Arica Soil Property Prediction and Clustering
(Graduate Project Report ECE/CS 498 Sp19)

I. INTRODUCTION

A. Background

Soil functional properties are those properties related to a soil's capacity to support essential ecosystem services such as primary productivity, nutrient and water retention, and resistance to soil erosion. Extracting such soil functional properties such as the pH of the soil, essential minerals content etc., requires tedious procedures like collecting soil samples and then performing various chemical tests on those samples which requires a lot of time and cost incurred in these methods may be high too. Hence, it would be of paramount importance to people in the agricultural domain to find cost efficient methods to deduce about the soil functional properties.

B. Problem Statement

To reduce the time and effort to extract soil functional properties infrared spectroscopy techniques can be used to observe certain properties of the soil which may be more accessible to the farmers and may not be as expensive as performing chemical tests and hence being convenient to predict the soil functional properties and in turn check the quality of the soil and take the necessary actions such as to keep in check the mineral content and other factors which may affect the growth of a particular crop and may also help in choosing particular soil types for different crops. Also, we aim to extract some structure in the soil functional properties like extract clusters of the soil functional properties which might be useful to classify the soil type for different types of crops and also try to extract some structure or clusters from the spectroscopy data and then try to establish a relationship between the two if there is any which might be useful to check if the soil is of a particular type without having to observe the particular soil functional properties.

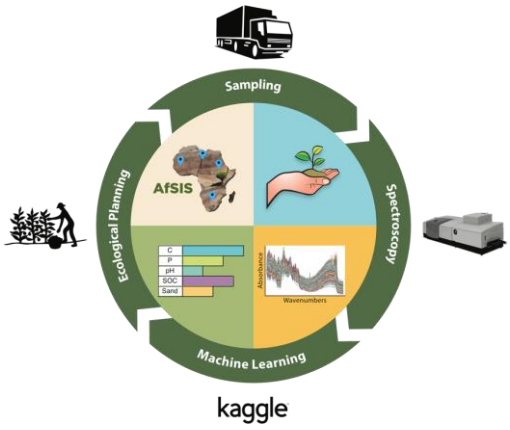

Figure 1: Kaggle Challenge

The target audience would be the farmers who might want to know if the soil in a particular area is good for a typical crop such as the soil properties may vary during different seasons and hence may affect the growth of different crop types and also the farmers who would want to keep a check on the condition of the soil for the growth of healthy crops and take the necessary action if need be.

C. Data Description

The dataset is available on https://www.kaggle.com/c/afsis-soil-properties/data. The training data consists 1158 samples of 3,578 features from the spectroscopy analysis of the soil and one feature which describes the depth from which the soil sample. It also consists of the soil functional properties: pH, Ca, P, SOC, Sand which would be needed to predict. The test data consists of 728 rows for evaluating the model.

II. RELATED WORK

We accessed the related work from the top solutions available on the Kaggle discussion board. They had used some complex signal processing techniques. Moreover, we concluded from their solutions that data is prone to overfitting and no single regression algorithm worked in a better way. All of them had used some sort of ensemble method to avoid overfitting and improve RMSE score.

III. METHODS

The main aim of this project is to use the raw spectroscopy data to predict the values of certain properties of the soil, namely, SOC, pH, Ca, P with a reasonable accuracy. As, the spectroscopy data is high dimensional we will be exploring the methods learnt in the class to perform dimensionality reduction as well as to do prediction. After prediction, we would like to explore for some structure in the data (which seems likely) and possibly come up with clusters of soil functional properties based on which certain soil types maybe classified. It is also possible that we may be able to say about the cluster of soil functional properties after observing the spectroscopy data if some similar structure exists in the data. We would be using the unsupervised learning techniques learnt in the class throughout the semester. The end result should help the target audience, that is, the farmers who are looking for accessible and relatively inexpensive methods to aid the healthy growth of their crop and also to throw light upon the soil types which may or may not be useful for the healthy growth of the particular crop.

