Prediction of location and type of inflammation for patients with uveitis based on blood values and laboratory tests.

Studer Roman   
BSc Data Science Student, FHNWBrugg-Windisch, Switzerland  
roman.studer1@students.fhnw.ch

Rau Alexandre  
BSc Data Science Student, FHNWBrugg-Windisch, Switzerland  
alexandre.rau@students.fhnw.ch

*Abstract*—

Keywords—component, formatting, style, styling, insert (key words)

# Introduction

## Uveitis

Uveitis is a term which describes an inflammation of the uvea. Uveitis can be divided into anterior (lat. In front), posterior (lat. In back), intermediate or panuveitis (more than one segment affected). For example, an anterior uveitis involves the iris [1]. Panuveitis is an inflammation of the whole uvea tract as well as the retina and the vitreous humor (glass body) [2]. Uveitis can lead to the loss of eyesight among other things.

## Project Description

The aim of the project was to identify important features for the diagnosis of uveitis. For this purpose, a dataset with information on more than 1000 patients, collected by Dr. H. Nida Sen et al. from the National Eye Institute, Washington DC, was made available. After an initial exploratory data analysis, a pre-processing pipeline was developed, which can be used together with machine learning algorithms from sklearn, a Python machine learning library. Various algorithms were employed to classify the dataset. The results, especially the feature importance’s, were recorded and documented. Three features were identified as target variables:

Location: This marker describes the location of inflammation with the categories Anterior, Posterior, Intermediate, Panuveitis and Scleritis. The category "Scleritis" refers to inflammation of episcleral and scleral tissue [3]. Prediction of the location based on laboratory values allows the identification of a subset of laboratory tests (via feature importance) that are suitable for prediction. This would allow a small subset of tests to be used for diagnosis based on the location of the inflammation. In addition to faster diagnosis, the reduced number of tests required would lead to a reduction in costs.

A second target feature is “Category”. This marker describes the origin of the ocular inflammation. This can be, for example, systematic, infectious, or idiopathic. This feature is based on the results of laboratory tests and has been recorded retrospectively. Predicting the category, aka the origin of the inflammation, can aid the diagnosis further.

The third target feature is the specific diagnosis itself. In the dataset, 27 different diagnoses were recorded (some can be collapsed based on similarity). The direct prediction of a diagnosis based on laboratory tests could support the medical staff in their decision making.

## Data description

We received a total of 1075 samples from patients affected by certain types of ocular inflammatory diseases. Mostly subtypes of uveitis such as pars planitis but also other diseases that have inflammation in the eye as a symptom or consequence, e.g., white dot syndrome or sarcoidosis. We count 426 male patients and 649 female patients. The difference between male and female patients can be explained as women are disproportionately affected by ocular inflammation [4]. Each sample is described by a total of 64 attributes. The attributes can be divided into laboratory tests (blood values), meta-information about the patient (such as gender or race), and features describing the diagnosis. For the purpose of the analysis, the binary feature "uveitis" was introduced which determines whether the patient has a form of uveitis based on the specific diagnosis.

