PROJECT REPORT ML-IR : A Machine Learning Approach to Automated Infrared Spectra Interpretation

Motivation - The Problem

A Brief Overview of Infrared Spectroscopy for Characterization of Unknown Chemicals

As a chemistry major, one of the early topics covered in Organic Chemistry 1 is the identification and characterization of an unknown chemical. Being able to accurately identify a chemical in an unknown sample is incredibly important in a variety fields including pharmaceutical chemistry (determing if you've made your product of interest, quality control, etc), environmental chemistry (testing for the presence of environmental contaminants in a sample, tracking biomarkers), and astrochemistry (characterizing the gases present in nebulas, alien atmospheres, etc). Infrared (IR) Absorbance spectroscopy also known as Vibrational Spectroscopy has been used for many decades to identify specific functional groups in molecules and plays a key role in the identification of unknown chemical species. In this approach unknown chemical structures can be thought of as Lego sets and the infrared absorbance spectra reveals useful information about which pieces are used. Because each specific "piece" (for instance a Carbon bonded to an Oxygen [C=O]) has a different infrared absorbance (corresponding to the energy to vibrate that bond) the infrared spectra can be used to glean valueable information about key features present in a molecular system.

For Example the IR absorbance spectra for n-butanol shows a number of useful features corresponding to the different functional groups in Butanol (CH3-CH2-CH2-CH2-OH):

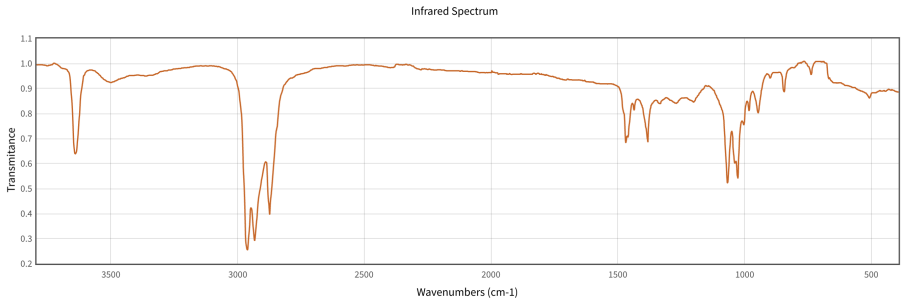

Figure 1: Butanol IR Transmission Spectrum

The y-axis of the chart represents the amount of light transmitted with given energy in units of wavenumbers on the x-axis. For butanol the sharp peak at ~3600 cm^{-1} corresponds to the O-H bond vibration and the broad features at

~3000 cm^{-1} correspond to C-H bond vibrations.

Translation - IR Spectroscopy With Machine Learning

In a ML sense, a functional group is a multi-label classifier (ie. one compounds IR spectra often can indicate the presence of multiple functional groups). Ideally, I should be able to feed in the IR-Absorbance Spectrum as a collection of features and the model should be able to output a multi-label prediction for each suspected functional group. Automated detection of functional groups and chemical unknowns in test samples allows automated testing stations for environmental contaminants or chemical product impurities.

Building the Dataset

In order to train a ML model to recognize IR absorbance features and map them to functional groups I developed a basis set of chemical systems that contain the functional groups of interest. To limit the scope and computational cost of this project I focus on small organic molecules only containing Carbon,Nitrogen,Hydrogen,and Oxygen. I also focus on the following functional groups: C-H, C=C, C≡C, C≡N, C-OH, C=O, C=OOH, N-H, C≡C−H.

A further issue is the lack of mass-downloadable data for IR Spectra, meaning that digitized spectra were downloaded manually from the NIST webbook. As such, ~100 Spectra were specifically chosen to form the training set (and accurately capture the functional gropups of interest). These spectra include 40 Systems with only one main functional group aside form C-H and 60 systems that largely mix different functional groups together. The dataset contains a variety of infrared spectra in both the gas, liquid, and solid phase; however, all spectra are over a similar range of infrared energies (500-4000 wavenumbers). The lowest resolution spectra has ~83 points. Functional group labels were added manually for each spectra and encoded in a binary fashion. Hexanol for instance got the following identificaiton [1,0,0,0,1,0,0,0,0] where index 0 and index 4 represent the C-H and O-H bonds respectively. Due to the small size of the dataset some classes were over-represented (C-H) while others were under-represented in the data (C=OOH). The following figure shows the counts of each functional group (class) in the dataset:

