IKT-Projektdokumentation

September 23, 2024

Projekt zur Vorhersage der Länge des Menstruationszyklus von Bettina Justus

Inhalt

1 Motivation

Frauengesundheit ist ein wichtiger und oft unterschätzter Bereich der medizinischen Forschung. Besonders der Menstruationszyklus spielt eine zentrale Rolle im Leben vieler Frauen und kann von verschiedenen biologischen sowie externen Faktoren beeinflusst werden. Eine präzise Vorhersage des Menstruationszyklus könnte Frauen dabei helfen, ihre Gesundheit besser zu managen und ihre täglichen Aktivitäten besser zu planen.

In diesem Projekt habe ich mich darauf konzentriert, mithilfe von Machine Learning die Länge des Menstruationszyklus vorherzusagen. Ursprünglich lag mein Fokus auf der Krankheit Endometriose, einer chronischen Erkrankung, die weltweit 8-15% der Frauen betrifft und oft mit unvorhersehbaren „Schüben“ einhergeht. [1] Aufgrund des Mangels an öffentlich verfügbaren Datensätzen zu diesem Thema entschied ich mich, zunächst die Grundlagen der Krankheit – den Menstruationszyklus – genauer zu betrachten.

Das Ziel dieses Projekts ist es, Frauen durch eine Vorhersage ihres Zyklus eine bessere Planbarkeit zu ermöglichen. Durch die Vorhersage der Zykluslänge mittels Machine Learning soll der erste Tag des nächsten Menstruationszyklus berechnet werden.

2 Sprint 1: Datenaufbereitung

In diesem Abschnitt beschriebenen Vorgänge sind im Jupyter Notebook 1_PrepAndCleanData.ipynb festgehalten.

2.1 Datensatz

Für mein Projekt nutzte ich den Menstrual Cycle Data-Datensatz *FedCycleData071012(2).csv*, den ich auf der Platform Kaggle gefunden habe. Dieser Datensatz erschien mir als gut geeignet, da er

über 80 Spalten verfügt, von denen 67 ursprünglich als Integer-Daten deklariert waren. Mit den numerischen Daten lässt es sich gut arbeiten, besonders wenn es um Machine Learning geht. Mein Hauptziel war die Vorhersage der Zykluslänge, und der Datensatz enthielt zusätzlich eine Vielzahl von weiteren, regelmäßig im Zyklus vorkommenden Merkmalen. Insgesamt umfasst der Datensatz 1.664 Einträge, wodurch eine solide Grundlage für mein Projekt gewährleiste wurde.

2.2 Datenimport und Überblick

Nachdem ich den Datensatz geladen hatte, stellte sich jedoch heraus, dass nur 5 Spalten tatsächlich als `Integer` deklariert waren, während der Großteil der Daten als `Object` erkannt wurden. Um die Daten für die Modellierung zu vereinheitlichen, habe ich im weiteren Projektverlauf alle numerischen Spalten in den Datentyp `Float` umgewandelt.

2.3 Auswahl relevanter Merkmale

Bei der Auswahl der für mein Projekt relevanten Daten habe ich entschieden, alle Spalten zu entfernen, die demografische oder persönliche Informationen enthalten, insbesondere solche, die sich auf den männlichen Partner beziehen (Spaltennamen mit einem M am Ende). Für mein Projekt sind vor allem die körperbezogenen Daten von Bedeutung, da sie direkten Einfluss auf den Menstruationszyklus haben. Nachdem erfolgreichen entfernen irrelevanter Spalten, überprüfte ich die Datentypen der verbliebenen Spalten genauer. Wie zuvor erwähnt, wurden viele Spalten als `Object` erkannt, obwohl ausschließlich numerische Werte in den dazugehörigen Zeilen zu sehen sind. Um eine Einheitlichkeit zu gewährleisten wurden alle Spalten in den Datentyp `Float` umgewandelt, obwohl diese teilweise als `Object` oder `Integer` erkannt wurden.

