**Status Report: Zircon fertility and magmatic intensity**

**Team Members: Faculty Advisors:**

1. Sai Phani Ram Popuri 1. Yingcai, Zheng
2. Kaethleen Siguenza 2. Jiajia, Sun
3. Ryan Nguyen
4. Hemanth Kumar Nichenametla

All the project code and documentation can be found here: [Link to Project GitHub](https://github.com/phaniram05/zircon-fertility-dsc)

**Project Objective:**

The main objective of this project is to predict magma fertility in porphyry copper deposits using Machine Learning. Porphyry copper deposits are copper ore bodies that are formed from hydrothermal fluids that originate from a voluminous magma chamber several kilometers below the deposit itself. Predating or associated with those fluids are vertical dikes of porphyritic intrusive rocks from which this deposit type derives its name from.

* Machine Learning Problem Type: Classification
* Balanced Dataset? Yes, the dataset classes are balanced.
* Performance Metrics in consideration: Precision, Recall, F1- score.

**Dataset Description:**

The dataset contains about 28 features representing the geographical information like the latitude and longitude, presence of trace elements like Titanium (Ti), Praseodymium (Pr), Holmium (Ho) etc., expressed as Parts Per Million (PPM). The complete list of features can be found in the table 1 below.

|  |  |
| --- | --- |
| **Name of the feature** | **Description** |
| Citation | References the research publication. |
| Location | Region in which the ore is present. |
| Latitude | Latitude coordinate of the location. |
| Longitude | Longitude coordinate of the location. |
| Rock Name | Name of the rock (ore) |
| Age Ma | Age of the magma (in million years) |
| Comment | Categorical feature expressing the ore to be barren or fertile. |
| Feature | Numerical encoding of comment.  0 = Barren, 1 = Fertile |
| Cu Tonnage Mt | Copper tonnage in metric units |
| Ti | Titanium trace in PPM |
| La | Lanthanum trace in PPM |
| Ce | Cerium trace in PPM |
| Pr | Praseodymium trace in PPM |
| Nd | Neodymium trace in PPM |
| Sm | Samarium trace in PPM |
| Eu | Europium trace in PPM |
| Gd | Gadolinium trace in PPM |
| Tb | Terbium trace in PPM |
| Dy | Dysprosium trace in PPM |
| Ho | Holmium trace in PPM |
| Er | Erbium trace in PPM |
| Tm | Thulium trace in PPM |
| Yb | Ytterbium trace in PPM |
| Lu | Lutetium trace in PPM |
| Y | Yttrium trace in PPM |
| Hf | Hafnium trace in PPM |
| U | Uranium trace in PPM |
| Th | Thorium trace in PPM |

**Table 1:** Description of features in the datasets

**Exploratory Data Analysis:**

To get more insights of the data, a set of box plots, heat maps and box plots have been drawn to identify the feature distributions, missing values, and the outliers respectively. Below figures show a sample of those findings.

A graph of a person with a blue line

Description automatically generated A graph of a distribution of a number of numbers

Description automatically generated with medium confidence

**Fig 1:** Distribution of age (ma) **Fig 2:** Distribution of “Hafnium” trace

A diagram of a box plot

Description automatically generated with medium confidence

**Fig 3:** Boxplot showing outliers in “Hafnium” trace.

A graph with numbers and lines

Description automatically generated with medium confidence A diagram of a box plot

Description automatically generated

**Fig 4:** Heat map of missing values **Fig 5:** Box plot showing outliers in “age” feature

**Data Preprocessing:**

Upon performing the exploratory data analysis, we have identified that most of the columns contain missing values and outliers. Furthermore, the dataset contains two geospatial data information namely latitude and longitude. We have plotted the kernel density estimators to identify the feature distributions.

**Note:** “Citation” feature is of little importance in our analysis. We have excluded it from the feature set for time being.

**Handling nulls in latitude and longitude columns:**

Since the features latitude and longitude are truth values, we cannot handle them using the regular null handling techniques. Therefore, we have collected their values based on their location respectively.

**Handling nulls in trace elements: (Ti, Ce, Pr etc.)**

Since these features belong to numerical category, and based on our observation, most of these features do not follow a standard normal distribution. As a first step, we have replaced their null values with the median of their fertility type respectively.