A. Exploratory Data Analysis

We used python libraries and Jupyter notebook environment

for this project. The training data was loaded and then visualized to get better understanding. No missing values were found. The data didn't require much preprocessing and cleaning. The features for train data are shown in Figure 2 and Figure 3. The distributions of target variables are shown in Figure 4.

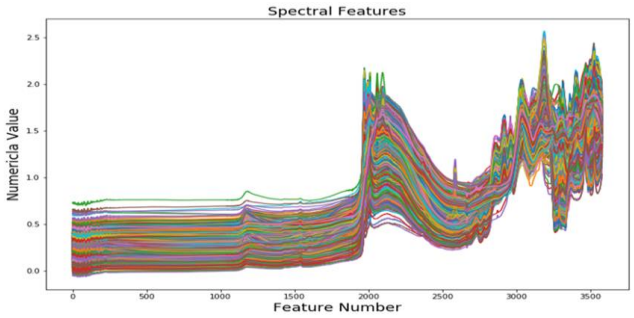

Figure 2: Spectral Features Distribution

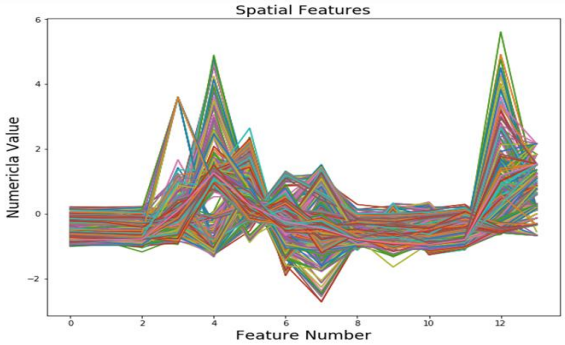

Figure 3: Spatial Features Distribution

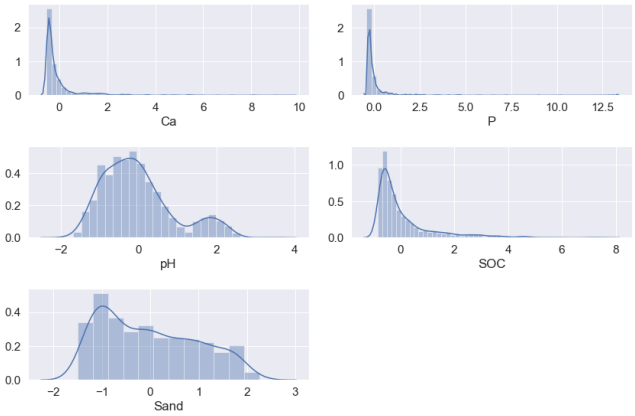

Figure 4: Target Variables Distributions

B. Data Pre-Processing

The data given here is a very high dimensional data as it contains 3579 features to predict the values of our target variables. To make the problem feasible we will need to use dimensionality reduction first and then use these reduced features to predict the target variables. To perform dimensionality reduction on the spectroscopy data, we will be using Principal Component Analysis.

1) PCA

As our first choice, we used PCA for spectral features. We recovered 99% of the variance in the first 9 principal components and hence would be using those. We tried PCA on spatial features as well but it didn't result in any significant dimensionality reduction. We decided to use 9 PCs and 16 spatial features for building prediction models.

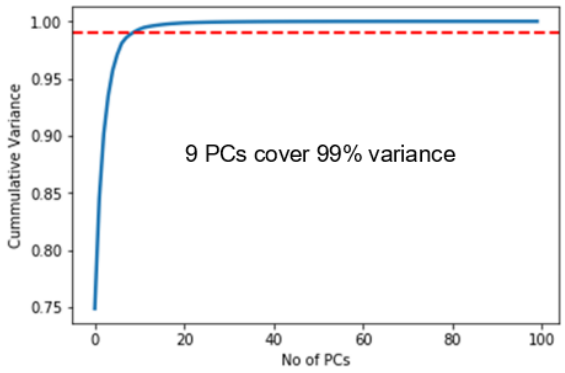

Figure 5: PCA Results on Spectral Features

2) Feature Selection

Feature selection, also known as variable selection, attribute selection or variable subset selection, is the process of selecting a subset of relevant features (variables, predictors) for use in model construction. Feature selection is different from feature extraction as feature extraction creates new features, but feature selection uses a subset of features. One of the simple techniques to do feature selection is using correlation- based feature selection. It is based on the hypothesis that the best features for a prediction problem are those which have the highest correlation with the target variable as well as have little to no correlation among themselves. We used several techniques to do feature selection which include correlation based and univariate feature selection using the scikit-learn feauture selection module. Correlations among features and target variables are shown in Figure 6.