Information about the patient includes “Subject ID”, “Gender” and “Race”. The location of the inflammation is described in the marker “location” and in “AC Abn Od Cells”, “AC Abn Os Cells”, “Vit Abn Od Cells”, “Vit Abn Os Cells”, “Vit Abn Od Haze”, and “Vit Abn Os Haze”. These qualitative, ordinal features describe the severity of the inflammation of the Anterior Chamber Cells (AC) in either the left eye (OS) or the right eye (OD). The inflammation can be rated as 0, +0.5, +1, +2, +3, +4. The higher the value, the more severe the inflammation is. If a value of +0.5 or higher is present a patient can be considered as "Active", else as "Quiet". The diagnosis is described in the features “categorical”, “EHR Diagnosis” and “specific diagnosis”. The laboratory tests provide a variety of results (mostly blood values) and include: "Calcium”, "Lactate Dehydrogenase", "C-Reactive Protein, Normal and High Sensitivity”, “WWBC”, “RBC ”, “Hemoglobin”, “Hematocrit”, “MCV”, “MCH”, “MCHC”, “RDW”, “Platelet Count”, “Neutrophil %”, “Lymphocytes %”, “Angiotensin Conv#Enzyme”, “Beta-2-Microglobulin”, “Lupus Anticoagulant”, “Lysozyme (Plasma)”, “Anti-CCP Ab”, “Anti-Dnase B”, “Anti-ENA Screen”, “Antinuclear Antibody (ANA)”, “Complement C3”, “Complement C4”, “DNA Double-Stranded Ab”, “HLA-A\*”, “HLA\_A\_1”, “HLA\_A\_2”, “HLA-B\*”, “HLA\_B\_1”, “HLA\_B\_2”, “HLA-Cw\*”, “HLA\_C\_1”, “HLA\_C\_2”, “HLA-DRB1\*”, “HLA\_DRB1\_1”, “HLA\_DRB1\_2”, “HLA-DQB1\*/DQ\*” , “HLA\_DQ\_1”. “HLA\_DQ\_2”, “HLA-DRB\_\*”, “HLA\_DRB\*\_1”, “HLA\_DRB\*\_2”, “Myeloperoxidase Ab”, “Proteinase-3 Antibodies”, “Rheumatoid Factor”, “HBc (HepB core) Ab”, “HBs (HepB surface) Ag”, and “HCV (HepC) Ab”. Features containing the prefix “HLA”, which stands for “Human Leukocyte Antigen” represent different haplotypes.

# Exploatory data analysis

The scope of exploratory data analysis was to evaluate and properly prepare the data for further elaboration while highlighting primary/principal insights.

The whole dataset was taken into consideration. Ascertaining and communicating a missing values strategy is paramount to ensure reliability, reproducibility and must be kept in consideration while analysing final results. For this, an overview of missing information was created [5] to allow to establish, during pre-processing, a satisfactory missing values approach.

Observations indicate that columns “\_others” and “notes” contain 79.07% missing values. Other columns have a similar issue; “anti-dnase\_b” is composed of 99.63% of missing values. Features “beta-2-microglobulin” and “lupus\_anticoagulant” contain approximately 65% missing values. This underlines the need for a highly flexible missing values strategy that is not limited to only imputing missing values but also to selectively remove features that score above a determined missing value percentage.

Next steps include controlling for data inconsistencies. Edge cases were found in the UOM columns, prompting an accurate evaluation and appropriate response during pre-processing. Then came formatting errors, where extensive work has to be invested to adapt non-standard missing values to machine readable information. Possible optimizations included collapsing variables. This includes the extreme where the target is strictly binary and less drastic measures, i.e., by removing or collapsing, low count occurrences in the “specific diagnosis” column. Totally removing features like “serodiagnoses” and “notes” are also available options to be considered. These features are considered non-essentials.

# Preprocessing

The data obtained needs to be cleaned before use. During the EDA process, some deficiencies were noticed. In order to resolve the issues, a pre-processing pipeline was developed, which allowed for great flexibility in data preparation with 13 parameters. The pipeline consists of a series of functions that each take a DataFrame as input and create a DataFrame as output. We take a closer look at the pre-processing by distinguishing categorical from numerical features. After importing the data from Excel format into a Pandas DataFrame structure, we started by editing the column names. Whitespace were removed, additions like "(Blood)" were removed and upper-case letters were replaced by lower-case letters. In a further step, columns with a relative proportion of missing values above >20% were removed. This includes the columns "anti-dnase-b" (>99%), "other\_" (>79%), "notes" (>79%), "beta-2-microglobulin" (>65%), "lupus\_anticoagulant" (>65%), "myeloperoxidase\_ab" (>62%), "proteinase-3\_antibodies" (>62%), "complement\_c3" (>23%), "complement\_c4" (>23%). Imputing values with such a large relative frequency of missing values in a small number of observations (n=1075) is questionable. A function to remove columns based on a substring in the column name was additionally introduced. This allows to remove columns based on a substring, e.g., "range".

