Prediction of location and type of uveitis based on blood values and laboratory tests.

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

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

***Abstract*— The aim of the work was to classify a dataset with multiple ocular inflammatory diseases. Majority uveitis. Models were created to predict several target features. Location of inflammation, origin of inflammation, and the specific diagnosis itself. Machine learning models can identify important features. As a result, a subset of the former laboratory tests can be used to support a diagnosis, resulting in cost savings. After an initial exploratory data analysis, a pre-processing pipeline was created which, in harmony with various machine learning algorithms, enabled the training of a total of 712 different models (from 7 different algorithms, mostly tree based). We noticed that the classification proved to be difficult as the data is hard to separate. For the prediction of the location, the most important features could be identified. We propose to further improve the best models found and thus strengthen the relevant feature importance criteria.**

Keywords—uveitis, machine learning, feature importance, classification

I. Introduction

A. 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.

B. 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 scikit-learn, a Python machine learning library [3]. 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 features:

Location: This feature 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 [4]. 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 lab tests required would lead to a reduction in costs.

A second target feature is "Category". This feature 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.

C. 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 [5]. Each sample is described by a total of *64* attributes (excluding range and UOM[1] 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 feature "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", "WBC" (white blood cell) , "RBC" (Red blood cells), "Hemoglobin", "Hematocrit", "MCV" (mean corpuscular volume), "MCH" (Mean corpuscular hemoglobin), "MCHC" (mean corpuscular

[1] UOM: Unit of Measurement

hemoglobin concentration), “RDW” (red cell distribution width), “Platelet Count”, “Neutrophil %”, “Lymphocytes %”, “Angiotensin Conv#Enzyme”, “Beta-2-Microglobulin”, “Lupus Anticoagulant”, “Lysozyme (Plasma)”, “Anti-CCP Ab” (Anti-cyclic citrullinated peptides), “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”, and the following types of hepatitis: “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. Other columns like “notes” were not considered.

II. Exploratory 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 [6] 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 features. 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.

One final aspect of explorative data analysis that we undertook was the creation of a correlation matrix for all lab test results. Normalized correlation is a numerical value that ranges from *-1* to *1*. If the value is high in either direction, the features will move in tandem. The sign indicates the direction of the correlation. The Pearson Correlation used measures the strength of the linear relationship between two numerical features. A positive result indicates a positive linear relationship, while a negative result indicates a negative relationship. And if the correlation result tends to zero it stands to indicate that the features change independently of each other. The Pearson correlation coefficient is used to check for possible linear relationships in data between different features. It is paramount to remember that correlation does not equal causation [7]. The goal was to spot where these correlations are located and what first impressions can be extracted from these results.

Figure 1, a correlation matrix on a subset of numerical features shows the linear relationships between all possible combinations of features. The full correlation matrix can be found in appendix B. We observe a moderate to strong correlation between the following feature pairs:

- *Neutrophil and Lymphocytes negatively correlate strongly with* -0.94.
- *MCH and MCV correlate strongly with* 0.89
- *MCH and MCHC correlate moderately with* 0.59
- *MCH and Hemoglobin correlate moderately with* 0.45
- *MCHC and Hemoglobin moderately correlate* 0.56
- *Hemoglobin and Hematocrit strongly correlate* 0.96
- *MCH and RDW negativly correlate with* -0.49

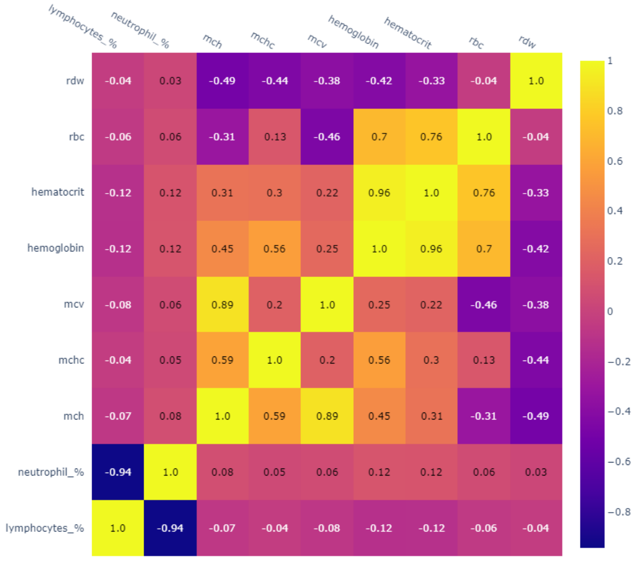

Fig. 1. Correlation Matrix of feature subset

Possible next steps to gain additional insight could include comparing correlations between different uveitis categories, location, and specific diagnosis.