Preprocessing - Converting IR Absorbance Spectrum to Features

IR spectra were individually downloaded in "*.JCAMP" format from the National Institute of Science and Technology (NIST) which hosts free scientific data for a large selection of small organic molecules. Once downloaded, the continuous IR absorbance values were binned and interpolated to generate 83 features ranging from 500-4000 cm^{-1}. Once binned and interpolated the values were normalized for each spectra such that the largest absorbance intensity, I, in

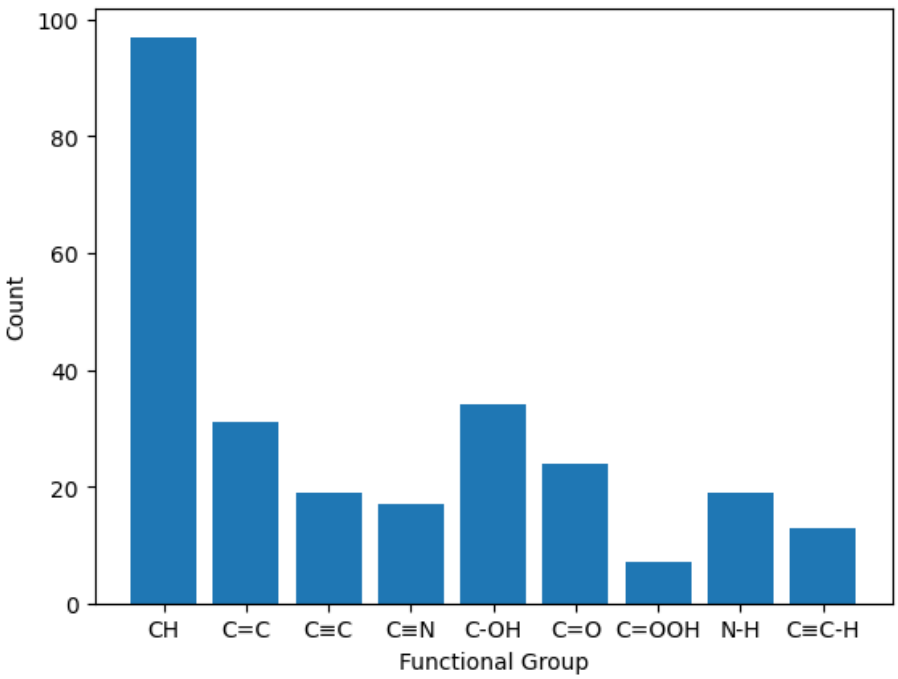

Figure 2: Functional Group Counts in Dataset

each spectra was set to (I/I) 1 and other intensities, i, were scaled to i/I. This normalization was performed to deal with various absorbance units and prevent features with differently scaled intensities from being over-represented in the data. Transmission spectra were also converted to absorbance spectra using the formula A = 1 - T where A is absorbance and T is transmission. Preprocessed experimental spectra were assigned appropriate functional groups and written to a master csv file, "Full_File.csv", with the following general format where A1-A84 are the binned absorbances and C-H ... are the functional groups.

Ex:

FileName	A1, A2, A3 ... A84	C-H, C=C ...
hexene	0, 0.1, 1 ... 0.1.	1, 1, ...

Once loaded into a csv file the data were read in as a pandas dataframe and split into training, testing, and validation sets of size 70, 15, and 15 respectively. Splitting the data presented another challenge, as simply using train_test_split(stratify=y) was not feasible due the number of possible class combinations and small dataset. For nine functional groups there are 2^9 combinations and stratify requires at least two instances of every class combination to evenly distribute instances. With 100 data points it was not possible to capture enough systems with duplicate functional groups for stratify to work. Instead MultilabelStratifiedKFold() was used to ensure adequate representation for each label in the test, train, and validation sets.

