Vorgehen

Setup

- Virtual Environment wurde mit Visual Studio generiert und die requirements wurden installiert
- Zusätzlich wurden andere Bibliotheken installiert wie Seaborn, Matplotlib, Sklearn…
- Zugang zu BigQuery und die Google Konsole war möglich
- ->Aber Verknüpfung mit dem Notebook hat Fehlgeschlagen
- Lösung von Part 1 wurde deshalb mit Python bearbeitet

Daten Analyse:

- Erstmal wurden die Daten analysiert, um zu schauen
- ->Ob die Daten bereinigt werden müssen oder Datentypen verändert werden müssen
- ->Ob Spalten oder Zeilen gelöscht werden müssen
- -> Oder überprüft in welchen Spalten NAs vorhanden sind und wie man die am besten füllen kann
- Mit einem Koorelationsplot wurde überprüft, welche Spalten als redundant angesehen werden können oder welche Features Relevant in Bezug zum Klassenlabel waren
- Duplicate check
- -> “Class Imbalance” Problem wurde identifiziert

Daten pre-processing:

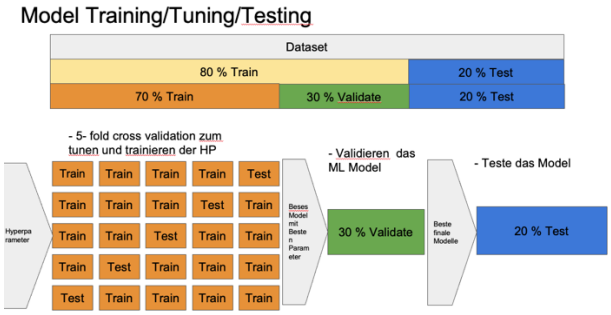
- Unrelevante Spalten wurden entfernt
- NA durch mean , mode oder median gefüllt
- Manche Zeilen mit NAs wurden entfernt
- Eine Spalten wurden neu transformiert
- -> von Numerischen wert zu einen Binären

Model trainiert,getuned validiert und getestet:

- Trainiert wurde auf allen Features und auf Feature Selektion basierten Features und verglichen (2 verschiedene Datensätze)
- „Decision tree“ wurde ausgewählt, da es einfach zum Interpretieren ist
- Daten wurden mit F1 Score , Accuracy ,ROC Konfusion-Matrix evaluiert
- -> F1 score ist sehr gut für Class Imbalance Problem
- -> Feature Selection wurde mit “tree feature importances” bestimmt

Wie wurde, trainiert ,getuned ,validiert und getestet?

1. Für den Decision tree trainieren und validieren wir mit 5 fold cross validation, um die idealen Hyperparameter(Tiefe des Baums) zu finden. Der mean der ACC und die Std. wurde für pro genutzten Parameter berechnet
2. Dann wird mit dem Idealen Parameter auf den ganzen Trainingsdatensatz trainiert und auf den Validation Datensatz validiert
3. Zum Schluss wird das Model mit dem besten Parameter genommen und mit dem Testdatensatz getestet und evaluiert



Warum Validierungs Datensatz:

- Eig. war die Idee, mehrere Algorithmen zu nehmen und diese zu tunen und dann mit dem Validierungsdatensatz, das beste Model zu nehmen und dieses dann zu testen
- -> Das würde aber den Rahmen sprengen, weshalb es doch nicht mehr weitergeführt wurde

Warum diese Methoden?:

- Cross Validation: Um Over Fitting zu vermeiden, da wir verschiede Datensätze nehmen und immer trainieren und evaluieren
- ->So nutzen wir den kompletten Datensatz für Training und Validierung der Hyperparameter
- Für jeden Hyperparamter, haben wir ein Model trainiert mit cross validation, damit wir sehen welches Hyperparamter das Beste ist
- Mit Cross Validation schauen wir, für jedes Model des Hyperparameters den mean und Std der ACC an
- Standard Deviation wurde angeschaut, um zu sehen, wie weit die vom Mittel wert ACC voneinander verstreut sind , denn eine hoher Std würde bedeuten, wir hätten verschieden gute bis schlechte Modelle bei diesen Parameter(je kleiner desto besser)
- F1 Score: Acc ist nicht gut bei unbalanced Datensatz
- ->Aber mit den F1 score kann man sehen wie viele Bsp. es falsch vorhergesagt hat

Probleme Generell

- BigQuery mit der Google Cloud Konsole konnte nicht mit dem Jupyter Notebook verbunden werden
- -> hat viel Zeit gekostet
- Zeitvariablen wurden vielleicht nicht richtig eingesetzte in mein Vorgehen
- Wenig Information vorhanden über den Datensatz, um fachlich zu sagen was relevant sein könnte von den Features