III. 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 up to 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 began 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% missing)*, "other_" *(>79% missing)*, "notes" *(>79% missing)*, "beta-2-microglobulin" *(>65% missing)*, "lupus_anticoagulant" *(>65% missing)*, "myeloperoxidase_ab" *(>62% missing)*, "proteinase-3_antibodies" *(>62% missing)*, "complement_c3" *(>23% missing)*, "complement_c4" *(>23% missing)*. 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 introduced additionally. This allowed to remove columns based on a substring, e.g., "range".

A. *Categorical Features*

Essentially, categorical features must be converted to the dtype "category" for correct handling by the encoder, which transform the data into a machine-readable format. However, some require special adjustments, merging of classes, or clean-up of capture errors. The categorical feature "Race" includes the class '*race or ethnic group data not provided by source*'. These values were treated as missing values and transferred to 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[2]) 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:

1) *We keep the categories 'anterior', 'intermediate', 'panuveitis', 'posterior' and 'sclerits'. All categories indicate a different section of the eye (or multiple at once) that show inflammation.*
2) *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 exhibits the class *'idiopathic'* most frequent. The classes *'nonneoplastic masquerade'* and *'neoplastic masquerade'* describe a pseudo-uveitis and were transferred to the class *'not_uveitis*' [8]. "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_" show the 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. A function converts invalid values to NaN values and encodes negative cases with *0* and reactive cases with *1*.

B. *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 on 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 constructed pre-processing function allowed for 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'*.

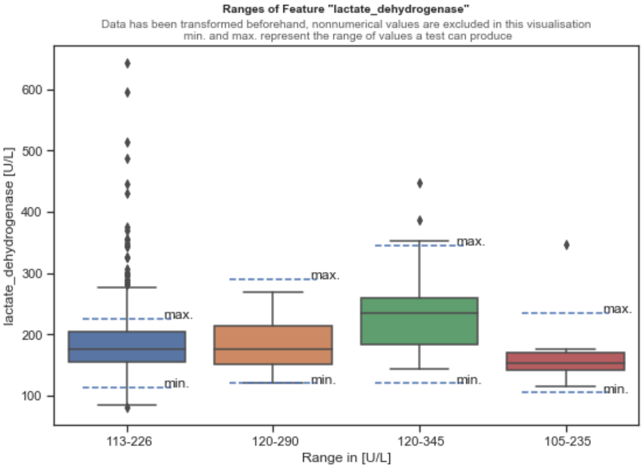

Fig. 2, Ranges of Feature "lactate_dehydrogenase"

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 at a later stage.

C. *Features containing numeric and categorical data*

Several features contain both numeric values and categorical values. These have been treated separately to the clean numeric and categorical features. These include: "Anti-CCP Ab", "Anti-ENA Screen", "Antinuclear Antibody", "DNA Double-Stranded Ab", "Myeloperoxidase Ab" and "Proteinase-3 Antibodies". The majority of "Anti-ENA Screen", "Antinuclear Antibody" and "DNA Double-Stranded-Ab" are negative, as in they contain the value *'NEGATIVE'*. Numerical values represent a positive expression. These features were converted to binary

[2] NaN: Not a Number

categorical features by encoding values '*NEGATIVE'* with 0 and values >*0* with 1. The features "Myeloperoxidase Ab" and "Proteinase-3 Antibodies" were removed on account of too many missing values.

D. Imputing missing values

In addition to the pre-processing function, we have introduced the possibility to impute missing values in various 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 [9]. 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 n nearest neighbours. Two observations are similar if both have similar non-missing values. [10]. For the models explained in IV, either categorical values were imputed with the most frequent value and numerical values were imputed using the mean value, or the values were imputed by a KNN with n = 3 (three neighbours).

E. 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 features necessary. There are two possibilities that were considered and implemented as options in the pre-processing pipeline:

1) *Assign a number to a class of a categorical feature. E.g. in the feature "category" 1 represents 'idiopathic', 2 represents 'systemic' etc.. However, this can lead to problems, as a higher number can be considered as "more important", which leads to an unwanted ranking of the classes.*
2) *For each class of a categorical feature, a new feature is created which records whether an observation falls into this class or not in a binary fashion. Example: the class 'idiopathic' in "category" becomes a new feature with the name 'category_idiopathic' which can take on 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 [11].*

IV. 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 feature, the total number of samples varies slightly, since samples with missing values in the target feature 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's and thus to differences in the features 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. MLP, SVM and KNN do not allow for easy feature importance extraction.

