# Methods

Past tense

Improvements

* Define paramaters for models, this can be done by printing the models used in the workflow to show the paramates
* Tom’s input

## General computational details

The workflow constructed was constructed in the jupyter-lab environment using the programming language python. All models used sklearn default settings and parameters unless stated otherwise. An Ubuntu virtual machine hosted by azure was used for Optimisation of hyperparamaters. The virtual machine contained x amount of random-access memory and x number of cores.

## Data collection method

**\*Tom\***

## Sites sampled from

**\*Tom\***

Laser ablation

Laser info details

## Dimensionality reduction techniques

The structure of the ion abundance data was visualised in two dimensions *via* the dimensionality reduction techniques; Principal Component Analysis and t-distributed stochastic neighbour embedding.

### Principal Component Analysis

Modify code to print details of model

### t-distributed stochastic neighbour embedding

Modify code to print details of model

## Machine learning models overview

Supervised Machine learning classifiers were used to map profiles of ion abundances of flint samples to bedrock or superficial sites sourced in the United Kingdom. The models were trained on the optimum features as determined by the feature selection process. The labels into which samples were classified as were either single bedrock/superficial sites or groups of sites.

## Preproccessing and feature selection

### Data preproccessing

**\*Tom\* - what prepoccessing did you do?**

Missing values were imputed with the mean average for that feature. Outliers were defined as values that exceeded 2 times the standard deviation from the mean average for that feature. Outliers were replaced with the mean +/- 2 times the standard deviation.

### Feature selection

The optimum features were identified by the implementation of recursive feature elimination (RFE) with Random Forest Classifiers. Random Forest Classifiers were chosen because the model computes a feature importance score for each feature from the model. (explanation of feature importance). The process builds a model with all features then drops the feature with the lowest feature importance score. Another model is built without the dropped feature and again the feature with the lowest feature importance score is dropped and another model built with all features except the two dropped features. This is repeated until a model with one feature is built.

Each model was evaluated by 3-fold stratified cross-validation with weighted-F1 score. The mean average model scores were visualised against all the feature samples and the best feature combination was chosen for all subsequent models.

## Novelty detection

It was necessary to identify artefacts that were likely to have been sourced from a flint deposit not sampled in this study to prevent erroneous classification of artefacts. The method would indirectly permit the formation of an ‘other’ class. The Local Outlier Factor model was used for this purpose.

Samples that were classified as outliers were not classified by the final classifier but instead classified as ‘other’.

**\*print details of model\***

## Learning Curves

Random forest classifiers were built and evaluated on increasingly larger training datasets, each model was evaluated by stratified x-fold cross-validation with weighted-F1 scores.

## Model evaluation and hyperparameter optimisation

All models were extensively evaluated to compare performances. To assess performance the dataset was split into 80% training and 20% test data randomly 100 times. This process was stratified so that the proportions of classes in the training data was representative of the proportions within the entire dataset. Hyperparamater optimisation was computed on the 80% train data by 5-fold stratified cross-validation. The original 80% train data was then paramaterised with the optimum hyperparamaters and evaluated by weighted-F1 score on the test data.

The model performances were compared by the visualisation of the spread of the 100 weighted-f1 scores *via*  boxplots.

## Machine learning models

The performance of 4 different classifiers were assessed. These were Random Forest, Support Vector Machine, Gradient Boosting machine and x.

Random Forest outperformed all other models evaluated and so was used as the final model.

## Data Availability

\*follow guidelines for what to put here