2.4 Fehlenden Werte

Um den Umgang mit fehlenden Werten zu entscheiden, ließ ich die Spalten mit fehlenden Daten anzeigen und stellte fest, dass 52 Spalten betroffen waren, was fast alle Spalten umfasst. Zuvor überprüfte ich auch auf Duplikate, stellte jedoch fest, dass die angezeigten Zeilen keine Duplikate waren, da sie sich inhaltlich unterschieden. Anschließend analysierte ich die Spalten, die mehr als 50% fehlende Werte aufwiesen, sowie die mit weniger als 50%, um einen besseren Überblick zu erhalten. Obwohl das Entfernen fehlender Daten Verzerrungen vermeidet, kann es zu einem erheblichen Informationsverlust führen. Jedoch aufgrund der hohen Anzahl fehlender Werte der Spalten mit mehr als 50%, entschied ich mich für die Entfernung dieser Parameter.

2.5 Ausreißer

Anschließend analysierte ich die Ausreißer in den Daten und berechnete dafür den Prozentsatz der Streuung dieser. Um die Ergebnisse anschaulich darzustellen, visualisierte ich diesen Prozentsatz in einer Grafik. Danach entschied ich mich, die identifizierten Ausreißer durch den Mittelwert der jeweiligen Spalte zu ersetzen, um die Daten auszubessern.

2.6 Korrelation zur Zielvariable

Nachdem die wichtigsten Schritte abgeschlossen wurden, schaute ich mir die Korrelationen meiner Zielvariable, der Zykluslänge, an. Leider stellte ich fest, dass keine starke Korrelation zu den anderen Variablen bestand. Dennoch betrachtete ich die Variable mit der höchsten Korrelation zur Zykluslänge näher. Dies zeigte, dass auch hier keine signifikante Korrelation vorhanden war, was

darauf hindeutet, dass traditionelle lineare Modelle wie die lineare Regression nicht geeignet sind, um präzise Vorhersagen auf Grundlage dieses Datensatzes zu treffen.

3 Sprint 2: Erster Modellversuch

In diesem Abschnitt beschriebenen Vorgänge sind im Jupyter Notebook 2_Prediction_firstTry.ipynb festgehalten. Die Datei 1_PrepAndCleanData.ipynb bildet die Grundlage der Iteration, weshalb die Fortschritte erst weiter unten im Dokument sichtbar sind.

3.1 Modellauswahl

Da die Anwendung einer linearen Regression nicht vielversprechend war, recherchierte ich nach alternativen Modellen und kam schnell zu dem Entschluss den *Random Forest-Algorithmus* zu verwenden. Während meiner Suche stieß ich auf ein ähnliches Projekt auf Kaggle, welches in der Zwischenzeit meiner Projektarbeit erschien und einen ähnlich Ansatz, wenn auch mit unterschiedlichen Parametern verfolgt. Zusätzlich zum *Random Forest-Algorithmus* verwendet jenes Projekt das *XGBoost (Extreme Gradient Boosting)* zu Modellierung. Aus Gründen der Vergleichbarkeit entschied ich mich deshalb dazu nebst des initialen *Random Forest* Ansatzes auch die Resultate von *XGBoost* zu analysieren.

3.1.1 Random Forest

Der Random Forest- Algorithmus ist eine weit verbreitete Methode, die die Ergebnisse mehrerer Entscheidungsbäume kombiniert, um eine einzelne Vorhersage zu generieren. Seine einfache Handhabung und Vielseitigkeit haben zu einer schnellen Verbreitung geführt, da er sowohl für Klassifizierungs- als auch für Regressionsaufgaben eingesetzt werden kann. [2]

3.1.2 Extreme Gradient Boosting

XGBoost (Extreme Gradient Boosting) ist eine optimierte und vielseitige Gradient-Boosting-Bibliothek, die für hohe Effizienz und Flexibilität entwickelt wurde. Sie löst Data-Science-Probleme schnell und genau, indem sie mehrere schwache Lernermodelle, wie Entscheidungsbäume, kombiniert, um eine präzisere Vorhersage zu erzielen. XGBoost unterstützt neben Regression und binärer Klassifikation auch benutzerdefinierte Ziel-Funktionen für Multiklassen-Klassifikationsprobleme. [3]

3.2 Erste Vorhersage

Zuerst teilte ich die Daten in Trainings- und Testdaten und ermittelte daraufhin den RMSE (Root Mean Square Error) für die ausgewählten Modelle. Der RMSE ist ein Maß für die Genauigkeit eines Modells, das angibt, wie gut die Vorhersagen mit den tatsächlichen Werten übereinstimmen. Ein niedriger RMSE-Wert zeigt eine hohe Modellgenauigkeit an, während ein höherer Wert auf größere Abweichungen hinweist.

