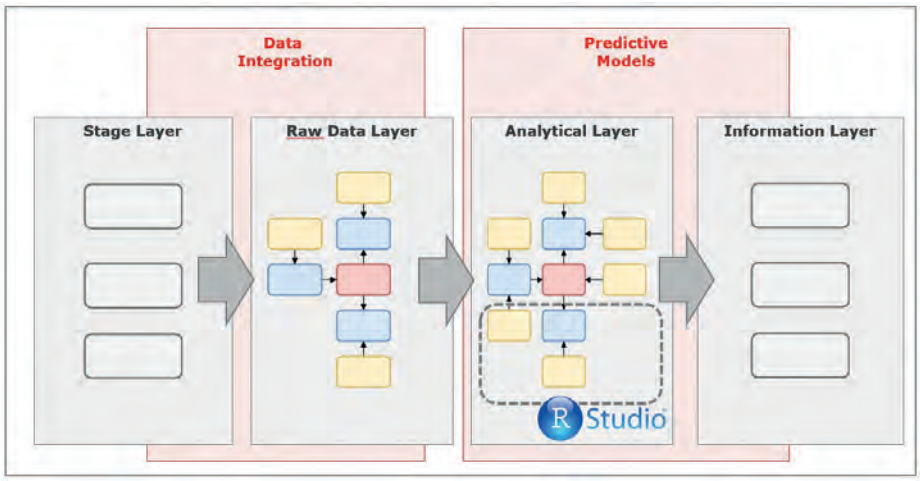


*Abbildung 1: Die Layer-Übersicht*

strukturierten Informationen liegen weiterhin in einem relationalem DBMS. Da die Datenmodellierung jedoch übergreifend über beide Technologien stattfindet, können strukturierte und unstrukturierte Daten gemeinsam ausgewertet werden.

Dieses Architektur-Konzept kann als Grundlage für diverse Predictive Systeme genutzt werden. Im Folgenden wird am Beispiel der Entwicklung eines Vorhersagemodells für Fußball-Ergebnisse aufgezeigt, wie sich auf Basis dieser Architektur Modelle entwickeln, validieren und implementieren lassen.

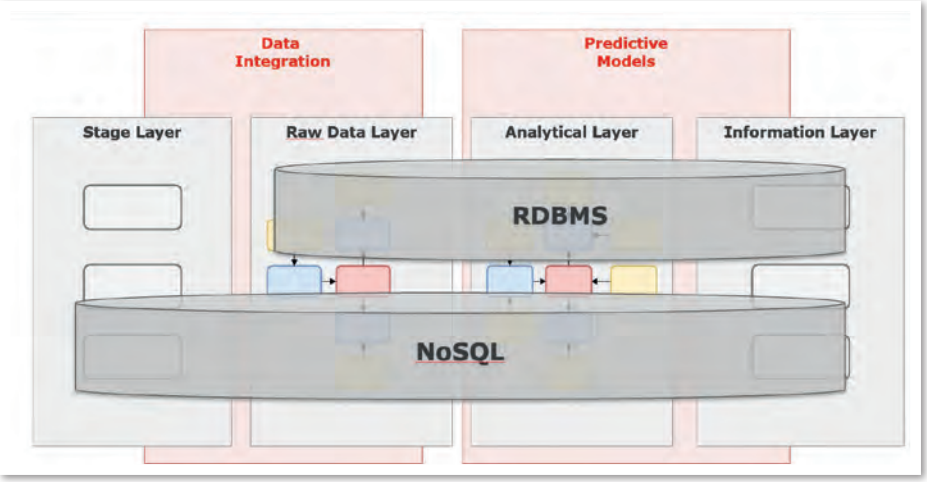


*Abbildung 2: Integration SQL und NoSQL*

## Vorhersagemodell für Fußball-Ergebnisse

Es existieren verschiedenste grafische Darstellungen für den Entwicklungsprozess eines Vorhersagemodells. Alle Darstellungen haben jedoch gemeinsam, dass sie aus ähnlichen Einzelschritten bestehen. Auf Basis der Problem- und Ziel-Definition wird das Vorhersagemodell entwickelt und optimiert, bis das gewünschte Ergebnis erreicht ist. Danach kann das Modell im produktiven Einsatz genutzt werden *(siehe Abbildung 3)*.