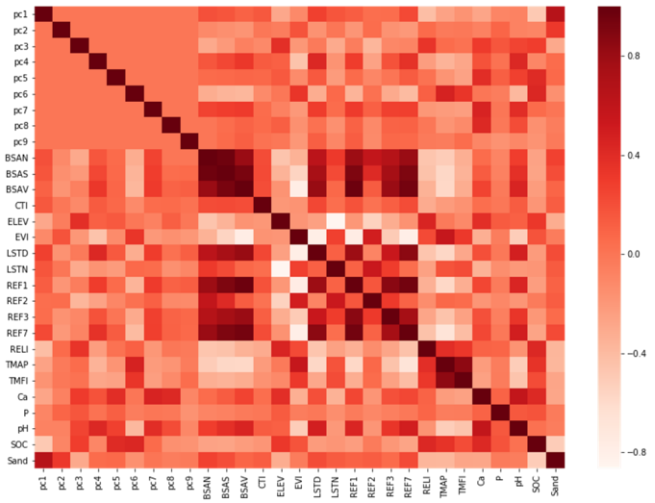

Figure 5: Correlations among Features & Target Variables

TABLE 1: FEATURE SELECTION METHODS

Based on Correlations
Select K Best
Select From Model
RFE
Feature Importance

Ca		P		pH		SOC		Sand	
pc7	0.459761	pc3	0.163206	EVI	0.513990	pc1	0.483247	pc1	0.661366
pc8	0.421471	pc9	0.161984	TMAP	0.508921	pc6	0.433818	TMAP	0.373788
pc5	0.395007	ELEV	0.143572	REF7	0.495445	RELI	0.428597	RELI	0.365227
ELEV	0.381699	LSTN	0.136926	LSTD	0.486732	pc5	0.405590	pc2	0.321864
LSTN	0.345367	pc5	0.119917	BSAV	0.453661	TMAP	0.355615	ELEV	0.314198
EVI	0.329683	REF2	0.102864	REF1	0.441190	ELEV	0.325435	pc3	0.292112
pc3	0.306172	pc2	0.095613	pc4	0.429878	pc3	0.287125	BSAN	0.265360
pc1	0.270517	RELI	0.091592	pc7	0.392337	BSAS	0.264505	REF7	0.226372
BSAV	0.259323	pc1	0.086123	pc6	0.382194	REF7	0.264103	LSTD	0.217102
REF1	0.245187	BSAN	0.082305	BSAS	0.371715	BSAN	0.253075	BSAS	0.211578