1) *KNN searches for the k most similar neigbours and assigns the most frequent label [12].*
2) *Decision Trees split the data based on rules and try to minimize entropy. A label is predicted by following these rules [13].*
3) *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 [14], [15].*
4) *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 [16].*
5) *MLP is the most basic form of a neural network [17].*

B. 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 test set has a relative share of 25% of the data. Using a Grid Search 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 [18]. The harmonic mean ensures 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.

$$F_1 = 2 \cdot {}^{\text{Precision} \cdot \text{Recall}}/_{Precision + Recall} \quad (1)$$

Since we did not only work with binary target features that record the membership to a single class, but with several, i.e., we perform a multiclass classification, we calculated the F1 score for each individual class and took the mean of it. This is called *Macro-F1* [19, p. 1]. Another metric suitable for the multiclass case is the Balanced Accuracy. It takes the mean of the recall over all classes [20]. Both metrics take values in the range [0,1] and are often expressed as percentages. We trained all mentioned algorithms with all three datasets on 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 each respective run.

C. *Location*

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. In a further step, all samples that are not uveitis positive were filtered out. The aim is to identify features that are relevant for the diagnosis of uveitis. Thus, samples that are not of a type of uveitis are of no interest.

a) *Binary Classification*

In the binary case, the target feature can only take on the value "anterior" or "posterior". Figure 3 shows the imbalance between the two classes. Based on this distribution, a baseline model was developed that randomly predicts based on the prior probability distribution. In this binary case, a Macro F1-score of *~0.49* (complete, uveitis positive dataset with OneHotEncoding and no KNN-imputer) is obtained on the test set. For a uniform distribution, we would expect a score of *0.5*. The classes have a prior probability of *anterior segment = 0.31* and *posterior segment = 0.69*. All seven algorithms mentioned were tested here.

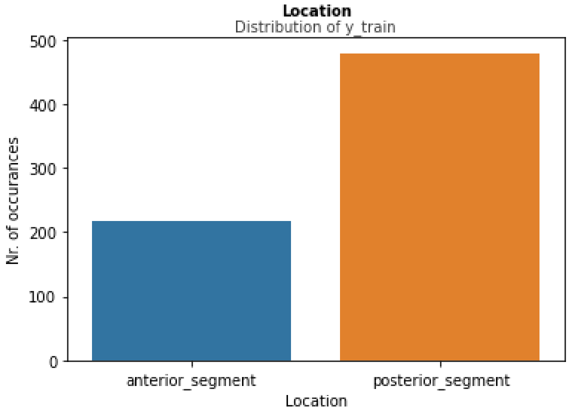

Fig.3, Distribution of binary location target feature

b) *Multiclass Classification*

In the Multiclass case, the class *'Scleritis'* was removed because the class has too few values. Thus, four classes remain: *anterior, intermediate, panuveitis, posterior*. Figure 4 shows a relatively uniform distribution of classes (total, uveitis positive dataset with OneHotEncoding and without KNN imputer). In the baseline model, we expect a score of *~0.25* with four uniformly distributed classes. We achieve a Macro F1 score of ~*0.24*. Prior probability is *anterior = 0.26, intermediate = 0.2, panuveitis = 0.26, posterior = 0.28*. As in the binary case, all the mentioned algorithms were used here.

c) *One vs. All*

Here we focus specifically on feature importance. For each class of location (e.g., *anterior, posterior* etc.) a separate model is trained. All other classes are combined and described with '*other'*. Figure 5 shows such a distribution in the case of anterior vs. all. The goal here was to develop a model that can predict the minority class and extract the important features from it. At this stage we only worked with XGBoost, as it one of the most promising candidates.

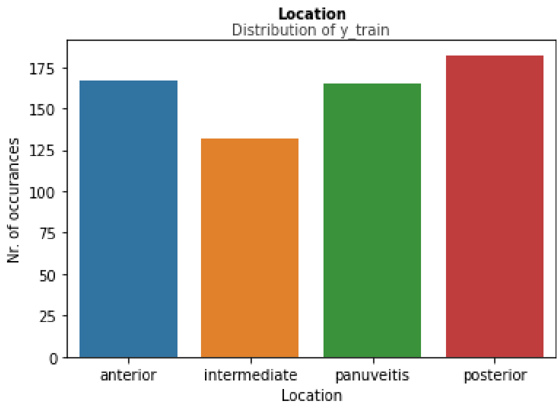

Fig. 4, Distribution of multiclass location target feature

D. *Category*

The "Category" feature can take the values *'idiopathic', 'infectious', 'neoplastic masquerade', 'nonneoplastic masquerade', 'not uveitis', 'scleritis*', *'systemic'* and *'wds'*. *'neoplastic masquerade'* and *'nonneoplastic masquerade'* were transferred to the category *'not uveitis'*. Again, features that allow a direct inference to the feature "Category" were removed. All features containing the following substrings were removed: 'hla', 'ac_', 'vit_', 'loc' (location), *'race'* and *'specific_diagnosis'*. The classes were merged as shown in figure 6. In the baseline, we obtained an f1-score of 0.24 for four classes.