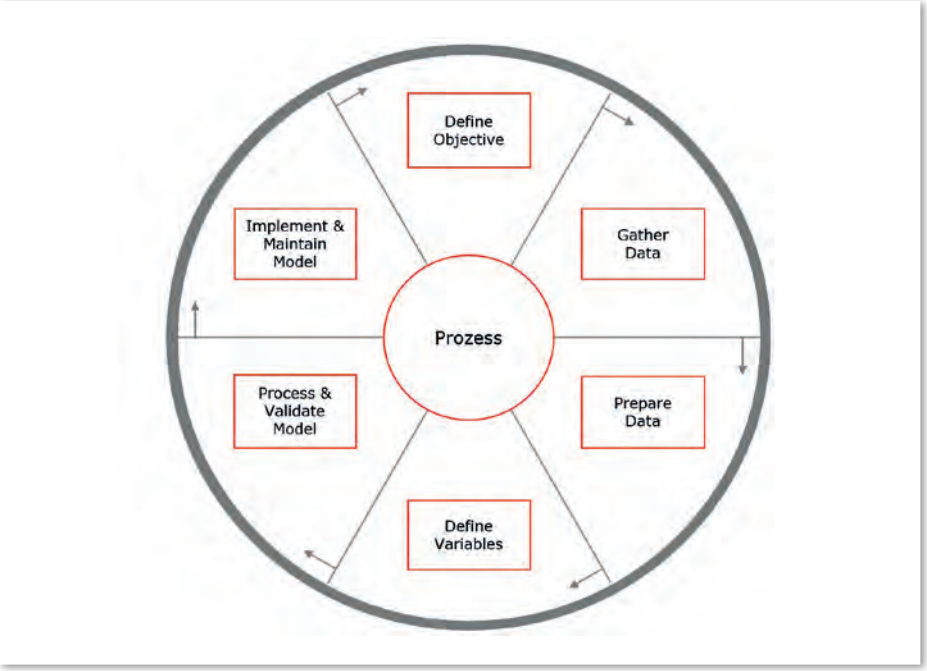


*Abbildung 3: Entwicklungsprozess für Vorhersagemodelle*

Der erste Schritt besteht darin, Problemstellung und Ziel genau zu definieren („Define Objective"). Dies ist notwendig, um darüber Klarheit zu erlangen, um was für ein Vorhersage-Problem es sich handelt. Regressionsprobleme erfordern beispielsweise andere Methoden als Klassifikationsprobleme. Bei der Vorhersage von Fußball-Ergebnissen (Heimsieg, Unentschieden, Auswärtssieg) handelt es sich um ein typisches Klassifikationsproblem. Einen Ansatz für dieses Klassifikationsproblem lieferten Dixon & Coles (Modelling Associated Football Scores and Ineffieciences in the Football Market, 1995). Sie beschreiben die Anzahl der Tore in einem Fußballspiel als eine Poisson-Verteilung *(siehe Abbildung 4)*.

Die Poisson-Verteilung ist eine diskrete Wahrscheinlichkeitsverteilung für eine bestimmte Anzahl von unabhängigen Ereignissen in einem festen Zeitintervall mit konstantem Mittelwert. Für die Berechnung werden nur zwei Parameter benötigt: der erwartete Mittelwert („λ") der Verteilung und die Anzahl der Ereignisse („X"). In den Bundesliga-Saisons von 2011 bis 2016 sind im Durchschnitt pro Spiel 2,89 Tore gefallen. *Abbildung 5* zeigt die reale Verteilung der Tore. Ist die Wahrscheinlichkeit für eine gewisse Anzahl von Toren bekannt, die das Heim- und das Auswärtsteam schießt, kann damit auch die Wahrscheinlichkeit für Heimsieg, Unentschieden oder Auswärtssieg berechnet werden.

