Lamiya Zaveri
CS 584: Theory and Applications of Data Mining

Final Project: Report

Description

Motivation

Precision medicine is an emerging approach for disease treatment and prevention that focuses on individual health factors such as genetic variability, environmental differences, and lifestyle factors. Rather than providing a single approach for a given disease that is designed for the average person, this approach attempts to provide a personalized set of treatments and preventative measures for each patient based on their unique attributes. For cancer treatment specifically, a malignant tumor can have many genetic mutations, but the challenge is to distinguish drivers, those that contribute to tumor growth, from passengers, those that are neutral.

Data

Source

This dataset is available as a Kaggle Competition at https://www.kaggle.com/c/msk-redefining-cancer-treatment.

Data Description

The dataset is provided by the Memorial Sloan Kettering Cancer Center (MSKCC) and the training dataset is split into two files. The first training file *training_variants* contains information about the genetic mutations, and the second training file *training_text* links each mutation in the first file to clinical evidence provided in a text format that was used to manually classify the mutation.

Fields in training_variants

- **ID**: ID used to match the mutation information in the first file to the clinical evidence in the second file
- **Gene**: Gene where the genetic mutation is located
- **Variation**: Amino acid change caused by the mutation
- **Class**: Class number to which the mutation has been assigned (integer value from 1 to 9); class identifiers are not included in the test files

Fields in training_text

- **ID**: ID used to match the mutation information in the first file to the clinical evidence in the second file
- **Text**: Clinical evidence used to classify the mutation

Purpose

The purpose of this project is to develop a machine learning model to classify a set of genetic mutations based on their role in contributing to tumor growth.

Methods

Approach

Using the annotated training dataset in which researchers and oncologists have manually annotated mutations and classified them among one of nine classes, I develop a model that automatically classifies genetic mutations and I use it to classify mutations in a subset of the dataset reserved for testing. I use the following general steps to develop and evaluate this multi-class classification model:

1. Read in the data
    a. Read in both the *training_variants* and *training_text* data files and merge them to create a complete set of attributes for each mutation, matching each set of attributes on the ID field.
    b. Obtain descriptive statistics for the training data and plot the distribution of class identifiers to determine whether or not the data is balanced or unbalanced.
2. Feature selection and dimensionality reduction
    a. Convert the text data into a Bag of Words (BOW) representation, in which each unique word used in the set of text Strings is treated as a feature. Use the CountVectorizer function to split each String into words, convert them into all lowercase letters, and strip them of any leading or trailing punctuation characters.
    b. Within this function, use the stop_words parameter to remove words commonly used in the English language, leaving only words not contained in this list of "stop words."
    c. The CountVectorizer returns a *n* x *m* matrix of frequency values, where *n* = number of mutations and *m* = number of features, for the number of times each word in the BOW was present in the clinical evidence.
    d. Use the TfidfTransformer function to scale the values based on the importance of each word. This method, called Term-Frequency Inverse Document-Frequency (TFIDF), multiplies term-frequency, the number of times a word appears in a document, by the inverse-document frequency (idf) function, in which the importance of a word decreases with the frequency of its occurrence in a set of documents. The idf function used by TfidfTransformer is as follows, where *t* = term, n_d = total number of documents, and df(*d*, *t*) = number of documents containing term *t*:
    $$idf(t) = log\frac{n_d}{df(d,t)} + 1$$
    e. Use the TruncatedSVD function to reduce the dimensionality of the data.
    f. Use the LabelEncoder function to hot encode the information in the Gene and Variation fields, and then merge these fields with the BOW fields obtained from the text data.
3. Classifier Evaluation for Unbalanced Data