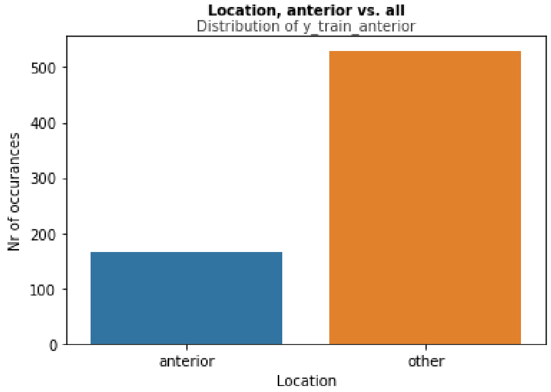

Fig. 5, Distribution of one vs. all location target feature for class "anterior"

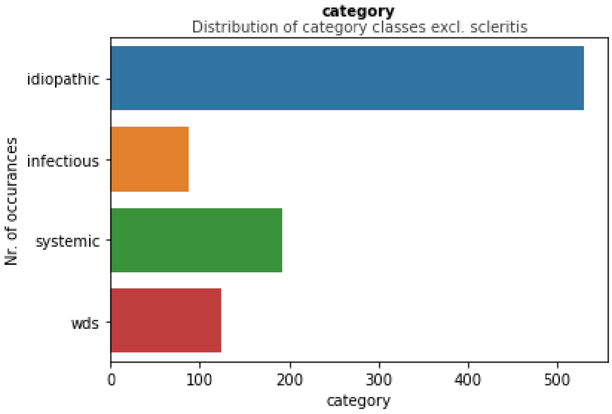

Fig. 6, Distribution of category feature after transformation

E. Specific Diagnosis

Specific Diagnosis is a target feature with a high number of classes. The class "idiopathic" is the majority class and is comprised of the sub-classes 'idiopathic_anterior', 'idopathic_posterior' and 'idiopathic_panuveitis'. We randomly removed samples with the class "idiopathic" until there were only 140 samples left in the dataset. We then obtained better results as the classes were more uniformly distributed. Classes with an absolute frequency of <20 were transferred to the class "other". The classes 'presumed_sarcoidoisis' and 'bx_proven_sarcoidoisis' were merged into the class 'sarcoidosis'. Thus, a total of 27 classes were merged into 12 classes. In the baseline model, we obtain a Macro F1 score of ~0.07. A score of 0.08333 would be expected for a uniform distribution. For the classification of “specific_diagnosis” all seven machine learning algorithms were used.

V. RESULTS

We recorded the Macro F1-Scores and Balanced Accuracy for every trained machine learning algorithm in the appendix A. In this section we discuss the best results per target feature divided into datasets used. The following tables record the maximum score achieved by the respective model. For comparison, the score of the baseline model is given. The algorithms were trained with cross-validation (cv = 3). The standard deviation of the target metric (out of 3 cross-validation runs) is given in std. The column Strategy can take on four values and records the pre-processing strategy. "OneHot" signifies that a OneHotEncoding of the categorical features has taken place, and missing values are imputed either with most frequent values or by the mean value. "No OneHot" means that only an imputation of missing values has taken place. "OneHot+KNN" means that in addition to a OneHotEncoding the missing values were imputed by a KNN-algorithm (k=3). "KNNImputer" means that only one imputation was done by KNN (k=3).

A. Location, Binary Classification

TBL 1. MACRO F1-SCORE AS TARGET METRIC

Dataset	**Strategy**	**Algorithm**	**Baseline**	**Score**	**Std**[a]
Complete	OneHot	XGBoost	0.49	0.54	±0.04
Male	No OneHot	AdaBoost	0.52	0.63	±0.02
Female	KNNImputer	XGBoost	0.57	0.58	±0.006

[a] Std: Standard Deviation.