Model Selection

Since I am interested in a multi-label classification, it seems that I am relatively limited in terms of ML models as KNN is one of the only models that supports multi-label classification as a default option (simply choosing the best multi-label based on the most represented K neighbors). However, sklearn provides a solution that allows us to apply other classifier models to this problem. Specifically, the OneVsRestClassifer() module. This module takes a classifier model (decision tree, SVC, randomforest, etc) and trains one instance of the classifier model for each binary class in the data. Using this module, each sub-model is trained to only separate out key features for each binary class from the rest of the and the results of predictins of each sub-model for each class are combined to generate a multi-label prediction (ie. each sub-model contributes a 1 or 0 for each class). In our case there are 9 functional groups (binary classes) so the OneVsRestClassifier() module would generate 9 binary classifiers. The following models were selected to with a goal of covering a comprehensive range of different algorithms: KNN, Logistic Regression, Decision Trees, Random Forest, Linear SVC, RBF SVC. Lastly, an ensemble approach was used where the method that performed best on each functional group was selected to only predict that functional group.

Evaluation Metrics

The models were evaluated using the accuracy score and the f1 score of the train, test, and validation data. The accuracy score reflects whether the ML model accurately predicted the presence or absense of each of the 9 functional groups. Since there are nine possible labels per spectra, a single incorrect prediction or eight incorrect predictions are treated the same in this approach despite one model clearly outperforming the other. To deal with this limitation of accuracy scores, F1 scores were also used. F1 scores measure the average of the precision and recall. Precision can be defined by the folliwing equation: $P = \frac{True_P}{True_P + False_P}$ and measures the tendency of the model to predict false positives (ie. wrong functional groups). Recall can be defined by: $R = \frac{True_P}{True_P + False_N}$ and is measures the tendency of the model to predict false negatives (ie. miss functional groups). This combination of precision and recall provides a more balanced assessment of the model strength especially for a relatively imbalanced dataset where some functional groups (C-H) are over-represented. By calculating F1 scores for each class and averaging a total F1 score was generated for the data. Individual class performances were also assessed using the individual scores.