$$P(X = x) = \frac{\lambda^x e^{-\lambda}}{x!}$$

*Abbildung 4: Formel für Poisson-Verteilung*

Nachdem die Problemstellung und das Ziel genauer definiert sind, besteht die nächste Aufgabe darin, Datenquellen zu suchen, die die benötigten Daten für eine Vorhersage liefern („Gather Data"). Dies können externe und interne Datenquellen sein. Das Internet stellt dabei natürlich die größte Quelle dar. Einige Firmen haben ihr gesamtes Geschäftsmodell auf das Sammeln und Verkaufen von Daten ausgelegt. In Bezug auf Fußball-Daten ist hier zum Beispiel Opta zu nennen. Im vorliegenden Beispiel wurde die Internet-Seite „football-data.co.uk" genutzt, die historische Fußball-Statistiken für mehr als zwanzig Ligen und bis zu zwanzig Jahren bereitstellt. Da das Ergebnis eines Vorhersagemodells stark von der Qualität der Daten abhängig ist, müssen diese Daten vor der Verarbeitung geprüft und notfalls korrigiert werden („Prepare Data"). Sind die Daten vollständig? Gibt es Lücken in den Daten? Sollten diese Lücken gefüllt werden? Existieren Ausreißer, die das Er-

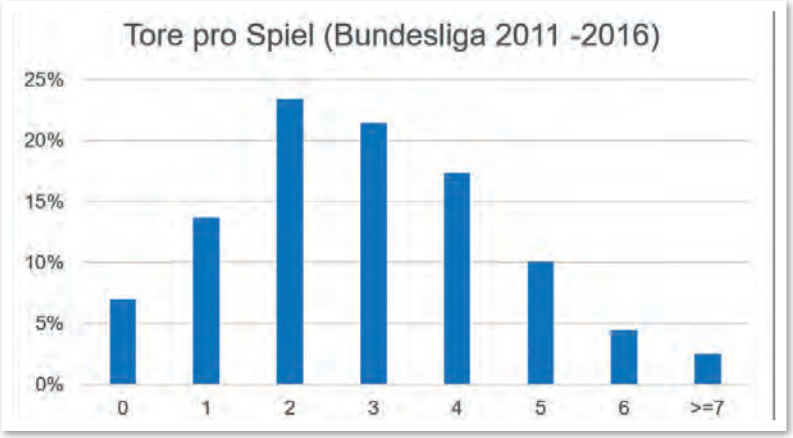


*Abbildung 5: Torverteilung Bundesliga (2011 – 2016)*

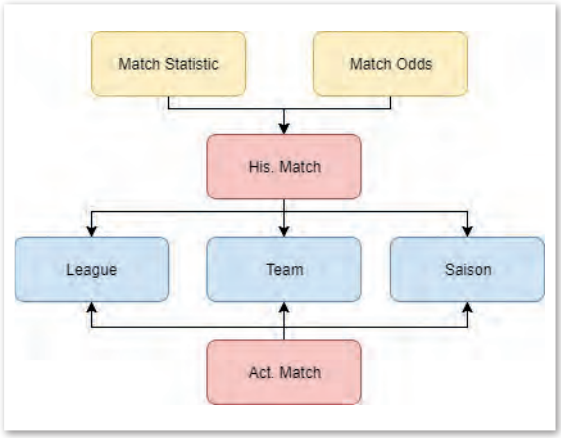


*Abbildung 6: Raw-Layer-Data-Vault-Modell*

gebnis verfälschen können? All dies sind Faktoren, die Einfluss auf die Genauigkeit eines Modells haben.

Bei der Implementierung eines Vorhersagemodells auf der TripleA-DWH-Architektur ist jedoch noch ein weiterer Schritt notwendig. Die verschiedenen Datenquellen sind in ein Data-Vault-Modell zu überführen, um die Flexibilität der Architektur für den weiteren Entwicklungsprozess nutzen zu können. *Abbildung 6* zeigt das entstehende Data-Vault-Modell für die verwendeten Rohdaten.

Die Objekte „Liga", „Team" und „Saison" sind als Hubs (blau) definiert. Ein Match stellt die Beziehung (rot) zwischen den drei Objekten dar – zwei Teams spielen in einer Liga in einer Saison zu einem gewissen Zeitpunkt gegeneinander. Diese werden unterschieden in historische Spiele und aktuelle Spiele. Die historischen Spiele dienen dem Simulieren von Vorhersagemodellen. Auf die aktuellen Spiele müssen die entwickelten Modelle angewandt werden. Für die historischen Spiele existieren zwei verschiedene Kontexte (gelb) – die Spiel-Statistik und die Wettquoten bei verschiedenen Wettanbietern. Dieses Datenmodell bildet die Basis für die weiteren Schritte und wird nach und nach erweitert.