TBL 2. BALANCED ACCURACY AS TARGET METRIC

Dataset	**Strategy**	**Algorithm**	**Baseline**	**Score**	**Std**
Complete	OneHot	XGBoost	0.48	0.56	±0.033
Male	OneHot+KNN	AdaBoost	0.51	0.65	±0.048
Female	KNNImputer	XGBoost	0.57	0.58	±0.008

The binary location (Table 1 and 2) is consistently best predicted by the ensemble algorithms (XGBoost and AdaBoost) across all three datasets. On the dataset consisting of the female patients, we only achieve an improvement of ~1% over baseline here. On the whole dataset, and specifically on the male dataset, we see a significantly better score of up to +14% in balanced accuracy. In Figure 7 we identify as the four most important features "Angiotensin Conv#Enzyme" (~24%), "WBC" (~23%), "MCHC" (~15%), "Platelet Count" (~14%) for predicting whether inflammation is present in the anterior or posterior region in male patients.

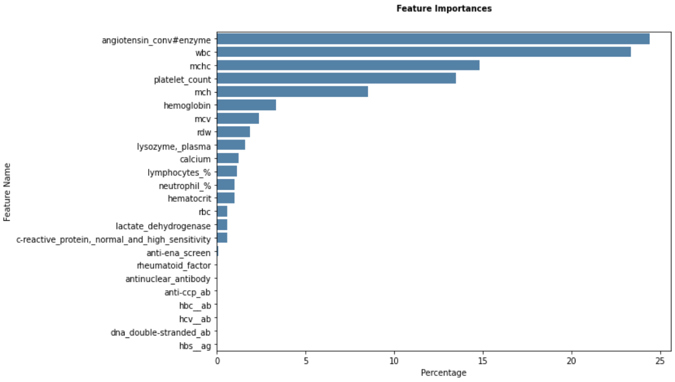

Fig. 7, Feature importance for male, positive uveitis data with AdaBoost and balanced accuracy

B. Location, Multiclass Classification (4 classes)

TBL 3. MACRO F1-SCORE AS TARGET METRIC

Dataset	**Strategy**	**Algorithm**	**Baseline**	**Score**	**Std**
Complete	KNNImputer	XGBoost	0.24	0.31	±0.009
Male	OneHot	AdaBoost	0.32	0.33	±0.052
Female	KNNImputer	Decision Tree	0.25	0.33	±0.042

TBL 4. BALANCED ACCURACY AS TARGET METRIC

Dataset	**Strategy**	**Algorithm**	**Baseline**	**Score**	**Std**
Complete	OneHot+KNN	XGBoost	0.24	0.32	±0.009
Male	OneHot	MLP	0.32	0.36	±0.018
Female	KNNImputer	Decision Tree	0.25	0.36	±0.026

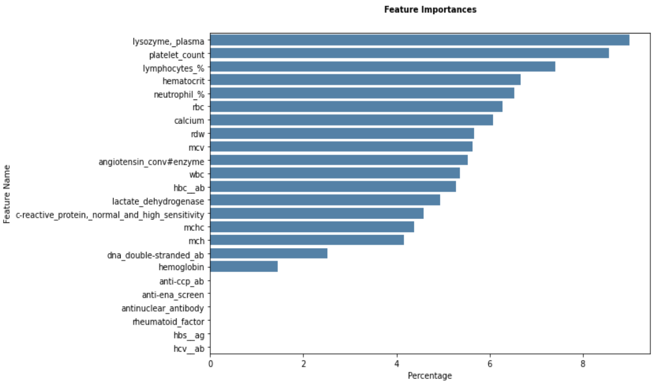

Fig.8, Feature Importance for complete, positive uveitis data with XGBoost

Tables 3 and 4 hold the best results for the multiclass case of location prediction. We see a good improvement over baseline (+8% at balanced accuracy) for the whole dataset. For the data set with the male patients only a small improvement is visible. In Figure 8 we see less pronounced feature importance’s than in the binary case. The most important four features for the prediction of the location over the whole data set are in the multiclass case "Lysozyme Plasma" (~9%), "Platelet Count" (~9%), "Lymphocytes" (~8%) and "Hematocrit" (~7%).

C. *One vs. All, Binary Classification*

In the case of One vs. All binary classification, all locations except one are assigned to the category "other". Here we are only interested in the score of the model in Table 5. We note the feature importance for all 5 categories of location.