Mit einem RMSE von 1.429 zeigt das Random Forest-Modell eine gute Leistung bei der Vorhersage der Zykluslänge. Im Vergleich dazu hat der XGBoost einen höheren RMSE von 1.516, was darauf hinweist, dass seine Vorhersagen in diesem Fall weniger präzise sind. Durch GridSearchCV versuchte ich, die Hyperparameter des XGBoost-Modells zu optimieren. Der beste RMSE aus der

Kreuzvalidierung lag bei -2.440, jedoch erzielte das Modell nach Tuning einen RMSE von 1.452, was nicht signifikant besser ist als der RMSE des Random Forest.

Zunächst habe ich erstmal das zuletzt optimierte Modell nach dem Hyperparameter-Tuning verwendet, um eine erste Vorhersage für die Länge eines nächsten Menstruationszyklus zu erstellen. Die vorhergesagte Zykluslänge betrug 30,43, also 30 Tage. Dieser Wert schien mir eher ungenau. Außerdem standen mir noch 23 Spalten nach der Datenbereinigung zur Verfügung, jedoch werden nur 22 für die Vorhersage benötigt und ich war mir unsicher, ob die eingegebenen Werte den jeweiligen Spalten korrekt zugeordnet sind.

4 Sprint 3: Modellverfeinerung

In diesem Abschnitt beschriebenen Vorgänge sind im Jupyter Notebook 3-1_Prediction_limitedFeatures.ipynb festgehalten. Die vorherigen Dateien bilden wieder die Grundlage, weshalb die Fortschritte wieder weiter unten im Dokument sichtbar sind.

4.1 Beschränken auf bestimmte Features

Um den Überblick der Spalten zu behalten und präzise Werte für diese Spalten dann eintragen zu können, entschied ich mich auf eine geringe Auswahl von Features zu konzentrieren.

Die folgenden Features dabei sind:

- `LengthofCycle`: Diese Variable ist entscheidend, da sie meine Zielvariable darstellt.
- `EstimatedDayofOvulation`: Dieses Feature weist die höchste Korrelation zur Zielvariable auf.
- `FirstDayofHigh`: Dieses Feature ist realistisch zu schätzen und eignet sich gut als Beispiel.

4.2 Modellversuche mit den gewählten Features

Zuerst verwendete ich den Random Forest-Algorithmus. Anschließend wurde ein Hyperparameter-Tuning für das Random Forest-Modell durchgeführt, gefolgt von der Anwendung des XGBoost-Algorithmus sowie dessen Hyperparameter-Tuning.

4.2.1 Random Forest:

- RMSE: 2.1679
- Beispielvorhersage: 28.2684

Nach Hyperparameter-Tuning:

- Best RMSE aus der Kreuzvalidierung: 2.2904
- RMSE des besten Random Forest-Modells auf den Testdaten: 2.1712
- Beispielvorhersage: 28.2482

4.2.2 XGBoost:

- RMSE: 2.1929
- Beispielvorhersage: 27.9983

Nach Hyperparameter-Tuning:

- Best RMSE aus der Kreuzvalidierung: 2.2783
- Beispielvorhersage: 28.6017

Insgesamt erschien mir die Vorhersage nach den Anpassungen genauer als zuvor. Das zuvor genannte Problem, dass weniger Spalten erforderlich waren als Feature einzugeben als eigentlich nach der Datenaufbereitung noch übrig waren, lag daran, dass die Zielvariable nicht angeben musste und ich dies nicht beachtete.

Außerdem gestaltet sich die Auswahl der Features als herausfordernd. Zunächst hatte ich vor, die Variablen mit der höchsten Korrelation zur Zielvariable zu wählen. Allerdings ist es für die UI wichtig, dass die eingegebenen Werte intuitiv und verständlich für die Nutzerinnen sind. Daher entschied ich mich, statt beispielsweise der "Fertility" das Feature "First Day of High" hier zu verwenden. Dennoch war die Wahl der Features noch einmal zu überdenken.