## Categorical Features

Essentially, categorical features must be converted to the dtype "category" for correct handling by the encoder. However, some require special adjustments, merging of classes, or clean-up of capture errors. The categorical variable "Race" includes the category "race or ethnic group data not provided by source". These values are treated as missing values, aka in the category 'unknown’ since they do not contain any information about the respective person. "Race or ethnic group data not provided by source" and "unknown race" were collapses into the category "unknown". Missing values (NaN's) are also marked with 'unknown'. The feature "location" is a special case. The classes of the categorical feature, out of a total of 5 classes, can be collapsed into the two classes "anterior" and "posterior". This allows us to model two different situations later:

### We keep the categories 'anterior', 'intermediate', 'panuveitis', 'posterior' and 'sclerits'. All categories indicate a diffrent section of the eye (or multiple at once) that show inflammation.

### We collapse mutliple categories to get an 'anterior' and 'posterior' category. Aka, collapse the location to inflammations in the front and the back of the eye (binary feature). To achieve this we collapse the categories 'intermediate', 'posterior' and 'panuveities' to the category "posterior\_segment". 'anterior' and 'scleritis' get collapsed to the category 'anterior\_segment'.

The class 'pan' is synonymous to 'panuveitis' and can be collapsed.

The column "category" describes the origin of the inflammation and takes in majority the class "idiopathic". The classes "nonneoplastic masquerade" and "neoplastic masquerade" describe a pseudo-uveitis and were transferred to the class "not\_uveitis" [6]. "specific\_diagnosis" contains 27 different classes, some with very low absolute frequencies. To reduce the number of classes, a function was developed that allowed the merging of classes with <n appearances into the class "other". Later, all classes with <20 appearances were merged. The features with prefix "ac\_abn\_" and "vit\_abn\_" have class "C" (which stands for "Cannot identify"). These values are deleted and replaced with NaN values. The features "hbc\_ab", "hbs\_ag" and "hcv\_ab" hold results for different types of hepatitis. A patient can be either negative or reactive. The function converts invalid values to NaN values and codes negative cases with 0 and reactive cases with 1.

## Numerical Features

Numerical features are mostly lab results which each have a corresponding column denoting the accepted range of values and the unit of measurement (UOM). These include but are not limited to "calcium", "lactate dehydrogenase" or "C-Reactive Protein, Normal and High Sensitivity". A feature can take more than one range. The ranges are given by the laboratory that performed the test which can differ from one to another. Figure 1 shows the distribution and differences of four different ranges for the feature "lactate dehydrogenase". We can see that a large part of the data (>50%) is in the ranges (annotated by the dashed lines). Such features can be converted into a categorical feature. The pre-processing function offers the possibility to convert all features with corresponding ranges into categorical features. Values below the specified minimum of the range are converted to 0 = 'below range', values in the accepted range are converted to 1 ='in range' and values above the maximum are converted to 2 = 'above range'.

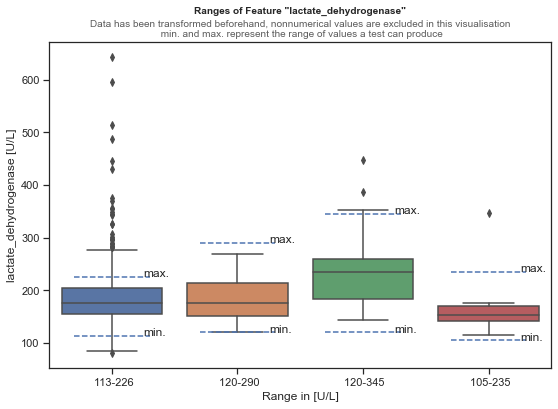


Figure 1, Ranges of Feature "lactate\_dehydrogenase"

The feature “C-Reactive Protein, Normal and High Sensitivity” contains multiple units of measurement (mg/dl and mg/L). In pre-processing, the values were standardized to mg/dl. HLA features were not used and were removed later.

## Features containing numeric and categorical data

## Imputing missing values

In addition to the pre-processing function, we have introduced the possibility to impute missing values in different ways. Analogous to the sklearn class SimpleImputer, missing values can be replaced by the mean, the median, the most frequent value or by a constant. In addition, we applied a KNN imputer which imputes missing values using a k-Nearest Neighbour. Here, the missing values are replaced by the mean of the values of the nearest neighbours. Two observations are similar if both have similar non-missing values. [7]. For the models explained in IV, either categorical values were imputed using most frequent and numerical values were imputed using the mean, or the values were imputed by a KNN with n = 3 (three neighbours).