TBL. 5 MACRO F1-SCORE AS TARGET METRIC

Class	F1-Score			Balanced Accuracy		
	Strategy	Score	Std	Strategy	Score	Std
Anterior	KNNImputer	0.51	±0.034	KNNImputer	0.53	±0.024
Intermediate	OneHot	0.47	±0.04	OneHot	0.49	±0.014
Panuveitis	OneHot+KNN	0.51	±0.033	OneHot+KNN	0.52	±0.021
Posterior	OneHot+KNN	0.53	±0.016	OneHot+KNN	0.53	±0.017
Scleritis	KNNImputer	0.48	±0.0003	OneHot+KNN	0.5	±0.015

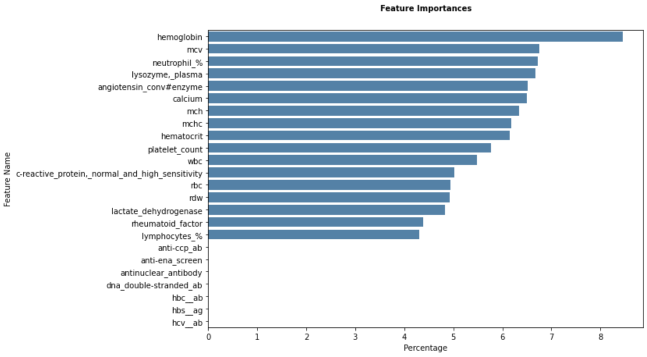

Fig. 8, Feature Importance of "Anterior vs. All", XGBoost,

In the case of Anterior vs. All (Figure 8), the feature "Hemoglobin" stands out with (~8%). The following 3 most important features are "MCV" (~7%), "Neutrophil" (~7%) and "Lysozyme Plasma" (~7%).

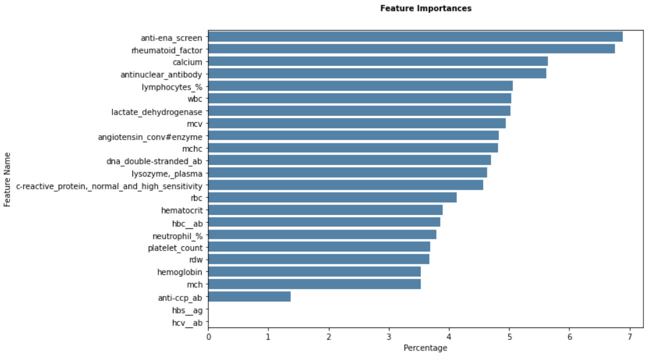

Fig. 9, Feature Importance of "Intermediate vs. All", XGBoost

In "Intermediate vs. All" (Figure 9), two features stand out: "Anti-Ena Screen" (~7%) and "Rheumatoid Factor" (~7%). The fact that the importance of other features decreases only slightly indicates that the algorithm has not identified any clear favorites.

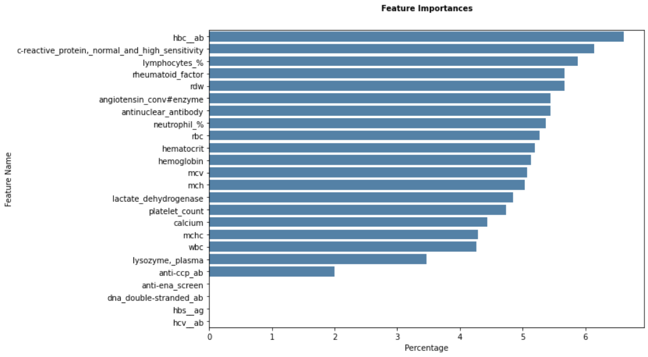

Fig. 10, Feature Importance of "Panuveitis vs. All", XGBoost

Figure 10 shows the feature importance of "Panuveitis vs. all". "HBc Ab" (~7%) and "C-Reactive Protein, Normal and High Sensitivity" (~6%) dominate. As in the multiclass classification for location, "Lymphocytes" (~6%) is also an important indicator.

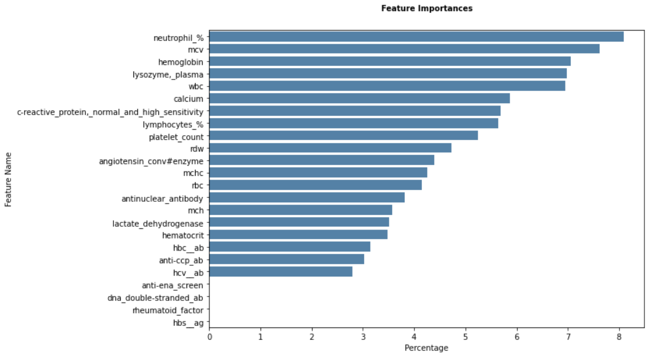

Fig. 11, Feature Importance of "Posterior vs. All", XGBoost

Figure 11 shows the feature importance for the posterior vs. all case. We identify 5 important features: "Neutrophil" (~8%), "MCV" (~8%), "Hemoglobin" (~7%), "Lysozyme Plasma" (~7%) and "WBC" (~7%).