a. Evaluate a set of three classification models (K-nearest neighbors - KNN, Support Vector Classification - SVC, and a pruned Decision Tree classifier) using the unbalanced dataset.
    i. Split the training data into training and testing sets, with 80% of the data being used for training and the remaining 20% for testing.
    ii. Perform a grid search with 10-fold cross-validation to tune the relevant parameters for each classifier.
    iii. Using the trained model and the set of parameters that produced the best results from cross-validation, make a prediction for the 20% of the data that was retained from the test set. Evaluate the classification model using the F-1 measure and confusion matrix for the test set.
b. The relevant parameters for the three classification models are as follows:
    i. **KNN**:
        1. **N_neighbors (k)**: The number of neighbors to use when determining the class assignment for a particular mutation.
    ii. **SVC**:
        1. **C**: The penalty parameter for the error term.
        2. **Gamma**: The kernel coefficient for a radial basis kernel function.
    iii. **Pruned Decision Tree**:
        1. **Max_depth**: The maximum depth or number of nodes traversed between the root of the tree and its leaves.
        2. **Min_samples_split**: The minimum number of samples required at an internal node in order to split the node further.

4. Balancing the data
    a. Use the model that produced the highest F-1 score to evaluate three methods of balancing the data, undersampling, oversampling, and a hybrid technique called SMOTE. SMOTE keeps all of the data points in the minority sample, and then interpolates new data points in order to reach the same number of records as in the majority sample. This technique works because of the smoothness principle, in which points within a neighborhood tend to belong to the same class.
5. Reevaluation of classifiers using the balanced dataset
    a. Use the method of balancing data that produced the highest F-1 score to reevaluate the three methods of balancing the data. The model that produces the highest F-1 score in this case is the best model.

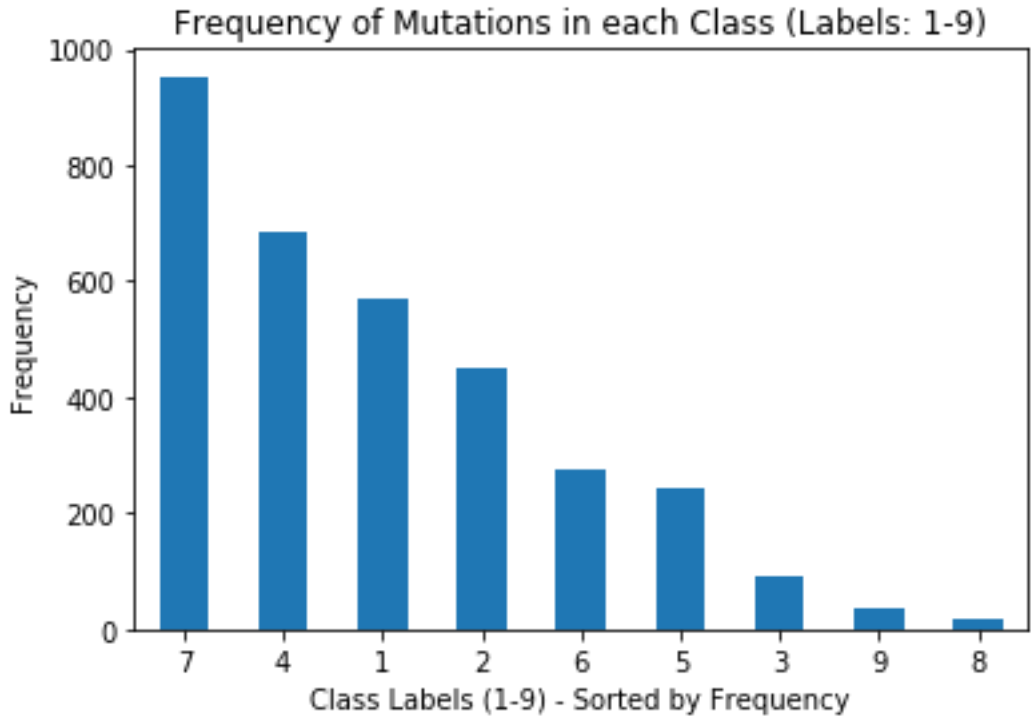
Additional Complexity

This project goes above and beyond the assignments we have done in class because it is a multi-class classification problem and because I use both the attributes provided in the first training file about each mutation as well as the text-based clinical evidence, parsing each dataset separately. For the text-based evidence, I use the text classification techniques we learned in class, such as creating a Bag of Words representation and using TFIDF, stop words, and SVD for dimensionality reduction.