Einer der wichtigsten Schritte vor der Erstellung eines Vorhersagemodells ist die Feature- oder auch Variablen-Selektion („Define Variables"). Die Verwendung schlechter Variablen für ein Vorhersagemodell führt auch zu schlechten Ergebnissen. Dabei sollte das Motto „weniger ist manchmal mehr" beachtet werden. Zu viele Variable können zu einem Overfitting führen oder das Modell unnötig verkomplizieren. Bei Machine-Learning-Algorithmen verlängert sich die Trainingszeit mit der Anzahl der Variablen.

Ein Beispiel für ein schlechtes Feature bei einer Fußball-Vorhersage ist der Ballbesitz, wie das Champions-League-Finale 2012 zwischen Bayern München und Chelsea London gezeigt hat. Für die Vorhersage von Fußball-Ergebnissen mit der Poisson-Verteilung werden Variablen benötigt, welche es ermöglichen, die erwartete Anzahl von Toren für die Heim- und die Auswärtsmannschaft zu berechnen.

In dem Vorhersagemodell von Dixon & Coles werden dafür die Angriffs- und Verteidigungsstärke der jeweiligen Mannschaften genutzt. Diese Variablen repräsentieren das Verhältnis der Anzahl der Tore beziehungsweise Gegentore eines Teams zum Liga-Durchschnitt. Wird mithilfe dieser Variablen beispielsweise die erwartete Tore-Anzahl für das Spiel von Bayern München gegen Schalke 04 am 4. Februar 2017 berechnet, ergeben sich für die Heimmannschaft 2,57 und für die Auswärtsmannschaft 0,35 erwartete Tore. *Abbildung 7* zeigt die Wahrscheinlichkeitsverteilung für die Heim- und die Auswärtsmannschaft. Das Spiel endete 3:0, dem Ergebnis mit der zweithöchsten Wahrscheinlichkeit.

Um die Simulation eines Vorhersagemodells durchzuführen, sind die Variablen für alle verfügbaren historischen Daten zu berechnen. *Abbildung 8* zeigt das um den Satelliten „Attack Defence Strength" erweiterte Datenmodell.

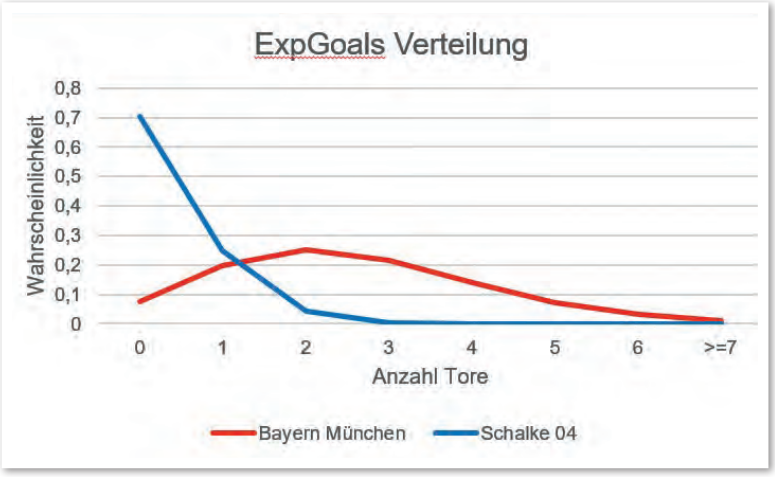


*Abbildung 7: Verteilung erwartete Tore Bayern München gegen Schalke 04 (4. Februar 2017)*