Hyperparameter Optimization

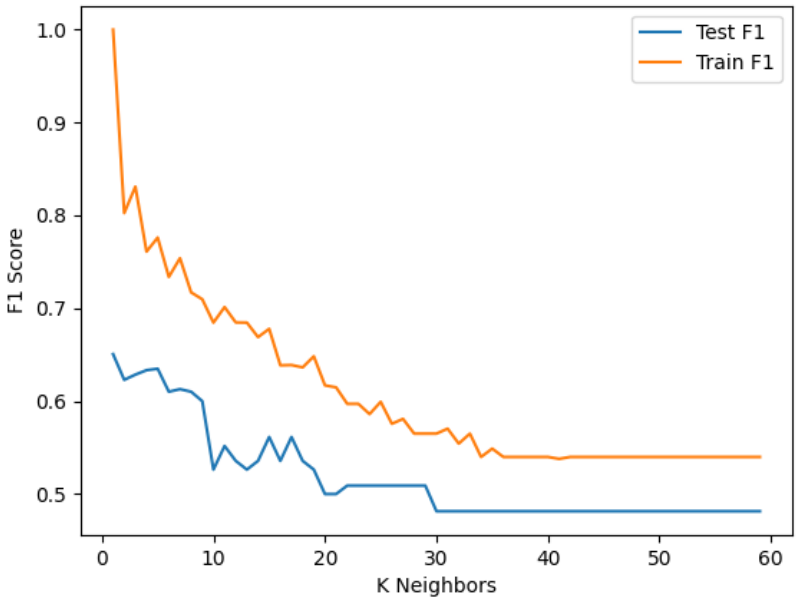

Figure 3: KNN Hyperparameter Optimization

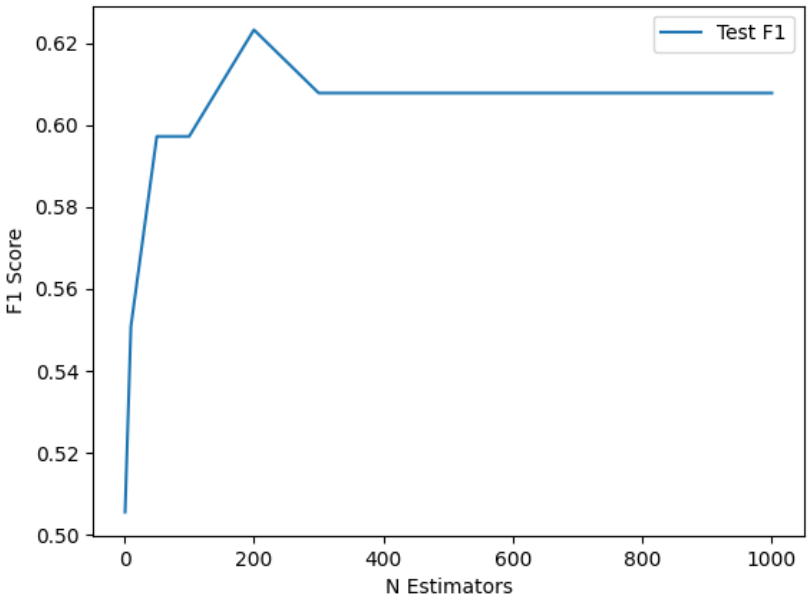

Figure 4: Random Forest Hyperparameter Optimization

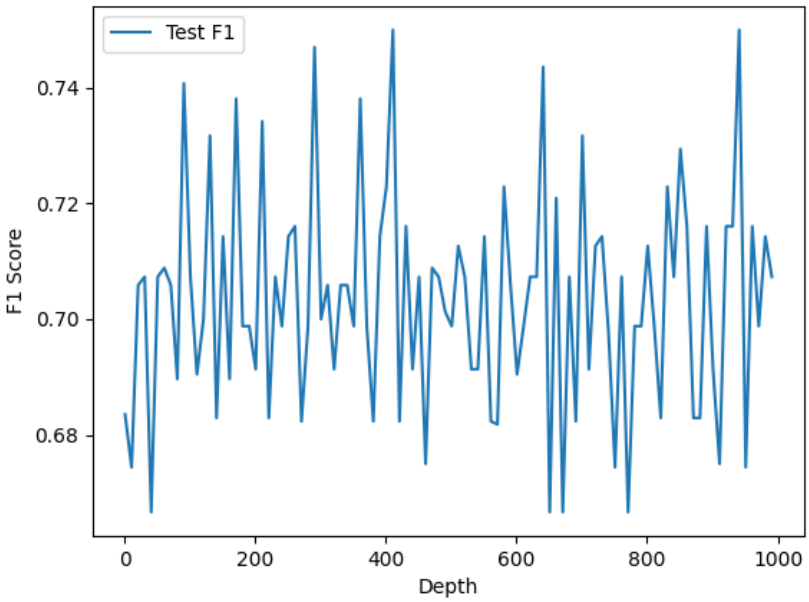

Figure 5: Decision Tree Hyperparameter Optimization

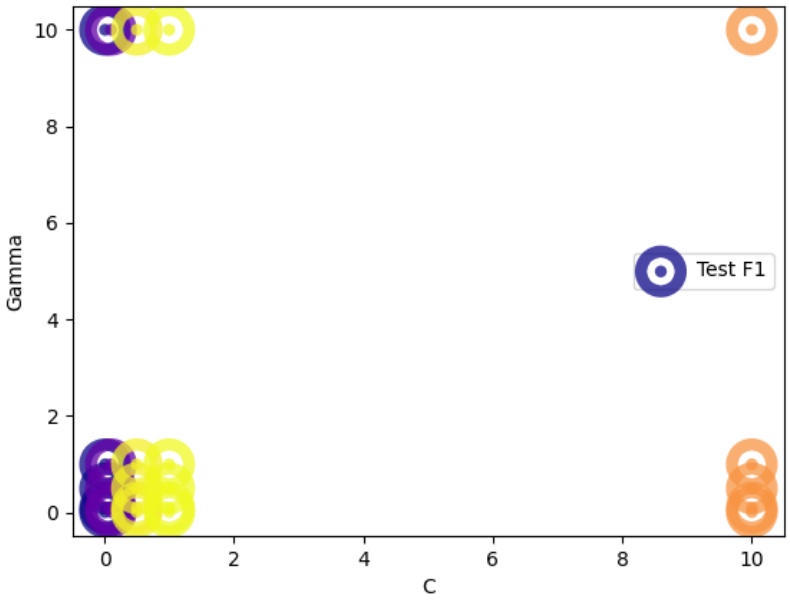

Figure 6: Linear SVC Hyperparameter Optimization

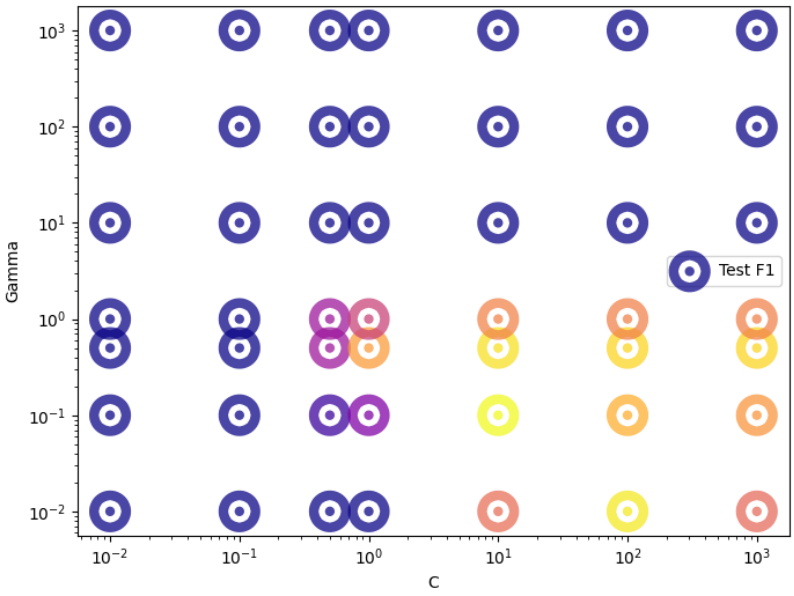

Figure 7: RBF SVC Hyperparameter Optimization

Hyperparameter optimization was performed for Random Forest, Decision Tree, Linear SVC, RBF SVC, and KNN machine learning models. The F1 Score calculated on the validation data was used as the optimization parameter when performing HPO. For models with more than 