## Encoding

Many machine learning algorithms can only work with numeric data. Strings or other data types can therefore not be used. This makes a transformation of the categorical variables necessary. There are two possibilities:

### Assign a number to a class of a categorical variable. E.g. in the variable "category" 1 represents 'idiopathic', 2 represents 'systemic' etc.. However, this can lead to problems, because a higher number can be considered as "more important", which leads to an unwanted ranking of the classes.

### For each class of a categorical variable, a new feature is created which binary records whether an observation falls into this class or not. Example: the class 'idiopathic' in "category" becomes a new feature with the name 'category\_idiopathic' which can take the values 1 or 0. 1 indicates that this observation in "category" takes the class 'idopathic'. This is called OneHotEncoding and is the preffered method as it prevents a ranking of the classes [8].

# Modelling

In this chapter we describe the modeling procedure and the algorithms used. We present the results in Appendix A, as well as in Chapter V. In addition to the total data set with n = 1075 samples before pre-processing (depending on the target variable, the total number of samples varies slightly, since samples with missing values in the target variable were removed), two additional data sets divided by gender were introduced. We thus investigate the assumption that we could obtain better results if we treated male and female patients separately. This could also lead to different feature importance and thus to differences in the variables relevant for prediction.

We focused on classification algorithms that are comprehensible, reproducible and allow the extraction of feature importance. We considered seven algorithms: Decision Trees, Random Forest, k-nearest Neighbour (KNN), Support Vector Machines (SVM), XGBoost, AdaBoost with Decision Trees as base estimators, and Multi-Layer Perceptron (MLP). With the exception of MLP, SVM, and KNN, all algorithms are tree-based, albeit a majority are ensemble methods.

### KNN searches for the most similar neigbours and assigns the most frequent label.

### Decision Trees split the data based on rules and try to minimize entropy. A label is predicted by following these rules.

### XGBoost, AdaBoost and Random Forest are all ensemble models that use multiple decision trees as a base estimator. XGBoost and AdaBoost are gradient boosted. All predict a label on a majority vote from the base estimators.

### Support Vector Machines try to split the data into groups so that around the splits (support vectors) exists the biggest possible margin without any samples in it.

### MLP is the most basic form of neural network.

## Evaluation

The data was split into training and test set using sklearn's train\_test\_split function. A stratified split was performed, meaning that the training and test sets have the same underlying distribution. The testset has a relative share of 25% of the data. Using a GridSearch approach and cross-validation (cv = 3), the models attempted to maximize the target metric. The models were trained with two target metrics appropriate for a multiclass scenario:

F1 score takes the harmonic mean between a model's Precision and Recall. Precision describes the relative proportion of positive assignments of a model that actually belong to this class. Recall describes the relative proportion of actually correctly identified samples. The harmonic mean means that the F1 score can only be high if both Recall, and Precision are high. If one of the values is strongly lower, the F1 score is lower than it would be with a simple arithmetic mean.

### (1)

Since we did not only work with binary target variables that record the membership to a single class, but with several, i.e., we perform a multiclass classification, we calculate the F1 score for each individual class and take the mean of it. Another metric suitable for the multiclass case is the Balanced Accuracy. It takes the mean of the recall over all classes. Both metrics take values in the range [0,1] and are often expressed as percentages.

We trained all mentioned algorithms with all three datasets with both target metrics. In each case, four different imputation methods (OneHotEncoding, No OneHotEncoding, OneHotEncoding with KNN as imputer and no OneHotEncoding with KNN as imputer) were used. In total, 712 different models were trained. (Although the actual number of trained models is significantly higher, since only the best models were retained during the respective run.

## Location

D The dataset was prepared for location prediction as follows: Features containing non-laboratory values were removed. Features that allow direct inferences to the location were also removed (i.e. ac\_abn\_... and vit\_abn\_... columns). The location can thus only be predicted directly from laboratory values, which prevents an influence through other, unrelated values and thus allows the selection of a subset of laboratory tests per location.

#### Binary Classification

In the binary case, the target variable can only take on the value "anterior" or "posterior".

#### Multiclass Classification

#### One vs. All

## Category

## Specific Diagnosis

# Results

##### Discussion

##### Conclusion

##### Acknowledgment

##### References