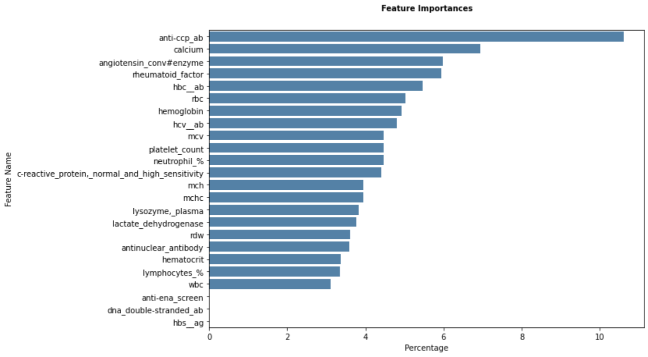

Fig. 12, Feature Importance of "Scleritis vs. All", XGBoost

In the case of "Scleritis vs. All" (Figure 12), the feature "Anti-ccp Ab" stands out by far with ~11%. Followed by the feature’s "calcium" (~7%), "Angiotensin Conv#Enzyme" (~6%) and "Rheumatoid Factor" with ~6%.

D. Catgory, Multiclass Classification (4 classes)

Tbl 6. Macro F1-Score as target metric

Dataset	Strategy	Algorithm	Baseline	Score	Std
Complete	OneHot	SVM	0.24	0.32	±0.019
Male	OneHot	Decision Tree	0.24	0.39	±0.038
Female	OneHot+KNN	KNN	0.2	0.33	±0.005

Tbl 7. Balanced Accuracy as target metric

Dataset	Strategy	Algorithm	Baseline	Score	Std
Complete	KNNImputer	Decision Tree	0.24	0.32	±0.017
Male	OneHot+KNN	Decision Tree	0.24	0.34	±0.04
Female	OneHot+KNN	KNN	0.2	0.33	±0.01

Tables 6 and 7 show the best algorithms for predicting the target variable "Category" we see a significant improvement over the baseline for all 3 datasets. Decision trees hold the majority. Note that a single decision tree is very susceptible to overfitting and therefore likely to perform poorly on new data. We recommend looking for alternatives.

E. Specific Diagnosis, Multiclass Classification (12 classes, collapsed from 27 classes)

Tbl 8. Macro F1-Score as target metric

Dataset	Strategy	Algorithm	Baseline	Score	Std
Complete	KNNImputer	KNN	0.06	0.2	±0.021
Male	OneHot+KNN	XGBoost	0.09	0.19	±0.017
Female	OneHot+KNN	SVM	0.08	0.19	±0.015

Tbl 9. Balanced Accuracy as target metric

Dataset	Strategy	Algorithm	Baseline	Score	Std
Complete	No OneHot	XGBoost	0.06	0.18	±0.02
Male	No OneHot	Decision Tree	0.1	0.22	±0.017
Female	OneHot+KNN	SVM	0.08	0.24	±0.043

The prediction of the specific diagnosis can be strongly improved from the baseline across all three datasets. For the complete dataset, the improvement is 14% using the KNN algorithm with F1 score as the target metric. In contrast to the other target metrics, models that are not based on trees also perform well. For the prediction of the dataset with female patients, the SVM algorithm is best suited with an improvement of 16%.

Conclusion

The data set provided proved to be difficult to classify. No clearly separable clusters could be identified. A strong overlap of the groups quickly leads to overfitting. Therefore, we tried to minimize this effect via cross-validation. As a result, in most cases only a slight improvement over the baseline is evident. The division of the data set into female and male subgroups led to a better prediction for some target features. The target features partly show a strong class imbalance. We propose to minimize this imbalance by either collecting more data or by downsampling the majority classes. However, this minimizes the number of available observations. An upsampling is not recommended due to the small number of observations in certain classes. Furthermore, we were able to determine important features, especially per location. We refer here to the section "Results". Tree based algorithms have consistently produced the best results and are also the easiest to interpret. We suggest to further refine the models identified here, including corresponding important features, and to consolidate them, for example, by means of permutation tests and further statistical tests.

Acknowledgment

We are grateful to Prof. Dr. Arzu Çöltekin and Dr. Fernando Benites for their organization and technical support throughout the project. We would also like to thank Prof. Dr. Nida Şen, Dr. Shilpa Kodati, and Dr. David Kuo, who not only provided us with the relevant data, but also assisted us with technical and medical questions regarding the data.