1 hyperparameter (SVC) a grid search was performed to determine the best combination of hyperparameters. The best set of hyperparameters are summarized in the table below:

Model	Hyperparameters Selected
KNN	n_neighbors=3
Decision Tree	max_depth=250
Random Forest	n_estimators=300
Linear SVC	C=1 ; Gamma=0.1
RBF SVC	C=10 ; Gamma=0.1

Results and Figures

Overall Model Performance on Validation Data

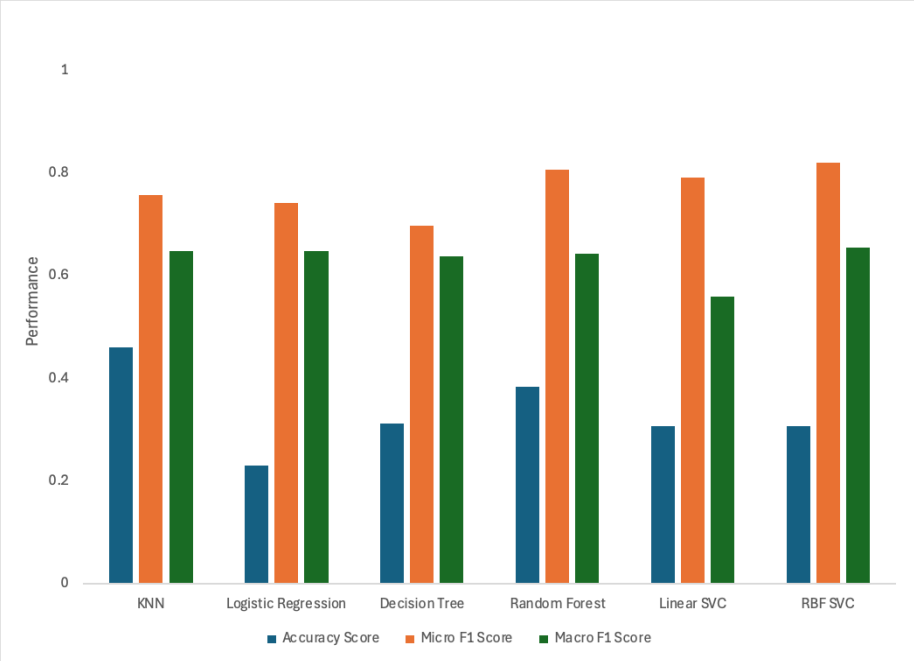

Figure 8: Average Accuracy Scores on Validation Data

Model	Accuracy Score	Micro F1 Score	Macro F1 Score
KNN	0.4615	0.7576	0.6487
Logistic Regression	0.2310	0.7429	0.6494
Decision Tree	0.3125	0.6977	0.6376
Random Forest	0.3846	0.8070	0.6444
Linear SVC	0.3077	0.7925	0.5593
RBF SVC	0.3077	0.8214	0.6556