Figure 6: Correlation based ranking of features

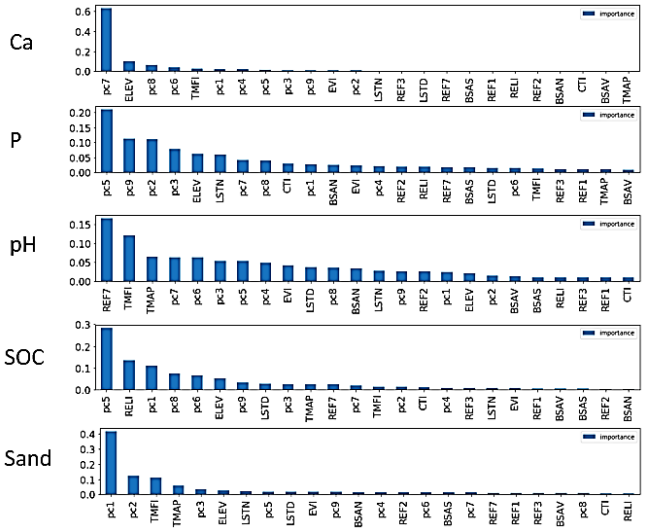

Figure 7: Relative Feature Importance

The feature selection methods that we tried are listed in table 1. Based on correlations, top 10 features for each target variable are shown in Figure 7. Random Forest Regression algorithm also gives information about the feature importance. The feature importance obtained from random forests are shown in Figure 8. We observed that feature selection methods didn't result in a significant improvement in rmse score. The performance of linear models increased slightly but feature selection degraded the performance of non-linear models. So, we decided to ignore feature selection and use 25 features for final models training.

3) Feature Scaling

Before training any model, feature scaling needs to be performed as all the features maybe on different scales. It is necessary to perform feature scaling as all the models which we will be using would require gradient based optimization which may give spurious results if all the features are not on the same scale.

C. Regression Models

After feature extraction, we will be using the extracted features to use supervised learning techniques to build and train our model to predict the values of the 5 target variables, namely, pH, P, Ca, SOC, Sand using the training data. We will then be using the test data to perform prediction and cross validation of our model for hyper parameter tuning.

TABLE 2: REGRESSION MODELS

Linear Models	Linear Regression
	Polynomial Regression
	Ridge Regression
	Lasso Regression
Non-Linear Models	KNN Regression
	SVR - Linear
	SVR - Poly
	SVR - RBF
	Decision Trees
Ensemble Models	Bagging Regressor
	Random Forest Regressor
	Extra Tree Regressor
	AdaBoost Regressor
	Gradient Boost Regressor

We first used Linear models to predict the target variables and these models performed very poorly as the problem was clearly a non-linear problem. Linear models included: Linear Regression, Ridge Regression and Lasso Regression.

We then moved on to non-linear models like Support Vector Regression and K-nearest neighbors which also weren't performing up to the mark and hence we had to resort to ensemble methods to get a reasonable accuracy. All the models used in the project are summarized in Table 1.

D. Cross Validation & Hyperparameter Tuning

Cross-Validation techniques are used to tune the hyperparameters of the model to improve the generalization error or the test error of the model for the given prediction problem. For example, for a Support Vector Regressor or Support Vector Classifier for the Radial-Basis Function Kernel (RBF Kernel), we have two parameters, one is 'C' which is responsible for controlling the overfitting of the model and the other is 'gamma' which is the reciprocal of 'sigma' used in the radial-basis function. Also, as we had 5 target variables, i.e., 5 prediction problems we had to select the best model for each of these 5 target variables. We used K-fold cross validation to validate our models to select the best hyperparameters of the models and selected the optimal models as the ones which had the lowest Root Mean Squared Error.

E. Stacking of Models

Using only a linear or non-linear model was not enough to give satisfactory results due to the scarcity of data and the problem of overfitting. To overcome this challenge we had to resort to ensemble methods using boosting. The idea of boosting can be explained using bagging methods which relies on the fact that if we take a weighted average of a number of independent classifiers then the accuracy of such a model increases and also it eliminates the problem of overfitting by taking random

subset of samples without replacement. But, in bagging we waste some amount of data which might not be ideal for training models and hence, boosting tries to make use of more data while keeping in mind the idea of classifiers being independent by randomly choosing subset of data with replacement and training a classifier on it and then taking the weighted average.