References

[1] S. Muñoz-Fernández and E. Martín-Mola, “Uveitis,” *Best Practice & Research Clinical Rheumatology*, vol. 20, no. 3, pp. 487–505, Jun. 2006, doi: 10.1016/j.berh.2006.03.008.

[2] R. Bansal, V. Gupta, and A. Gupta, “Current approach in the diagnosis and management of panuveitis,” *Indian J Ophthalmol*, vol. 58, no. 1, Art. no. 1, 2010, doi: 10.4103/0301-4738.58471.

[3] “scikit-learn machine learning in Python — scikit-learn 0.24.2 documentation.” https://scikit-learn.org/stable/ (accessed Jun. 25, 2021).

[4] P. M. Alan, B. H. Feldman M.D., J. Hung, J. H. Tsai, and Dr. K. Hossain, “Scleritis - EyeWiki,” Mar. 12, 2021. https://eyewiki.aao.org/Scleritis (accessed Mar. 12, 2021).

[5] S. Hn, D. J, U. D, F. A, C. Cc, and G. Da, “Gender disparities in ocular inflammatory disorders,” *Current eye research*, vol. 40, no. 2, Feb. 2015, doi: 10.3109/02713683.2014.932388.

[6] A. Bilogur, *ResidentMario/missingno*. 2021. Accessed: Jun. 22, 2021. [Online]. Available: https://github.com/ResidentMario/missingno

[7] M. Mukaka, “A guide to appropriate use of Correlation coefficient in medical research,” *Malawi Med J*, vol. 24, no. 3, pp. 69–71, Sep. 2012.

[8] R. E. Smith, R. A. Nozik, and G. Grabner, “Pseudouveitis („Maskerade-Syndrome“),” in *Uveitis: Klinik, Diagnose, Therapie Ein Leidfaden für die Praxis*, R. E. Smith, R. A. Nozik, and G. Grabner, Eds. Berlin, Heidelberg: Springer, 1986, pp. 238–241. doi: 10.1007/978-3-642-70809-1_38.

[9] “sklearn.impute.SimpleImputer — scikit-learn 0.24.2 documentation.” https://scikit-learn.org/stable/modules/generated/sklearn.impute.SimpleImputer.html (accessed Jun. 25, 2021).

[10] “sklearn.impute.KNNImputer — scikit-learn 0.24.2 documentation.” https://scikit-learn.org/stable/modules/generated/sklearn.impute.KNNImputer.html (accessed Jun. 23, 2021).

[11] “sklearn.preprocessing.OneHotEncoder — scikit-learn 0.24.2 documentation.” https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.OneHotEncoder.html (accessed Jun. 23, 2021).

[12] “sklearn.neighbors.KNeighborsClassifier — scikit-learn 0.24.2 documentation.” https://scikit-

learn.org/stable/modules/generated/sklearn.neighbors.KNeighborsClassifier.html (accessed Jun. 25, 2021).

[13] "sklearn.tree.DecisionTreeClassifier — scikit-learn 0.24.2 documentation." https://scikit-learn.org/stable/modules/generated/sklearn.tree.DecisionTreeClassifier.html (accessed Jun. 25, 2021).

[14] "XGBoost Documentation — xgboost 1.5.0-dev documentation." https://xgboost.readthedocs.io/en/latest/ (accessed Jun. 25, 2021).

[15] "sklearn.ensemble.AdaBoostClassifier — scikit-learn 0.24.2 documentation." https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.AdaBoostClassifier.html (accessed Jun. 25, 2021).

[16] "sklearn.svm.SVC — scikit-learn 0.24.2 documentation." https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVC.html (accessed Jun. 25, 2021).

[17] "sklearn.neural_network.MLPClassifier — scikit-learn 0.24.2 documentation." https://scikit-learn.org/stable/modules/generated/sklearn.neural_network.MLPClassifier.html (accessed Jun. 25, 2021).

[18] "Precision-Recall — scikit-learn 0.24.2 documentation." https://scikit-learn.org/stable/auto_examples/model_selection/plot_precision_recall.html (accessed Jun. 25, 2021).

[19] J. Opitz and S. Burst, "Macro F1 and Macro F1," *arXiv:1911.03347 [cs, stat]*, Feb. 2021, Accessed: Jun. 25, 2021. [Online]. Available: http://arxiv.org/abs/1911.03347

[20] K. H. Brodersen, C. S. Ong, K. E. Stephan, and J. M. Buhmann, "The Balanced Accuracy and Its Posterior Distribution," in *2010 20th International Conference on Pattern Recognition*, Aug. 2010, pp. 3121–3124. doi: 10.1109/ICPR.2010.764.