Mein vorzeitiger Entschluss nach dieser Analyse, wäre es den Random Forest-Algorithmus zu verwenden. Er hatte hier eine genauere Vorhersage mit einem niedrigeren MSE (Mean Squared Error) geliefert.

4.3 Modellversuche mit zusätzlichen Features

In diesem Abschnitt beschriebenen Vorgänge sind im Jupyter Notebook 3-2_Prediction_specificFeatures.ipynb festgehalten. Die vorherigen Dateien bilden wieder die Grundlage, weshalb die Fortschritte wieder weiter unten im Dokument sichtbar sind.

Dennoch wollte ich nocheinmal die beiden Modelle miteinander vergleichen, indem ich wieder weitere Features in die Modelle integrierte. Der Gedanke dahinter war, dass zusätzliche Features, auch wenn sie eine geringere Korrelation aufweisen, möglicherweise für die Nutzerinnen leichter verständlich sind.

Also verwendete ich erneut die beiden Algorithmen und führte jeweils ein Hyperparameter-Tuning durch, um das am besten geeignete Modell auszuwählen.

4.3.1 Auswahl der Features

Ich wählte folgende Parameter dafür aus:

- `LengthofCycle`: Länge des Zyklus, ist die Zielvariable.
- `EstimatedDayofOvulation`: Der geschätzte Tag, an dem der Eisprung stattfindet, basierend auf dem Zyklus oder den Symptomen.
- `LengtofLutealPhase`: Die Länge der Lutealphase des Menstruationszyklus, die Zeit nach dem Eisprung bis zum Beginn der nächsten Menstruation.
- `LengthofMenses`: Die Anzahl der Tage, die die Menstruation dauert.
- `MensesScoreDay1-5`: Bewertungen der Blutung oder Blutungsstärle für die Menstruation an den einzelnen Tagen (von Tag 1 bis Tag 5).
- `TotalMensesScore`: Eine zusammenfassende Bewertung der Menstruationsperiode basierend auf der Blutungsstärle der einzelnen Tage.
- `NumberofDaysofIntercourse`: Die Anzahl der Tage, an denen Geschlechtsverkehr stattgefunden hat.
- `UnusualBleeding`: Ob ungewöhnliche Blutungen außerhalb der normalen Menstruationsperiode festgestellt wurden.

4.3.2 Random Forest:

Neue Ergebnisse

- RMSE: 1.6350139602657356
- Beispielvorhersage: 28.0

Nach Hyperparameter-Tuning:

- Best RMSE from CV: 1.7093076143185757
- RMSE des besten Random Forest Modells auf den Testdaten: 1.8475379011631308
- Beispielvorhersage: 32.52626526526528

vorherige Ergebnisse zum Vergleich:

- RMSE: 2.1679
- Beispielvorhersage: 28.2684

Nach Hyperparameter-Tuning:

- Best RMSE aus der Kreuzvalidierung: 2.2904
- RMSE des besten Random Forest-Modells auf den Testdaten: 2.1712
- Beispielvorhersage: 28.2482

4.3.3 XGBoost:

Neue Ergebnisse

- RMSE: 1.7786670119939103
- Beispielvorhersage: 28.066196

Nach Hyperparameter-Tuning:

- Best RMSE from CV: 1.5873982580409962
- Beispielvorhersage: 27.723345

vorherige Ergebnisse zum Vergleich:

- RMSE: 2.1929
- Beispielvorhersage: 27.9983

Nach Hyperparameter-Tuning:

- Best RMSE aus der Kreuzvalidierung: 2.2783
- Beispielvorhersage: 28.6017

Insgesamt kamen mir die Vorhersagen wieder genauer vor. Die neuen Features haben sich als vorteilhaft erwiesen, und der MSE ist deutlich präziser als zuvor. Besonders hervorzuheben ist das Hyperparameter-Tuning beim XGBoost, das den niedrigsten RMSE geliefert hat. Dennoch entschied ich mich aufgrund des zweit niedrigsten RMSE, Random Forest entgültig zu verwenden, da beim Runden der Ergebnisse beider Modelle das selbe Ergebnis von 28 Tagen rauskam. Ein weiterer Vorteil des Random Forest ist die signifikant kürzere Rechenzeit (1 Minute im Vergleich zu 6 Minuten), was für eine effizientere Implementierung spricht.