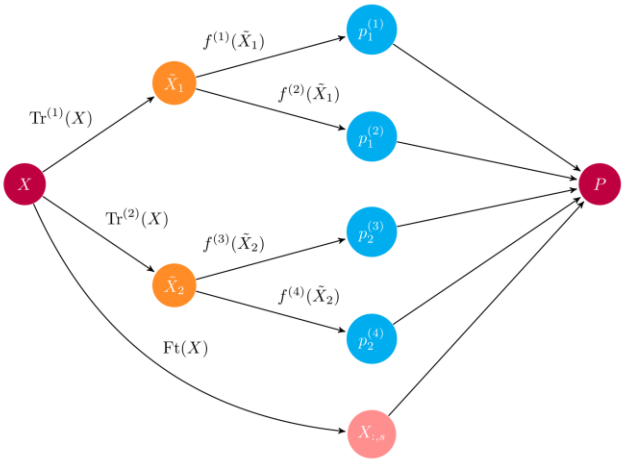

Figure 8: Example schematic of an ensemble

After going through a various number of steps, the model which gave the best results on the test data is an ensemble model implemented using the ml-ensemble library. We used a novel ensemble technique which uses several linear and non-linear models as input to another layer of non-linear models to do prediction. For training we followed the procedure described above and for doing the prediction we first projected the data onto the components which were extracted using Principal Component Analysis on the Training data. After preprocessing, we used those extracted features to feed into each of our trained model. We used different linear and non-linear regression algorithms and did hyperparameter tuning using Grid-Search-CV to get optimal models. Data was fed to first layer of base models whose predictions were input to 2nd layer of models and then finally their predictions to final model layer.

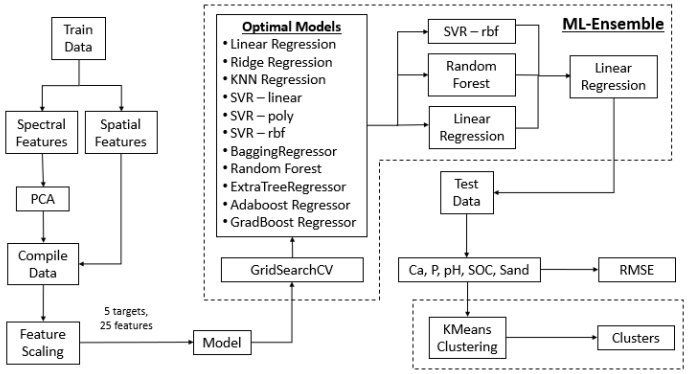

Figure 9: Block diagram for solution approach

The procedure followed for the training of the model and then predicting the values is summarized in the figure 11.

IV. RESULTS & DISCUSSION

The rmse scores using cross validation on train data are summarized in table 3. It contains the validation mean scores for individual optimal models for each target variable and for final ensemble model as well.

TABLE 3: RMSE SCORES FOR REGRESSION MODELS

RMSE Scores	Ca	P	pH	SOC	Sand
LinearRegression	0.474129	0.509118	0.514583	0.626524	0.577974
Ridge	0.477349	0.509295	0.513086	0.627081	0.574445
KNN	0.416879	0.618948	0.524237	0.625840	0.386477
SVR_Linear	0.520894	0.490988	0.514074	0.662105	0.575716
SVR_Poly	0.344661	0.472887	0.456611	0.510234	0.426289
SVR_RBF	0.315481	0.496421	0.416152	0.432565	0.281697
BaggingRegressor	0.460271	0.360970	0.494940	0.604393	0.395441
RandomForestRegressor	0.306916	0.383201	0.454969	0.607998	0.356133
ExtraTreeRegressor	0.307119	0.387393	0.419065	0.522881	0.341521
AdaBoostRegressor	0.382999	0.395236	0.435237	0.492157	0.379222
GradientBoostingRegressor	0.385355	0.372973	0.451079	0.559635	0.342491
ML-Ensemble	0.285576	0.438138	0.398420	0.497032	0.313649

For training data, the predicted values were plotted against the actual values for each target variable using the ensemble model. These plots are shown in figure 10.