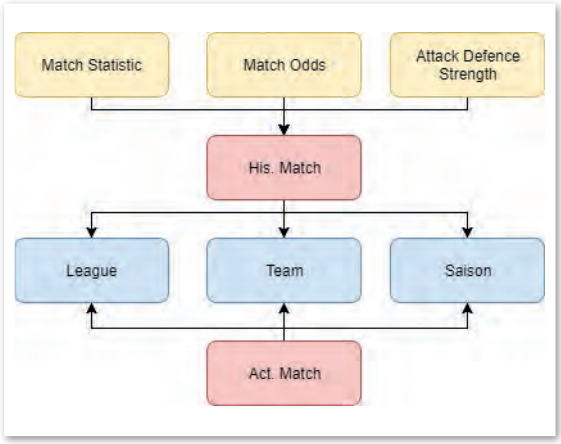


*Abbildung 8: Analytical-Layer-Data-Vault-Modell mit Feature-Berechnung*

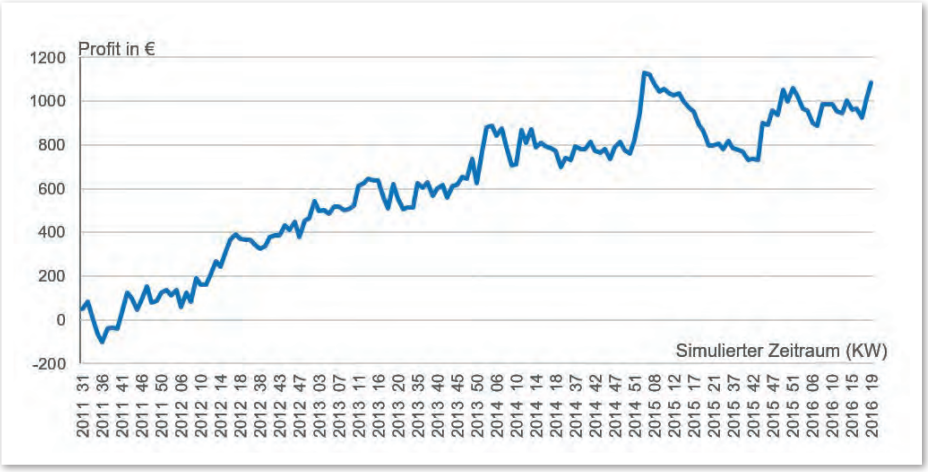


*Abbildung 9: Modell-Simulation für Bundesliga 2011-2016*

Hier zeigt sich die Flexibilität eines Data-Vault-Datenmodells. Neue Features werden einfach als neuer Kontext („Satellit") in das Datenmodell integriert, ohne dass bestehende Strukturen und Prozesse angepasst werden müssen. Genauso verhält es sich, wenn Features nicht mehr nötig sind. Dann wird der entsprechende Satellit einfach wieder entfernt.

Nachdem die historischen Features zu Verfügung stehen, kann das Vorhersagemodell unter Verwendung des Sandbox Prototyping getestet und optimiert werden („Process & Validate Model"). Der Optimierungsprozess unterscheidet sich dabei abhängig von der verwendeten Vorhersagemethode. Eine lineare Regression kann beispielsweise durch das Verwenden von Polynomen der Features oder das Wechseln auf eine robuste lineare Regression optimiert werden. Für die Optimierung eines Modells ist stets eine Analyse der Gründe einer schlechten Vorhersage notwendig.

Auch die Vorhersage von Fußball-Ergebnissen mithilfe der Poisson-Verteilung besitzt einige Nachteile, die ausgeglichen werden müssen. Im Vergleich zu den reellen Torverteilungen ist die vorhergesagte Wahrscheinlichkeit für null Tore zu gering. Dies kann durch das Nutzen einer Zero-Inflated-Poisson-Verteilung verbessert werden. Zudem ist die Wahrscheinlichkeit für Unentschieden zu niedrig. Werden jedoch die historischen Unentschieden-Statistiken als Korrekturfaktor für das Modell genutzt, kann auch dieser Nachteil ausgeglichen werden.

*Abbildung 9* zeigt die Simulation des optimierten Vorhersagemodells gegen die Wettquoten des Buchmachers Bet365. Auf Basis der vorhergesagten Wahrscheinlichkeiten werden unterbewertete Wetten identifiziert und auf diese gesetzt. Das Modell liefert bei einer Anzahl von mehr als 1.500 Wetten mit einem Einsatz von 10 Euro pro Wette einen Profit von 1.120 Euro (Y-Achse) über den simulierten Zeitraum (X-Achse). Das entspricht einer Verzinsung von sieben Prozent des eingesetzten Kapitals.