5 Sprint 4: Implementierung der Streamlit App

Notizen zur Erstellung der Streamlit App wurden in diesem Jupyter Notebook festgehalten: 4_StreamlitApp_Notes.ipynb.

5.1 Das User Interface

Um die Vorhersage für Nutzerinnen zugänglich zu machen, entwickelte ich ein passendes User Interface und verwendete dafür das Framework Streamlit. Dafür habe ich nunächst Streamlit installiert und die Datei `app.py` im Projektordner `MenstrualCyclePredictionApp` erstellt. Daraufhin fügte ich zu den jeweiligen Features passende Input Felder hinzu. Ich speicherte den zuletzt aufbereiteten Datensatz ab und legte ihn ebenfalls in den Projektordner, um diesen in die `app.py` laden zu können. Daraufhin integrierte ich den ausgewählten Random Forest-Algorithmus und erstellte einen Button *"Vorhersage der Zykluslänge anzeigen"*, um mit einem Klick darauf die Vorhersage auszuführen. Nach dem ersten erfolgreichen Durchlauf, entschied ich mich die Blutungsstärke als Schieberegler statt einfaches Input Feld darzustellen, um die User Experience zu verbessern. Zusätzlich fügte ich ein Informationsfeld hinzu, damit die App verständlich erklärt wird. Abschließend verfeinerte ich noch die Formulierungen.

5.2 Deployment

Für das Deployment meiner Streamlit-App habe ich zunächst das Git-Repository Menstral Cycle Prediction erstellt und anschließend die Ordnerstruktur des Projekts angepasst. Über die laufende Streamlit-App habe ich dann den Deploy-Button verwendet, wodurch ich zur Community-Deploy-Plattform von Streamlit weitergeleitet wurde. Die App wurde dann unter der URL https://menstrual-cycle-prediction.streamlit.app/ bereitgestellt.

Zu Beginn traten jedoch einige Fehler auf, die durch den Import des locale-Moduls verursacht wurden. Dieses Modul verwendete ich, um die vorhergesagten Wochentage auf Deutsch anzuzeigen zu lassen. Um dieses Problem zu lösen, versuchte ich mithilfe einer `package.txt` dem Server mitzuteilen, dass die Lokalisierung generell auf Deutsch gesetzt werden sollte. Dennoch blieb der Fehler bestehen, sodaa ich mich schließlich dazu entschied, die Wochentage auf Englisch zu belassen. Das Datumsformat behielt ich aber im deutschen Stil (TT.MM.JJJJ) bei. Diese Anpassung führte dazu, dass das Deployment schließlich erfolgreich abgeschlossen werden konnte, und die App nun über die genannte URL erreichbar ist.

6 Zusammenfassung und Ausblick

In diesem Projekt wurde erfolgreich ein Vorhersagemodell zur Bestimmung der Länge des Menstruationszyklus entwickelt, das auf den Algorithmus Random Forest basiert. Es erwies sich als effizient und lieferte präzise Vorhersagen, wodurch die Möglichkeit einer zuverlässigen Zyklusvorhersage gegeben ist.

Dieses Projekt gab mir wertvolle Einblicke in das Thema Machine Learning und zeigte die Herausforderungen bei der Auswahl aussagekräftiger Features in Datensätzen auf. Zukünftige Projekte werde ich mit einer sorgfältigeren Auswahl von Daten angehen, insbesondere wenn ich spezifischere Vorhersagen, wie über Endometriose-Schübe, anstrebe. Die Verwendung von Streamlit hat sich als äußerst nützlich erwiesen, um Machine Learning-Modelle in einem ansprechenden User Interface zu präsentieren, was ich für zukünftige Anwendungen unbedingt empfehlen kann.

7 Quellenverzeichnis

- [1] https://www.endometriose-vereinigung.de/was-ist-endometriose/
- [2] https://www.ibm.com/de-de/topics/random-forest
- [3] https://docs.kanaries.net/de/topics/Python/xgboost