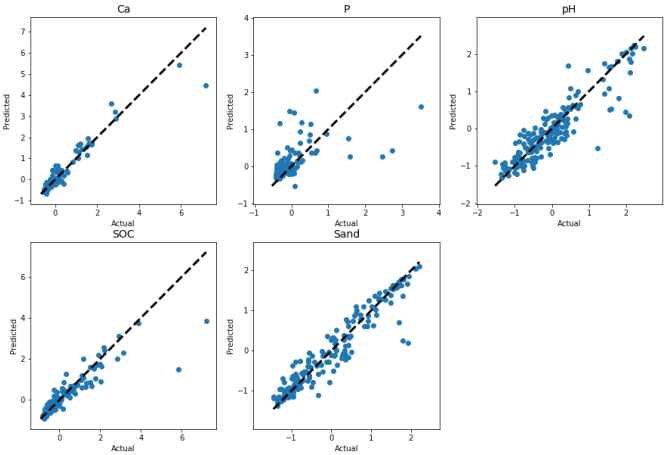

Figure 10: Predicted vs. Actual Values

It can be seen that in most of the target variables, non-linear models performed better than the linear models. From the non-linear models particularly SVR-rbf and random forests performed better. That was the motivation to ignore feature selection methods as well. We had already reduced our feature dimension to 9 PCs. So we decided to go with 25 features. Ensemble methods performed better. That encouraged us to go one step further where we created an ensemble by stacking different models. It can be seen that score of final stacked model was better than the individual models.

Individual regressions might be capturing a particulalr aspect of the feature space. By creating a stack of models, we are combining the strengths of different models. Moreover,

creating an ensemble also helps us to reduce the effects of overfitting. Final stacked model can be thought of as a weighted average of predictions from individual models.

The best score on Kaggle was calculated using a metric called MCRMSE which is basically the mean over the 5 mean squared errors calculated and is calculated as follows:

$$\text{MCRMSE} = \frac{1}{5}\sum_{j=1}^{5}\sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_{ij} - \hat{y}_{ij})^2}$$

We checked the scores on Kaggle test data for each individual model and for ensemble models as well. Our best score for MCRMSE on the Kaggle website was 0.5601 which we obtained through our ensemble model. It also validates our reasoning that ensemble of models should be created to avoid overfitting and combine strengths of different models.

The MCRMSE score that won Kaggle competition was 0.46892. As per our observations, there was still some room for improvement in 2nd target variable 'P'. Moreover, we suggest combining some signal processing techniques from Kaggle solutions and our approach of stacking models to improve the score even better than the Kaggle final score.

V. Relationships among Soil Properties

A. Correlation among Target Variables

After solving the prediction problem we explored some structure in the target variables as they seemed to be correlated to each other. For example, the amount of Calcium (Ca) does influence the pH of the soil, as Calcium is Basic and increasing the quantity of Calcium in the soil would increase the pH of the soil. We first looked at the correlation values of these target variables and found some interesting results:

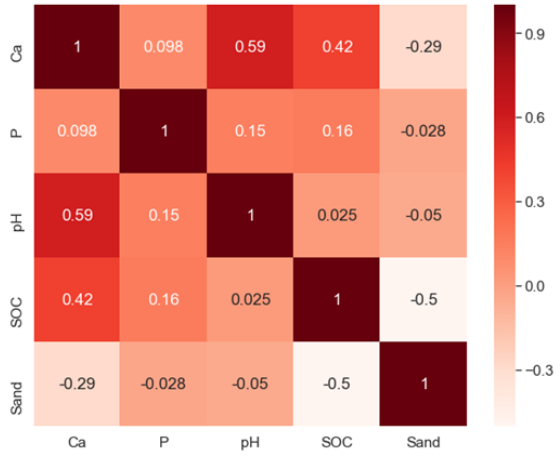

Figure 11: Correlations among soil properties

B. Clustering of the Target Variables

After exploring the correlations among these variables we sought to perform a hard clustering on the 5 target variables. These clusters (if well-formed) would signify different soil types and values of the functional properties which lie in these clusters could be represented by an error term around the cluster centroids. The clustering of these values could help in deducing the values of target variables if we have limited information about one or more of them. We used the elbow method for choosing the optimal number of clusters and found out that 8 clusters are the optimal number of clusters for this type of data. Also, we have labeled these clusters according to the range of values of target variables of the cluster centroids.