Performance by Class (F1-SCORE)

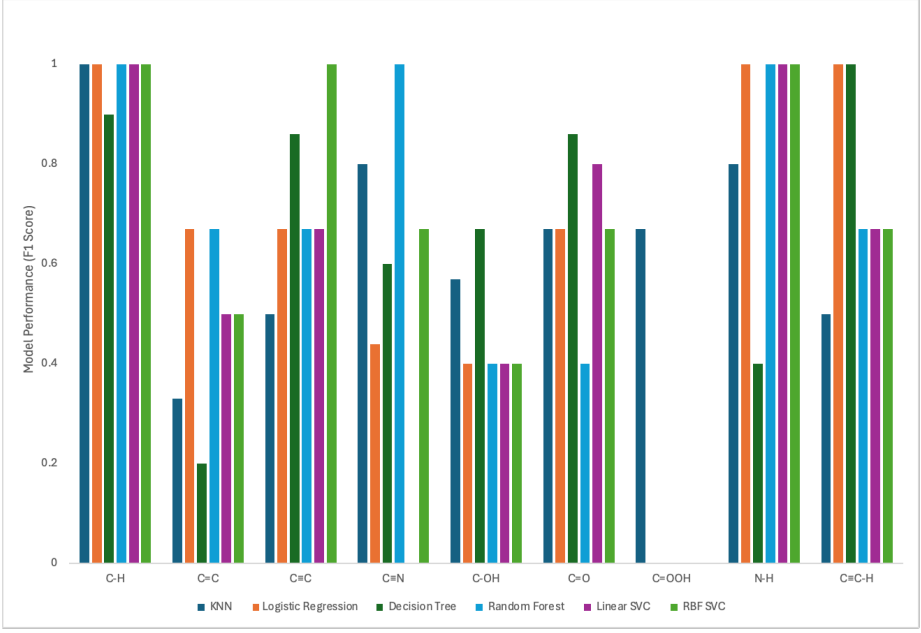

Figure 9: F1 Scores on Validation Data by Class

Model	C-H	C=C	C≡C	C≡N	C-OH	C=O	C=OOH	N-H	C≡C-H
KNN	1.0	0.33	0.5	0.8	0.57	0.67	0.67	0.8	0.5
Logistic Regression	1.0	0.67	0.67	0.44	0.40	0.67	0.0	1.0	1.0
Decision Tree	0.9	0.20	0.86	0.6	0.67	0.86	0.0	0.4	1.0
Random Forest	1.0	0.67	0.67	1.0	0.40	0.40	0.0	1.0	0.67
Linear SVC	1.0	0.50	0.67	0.0	0.40	0.80	0.0	1.0	0.67
RBF SVC	1.0	0.50	1.0	0.67	0.40	0.67	0.0	1.0	0.67

Interpretation

For average evaluation metrics the top three models were KNN, Random Forest, and RBF SVC. There was no one model that outperformed the others across all classes and some classes like C-H with a distinct feature at 3000 wavenumbers and high representation were far easier to predict than others like C=OOH.

I have also implemented some graphical interpretation tools that show the experimental spectra along side the key features in the spectra that led to the identification of a specific peak for methods were this is approach is applicable (Logistic Regression, Random Forest, Linear SVC). For other models the model simply returns a predicted list of functional groups for the spectra of interest.

Let's look at an example compound in the validation set: acetamide

Acetamide has the C=O, N-H, and C-H Functional Groups which have characteristic features at 1600 cm^{-1},3000 cm^{-1}, and 1600 cm^{-1} respectively lets see how the different models perform here using the graphical interpretation tool.