**Ex:** If for a record the “Hf” value is missing and the corresponding type of fertility is of “Barren”, then we replaced the missing value with the median “Hf” ppm of the “Barren” type of all the records present inside the data set.

**Outlier Handling:**

Based on the boxplots drawn, we have identified that almost every trace element has quite a few outliers to handle. Before jumping into the conclusion that the marks in the boxplot are outliers, we have tried finding the legitimate range of trace elements for barren and fertile ores but unable to find standard ranges. Therefore, we have decided to use Inter-Quartile Range (IQR) technique to handle them gracefully.

The IQR approach follows the below steps in handling outliers:

1. Calculate the first quartile (Q1) and third quartile (Q3) of the data.

2. Calculate the IQR as the difference between Q3 and Q1: IQR = Q3 - Q1.

3. Define the lower and upper bounds for outliers as

* Lower bound: Q1 - 1.5 \* IQR
* Upper bound: Q3 + 1.5 \* IQR

4. Any data points falling below the lower bound or above the upper bound are considered outliers and replaced with the lower bound / upper bound respectively.

Below images show the box plots of the feature “age (ma)” before and after handling the outliers:

A diagram of a box plot

Description automatically generated A diagram of a box plot

Description automatically generated

**Fig 6:** Box plot of “age (ma)” before handling outliers. **Fig 7:** Box plot of “age (ma)” after handling outliers.

To summarize, we checked for nulls and outliers in the raw features and handled them before building the machine learning models.

**Feature Engineering:**

As discussed earlier, we have latitude and longitude columns in our data set. There are many possibilities these features can be engineered to use in the model building. The one approach we considered is to cluster the data points based on the characteristics and use the cluster ID to replace the latitude and longitude columns. To enable this, we made use of agglomerative clustering technique to group the data points and the corresponding cluster IDs have been obtained.

**Reference:** Agglomerative Clustering

<https://scikit-learn.org/stable/modules/generated/sklearn.cluster.AgglomerativeClustering.html>

The below scatter plot shows the clustered data points based on their latitude and longitude coordinates.

**Note:** The number of clusters have been chosen arbitrarily as 5. We are planning to formulate an elbow plot to find the optimal number of clusters.

A chart with colored dots

Description automatically generated

**Fig 8:** Clustered data points based on latitude and longitude**.**

**Feature Correlation:**

To understand the impact of features over the target variable, a correlation matrix has been drawn and can be seen below:

A graph with red and blue squares

Description automatically generated

**Fig 9:** Feature Correlation

The color scale next to the heat map indicates the intensity of relation between the variables. It can be observed from the graph that most of the trace elements are highly positively correlated to each other. Therefore, it is advisable to use a subset of those trace elements in the final model building to improve the performance.

**Note:** We are working on feature selection techniques as on date. The techniques under consideration include Forward Feature Selection, Backward Feature Selection, Feature Selection based on a statistical measure like P-value.

**Model Building:**

We have started experimenting with the refined data by building a simple logistic regression (LR) model considering all the features. The results obtained can be seen below with the baseline model, without any hyper-parameter tuning.

**Note:** Just the baseline model with all the features.

**Logistic Regression Model:**

Before feeding the processed data to the model, the following operations have been performed in sequence over the data and can be found in the table below:

|  |  |
| --- | --- |
| **Operation** | **Description** |
| Train-test Split Ratio | 80:20 to evaluate the model performance. |
| Feature Scaling | Standard scaler has been used for quicker convergence of the algorithm. |

The training data has been fit to the scikit-learn library’s Logistic Regression model.

**Performance metrics evaluated over test set:** Accuracy, Precision, Recall, and F1-Score.

A chart of different colors

Description automatically generated A number of numbers on a white background

Description automatically generated

**Fig 10:** Confusion Matrix **Fig 11:** Classification Report

**Action Items to be done:**

1. Perform feature selection to reduce the dimensionality of the data.
2. Hyper-parameter tuning using the validation set.
3. Explore different classification models like the Decision Trees and Random Forest.
4. Model Comparison
5. Poster preparation

**References:**

1. <https://scikit-learn.org/stable/user_guide.html>
2. <https://medium.com/@khadijamahanga/using-latitude-and-longitude-data-in-my-machine-learning-problem-541e2651e08c>
3. <https://heartbeat.comet.ml/working-with-geospatial-data-in-machine-learning-ad4097c7228d>