Hat sich der Prototyp eines Vorhersagemodells durch Tests und Simulationen als erfolgreich herausgestellt, kann das Modell in den Standard-Verarbeitungsprozess integriert werden („Implement & Maintain Model"). An dieser Stelle wird die R-Integration der verschiedenen Datenbanken genutzt. Das Vorhersagemodell oder das trainierte Modell wird ähnlich wie die berechneten Features als zusätzlicher Kontext („Satellit") in das Datenmodell integriert.

Die Flexibilität der Data-Vault-Modellierung ist auch hier ein Vorteil. Es können verschiedene Vorhersagemodelle auf einfache Art und Weise nebeneinander implementiert werden. Im Information Layer lassen sich danach Berichte erstellen, um die Ergebnisse verschiedener Vorhersagemodelle zu kombinieren und für Auswertungen zur Verfügung zu stellen.

*Abbildung 10* zeigt das erweiterte Datenmodell mit der implementierten Poisson-Vorhersage. Das Vorhersagemodell muss nur für die aktuellen Spiele berechnet werden. Daher sind die Feature-Berechnungen und die Poisson-Vorhersage als zusätzliche Satelliten für die aktuellen Spiele implementiert.

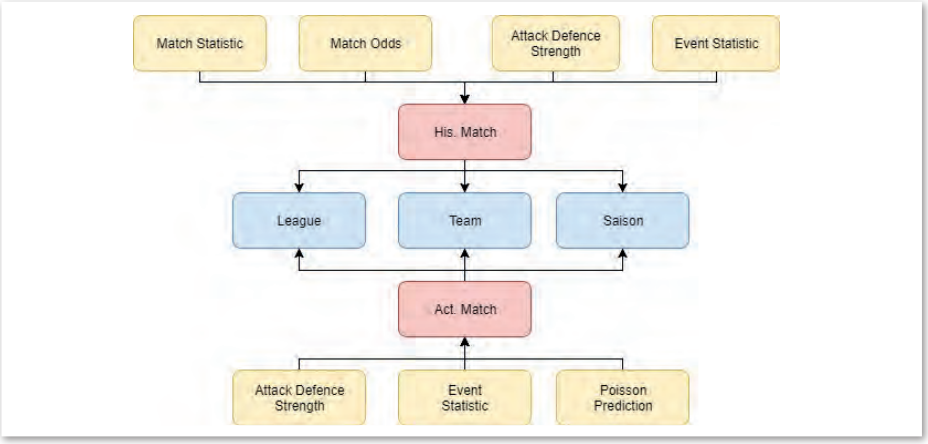


*Abbildung 10: Analytical-Layer-Data-Vault-Modell mit implementierter Vorhersage*

## Erkenntnisse

Es hat sich gezeigt, dass bereits ein sehr einfaches Vorhersagemodell reicht, um gegenüber dem Buchmacher im Vorteil zu sein. Während der Entwicklung und Implementierung des Vorhersagemodells für Fußball-Ergebnisse hat das Architektur-Konzept „TripleA DWH" klare Vorteile bei der Flexibilität und Erweiterbarkeit des Gesamtsystems gezeigt. Data Vault 2.0 bietet damit nicht nur Vorteile für Standard-DWH-Implementierungen, sondern auch für den Aufbau eines Predictive Systems. Eine Integration mehrerer Vorhersagemodelle ist ohne Problem möglich. Die Möglichkeit, R Code direkt auf den Daten auszuführen, führt dazu, dass die Gesamt-Architektur eines Predictive Systems weit weniger komplex ist als bei der Berechnung auf separaten Servern.

## Quellen


- Data Vault 2.0: *http://danlinstedt.com/#*
- „Seven Steps to Effective Predictive Modeling": *http://oliviagroup.com/training/predictive-modeling-training*
- Dixon & Coles, „Modelling Associated Football Scores and Inefficiencies in the Football Betting Market": *http://www.math.ku.dk/~rolf/teaching/thesis/DixonColes.pdf*



*Andre Dörr*
*andre.doerr@trevisto.de*