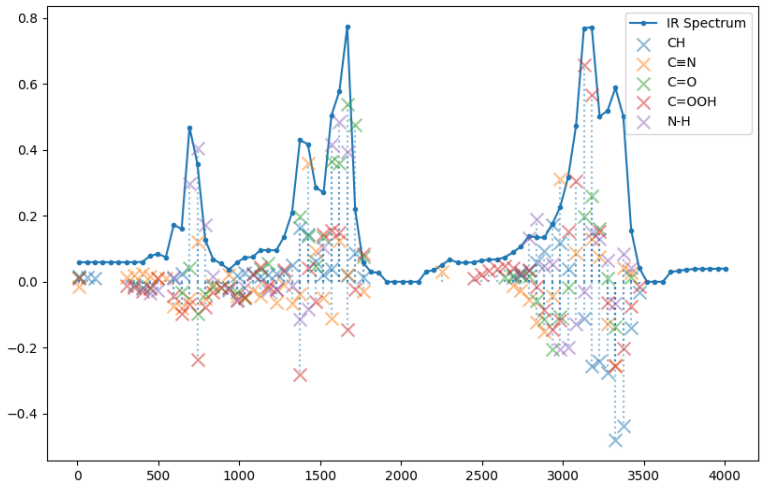

Figure 10: Logistic Regression Spectral Interpretation

The logreg model accurately finds the three correct functional groups but mistakenly also suggests C≡C and C=OOH, slight over fitting here as small features are being latched onto.

Linear SVC

The linear SVC model accurately finds the three correct functional groups based on the major features.

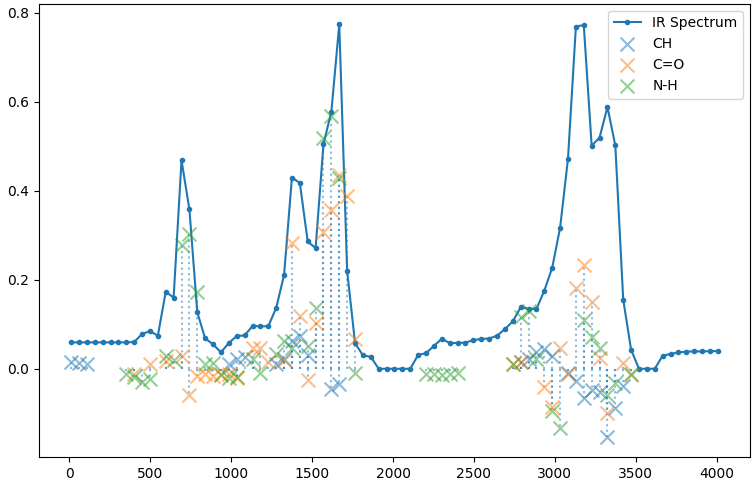

Figure 11: Linear SVC Spectral Interpretation

Random Forest

The random forest model accurately predicted 2 out of 3 functional groups but failed to identify the C=O stretch at 1600 cm^{-1}.

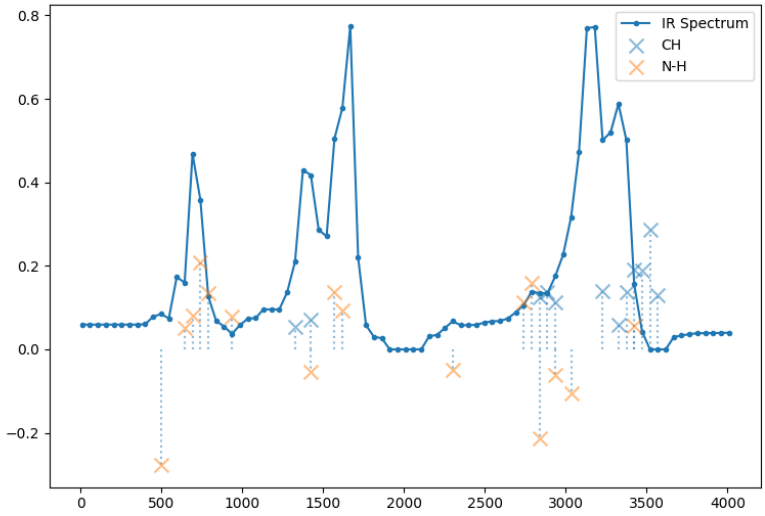

Figure 12: Random Forest Spectral Interpretation