**Antonio Serrano Redondo**

**Churn Projekt**

**CODE ERKLÄRUNG: DECISSION TREE**

**Das ziel dieses Dokuments ist eine Zusammenfassung des Codes mitteilein. Themen, die wichtig über dem Code sind, auch in “Lessons learns“ besprochen werden.**

**Der Modell ist ein Decission Tree Classifier, (wir benutzen nicht Regression weil unsere Ziel Spalte ist eine Kategorie -Ja, Nein-).**

**Der Wähl liegt in die Möglichkeit, dass Decission Tree hast um mit Kategorien und Werte zu arbeiten.**

**Algorithmus Vorgehen:**

1. **Data Vorbereitung (allgemein für die Gruppe, steht in Gruppe Bericht)**
2. **Split in Training Data und Test Data. Hier wir nehmen 80% Training Data und 20% Test Data.**

**Y Spalte : Event (Mitarbeiter kündigt)**

**X Spalte : Feature Spalten**

1. **Wir trainieren die Daten und machen wir ein Prognose, um zu sehen wie gut der Algorithmus ist. Wir bekommen 1 in Trainings Data. Das bedeutet, dass unsere Modell sehr gut mit dem Trainierte Data rechnen kann, aber das bedeutet nicht, dass mit neuen Data sehr gut rechnen auch kann. Wir überprüfen es mit dem Test Data, und hier haben wir ein score 0.64.**

**Wir können das bei Decission Tree erwarten, ein Overfitting bei Training Data zu bekommen. Deswegen, um zu eigentlich prüfen, dass der Modell mit “hidden data“ gut rechnen kann, versuchen wir Cross Validierung Score in die Training Data.**

**Wir rechnen Cross Val. Score in Training Data und wir merken, dass die beste Folds Nummer ist 8, weil die Beste score hast mit sehr wenig Standard Deviation. Deswegen benutzen ab diesem Moment immer 8 Folds.**

**4. Optimierung.**

1. **Features Wichtigkeit: Wir nehmen die Spalte die unwichtig sind raus. Der Modell rechnet sie Wichtigkeit jeder Spalte (summe Spalte ist 1). Damit können wir wissen, welche Spalten raus dürfen, ohne wichtige Information zu verlieren. Wir haben gesetzt wie Grenze 2%. Das bedeutet, die Spalte deren Wichtigkeit weniger als 2% ist, werden herausgefiltert. Mit diesem Kriterium, 5 Spalten sind Weg.  
   ['coach', 'greywage', 'way', 'industry', 'gender', 'profession']**

**Wir rechnen wieder der Algoritmus mit die Spaltenraus, aber der Score hat sich nicht verbessert. Diesem Schritt der Optimierung war nicht hilfreich.**

**B) Skalierung: Hier versuchen wir die Daten ähnlicher machen, um auch die Modell vereinfachen. Wir versuchen mit wei Optionen, Standard Scale und MinMax Scale.**

**Wir führen wieder die Cross Val Score auf Training Data us. Leider ist die Skalierung auch nicht hilfreich.**

**C) Grid Search: Wir versuchen mit dem GridSearch das Modell zu verbessern durch eine Parameter Optimierung. Das Default Decission Tree Classifier hat alle Paramenter mit ein Default Option, die verändert werden können.**

**Das Ziel ist, die beste Parameter zu finden, die einen besseren Score rechnen. Mein Ansatz war mit dem Parameter mitspielen, leider wenn ich versuche verschiedene Parameter Werte zu bieten, crash den Laptop. Deswegen musste ich nur eigene Parameter mit nicht zu viele Werte setzen.  
  
Zum Beispiel bei Criterion, Falls ich die drei Optionen setzen (Gini, Entropy, Log\_loss) sodass der Algorithmus alle Kombinationen versuchen kann, bekomme ich keine Score nach 60mn.  
  
Deswegen mit weniger Parameter Optionen und sehr leicht, wird der Score nur 0.55 werden. Das bedeutet, dass die Default Parameters sind besser als die Grid Search, leider haben wir nicht gefunden, welche wären am besten.**

**Wir rechnen die “Best parameters” (die wir gerade wissen die nicht die beste sind) auf die Test Data und bekomment wir auch 0.55.**

**Der Ansatz die Parameter Optimierung war ein kleines Baum zu erstellen, ohne zu viele Nodes und Leafs. Auch die min samples möchten wir, dass nicht zu viele sind.**