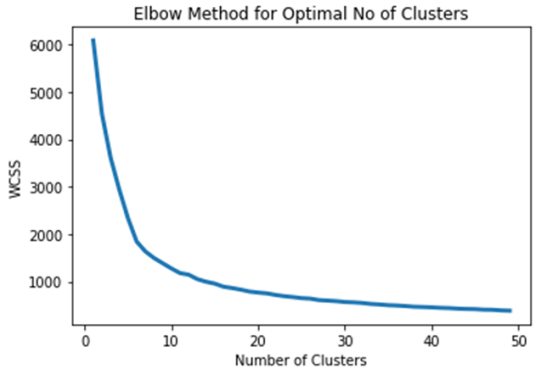

Figure 12: Optimal number of clusters for soil properties

TABLE 4: CLUSTERS CENTERS FOR SOIL PROPERTIES

Ca	P	pH	SOC	Sand	Cluster	Description
-0.377770	-0.219385	-0.449092	-0.388017	-0.074105	0	Low-Ca Low-P Neg-pH Low-SOC Mid-Sand
-0.040148	-0.124162	-0.574790	2.896514	-0.928389	1	Low-Ca Low-P Neg-pH High-SOC Low-Sand
0.747014	0.068563	1.537137	-0.304553	-0.389595	2	Mid-Ca Mid-P Pos-pH Low-SOC Mid-Sand
0.258263	3.141500	0.430292	0.317031	-0.174210	3	Mid-Ca Mid-P Pos-pH Mid-SOC Mid-Sand
4.859856	-0.190742	1.900154	2.483493	-0.967938	4	High-Ca Low-P Pos-pH High-SOC Low-Sand
-0.404145	-0.172640	-0.130470	-0.609670	1.282962	5	Low-Ca Low-P Neg-pH Low-SOC High-Sand
-0.137051	-0.209962	-0.449387	0.459577	-0.983348	6	Low-Ca Low-P Neg-pH Mid-SOC Low-Sand
1.040350	9.840077	0.638322	2.813268	-0.654799	7	Mid-Ca High-P Pos-pH High-SOC Low-Sand

VI. Conclusion

At the end of this project we have 5 ensemble models which predict the values of soil functional properties with an MCRMSE score of 0.5601. The winning score of this competition was 0.4609 which also uses 5 ensemble models which are an ensemble of neural network models but with different pre-processing techniques which are prevalent in signal processing. Therefore, this highlights an important point that efficient pre-processing techniques lead to improving the accuracy of the model. (As we do not lose much information in the process).

We also tried to explore for some structure in the target variables and found that there is a strong relationship between Ca pH and SOC values which means that if we are able to predict Ca values well then pH and SOC values would also be predicted well.

VII. Challenges

We faced several changes during the project. First, there were a smaller number of sample data points as compared to very high number of features. So, the data was high dimensional. In this case, overfitting is a common problem instead of getting a generalized model. To solve this challenge, we used PCA and

feature selection models. Luckily PCA worked for us and solve the problem of high dimensions without any major loss in variance. Moreover, there was noise in data and we were not sure whether the given features had any correlations with the target variables. For that purpose, we tried to combine strengths form different models and improve our score as much as we could. The next challenge was the limited time that we had for the project. We had to manage our time along with other courses. To solve this problem, we divided the tasks from the first day and set some timeline to complete those tasks. That helped us in completing the project on time.

VIII. Contributions

The individual contributions from team members are listed in the table 5.

Table 5: Contributions

Data Loading & Visualization	Shuyue
Data Preprocessing	Navjot
PCA	Qasim
Finding Correlations	Navjot
Feature Selection	Qasim, Navjot
Initial Models Training & comparison	Shuyue
hyperparameter Tuning	Navjot, Qasim
Ensemble Learning & Stacking Models	Qasim, Navjot
Training final models for each target variable	Navjot, Shuyue
Compiling Final Results	Shuyue, Navjot
Clustering	Shuyue, Navjot
Clusters Visualization	Navjot, Shuyue
Presentation	Qasim

References

[1] https://www.kaggle.com/c/afsis-soil-properties/overview
[2] https://scikit-learn.org/stable/
[3] http://ml-ensemble.com/
[4] http://rasbt.github.io/mlxtend/user_guide/regressor/StackingCVRegressor/
[5] https://www.kaggle.com/flennerhag/ml-ensemble-scikit-learn-style-ensemble-learning