Results

Descriptive Statistics

Number of mutations: 3,321



Number of unique words in BOW representation: 281,467

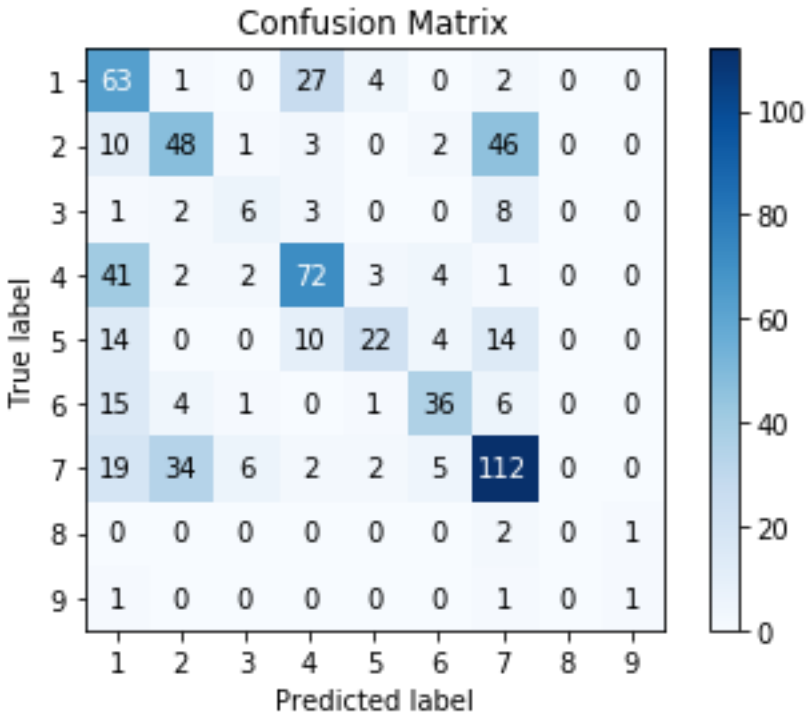
Number of components retained after SVD & combining the first and second files: 106

Classifier Evaluation for Unbalanced Data

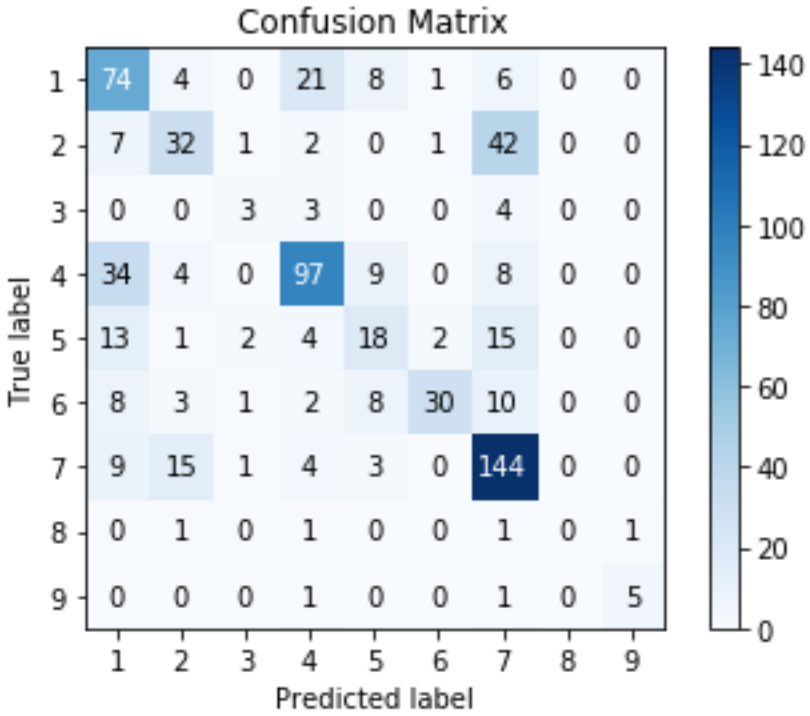
KNN:

K: 3

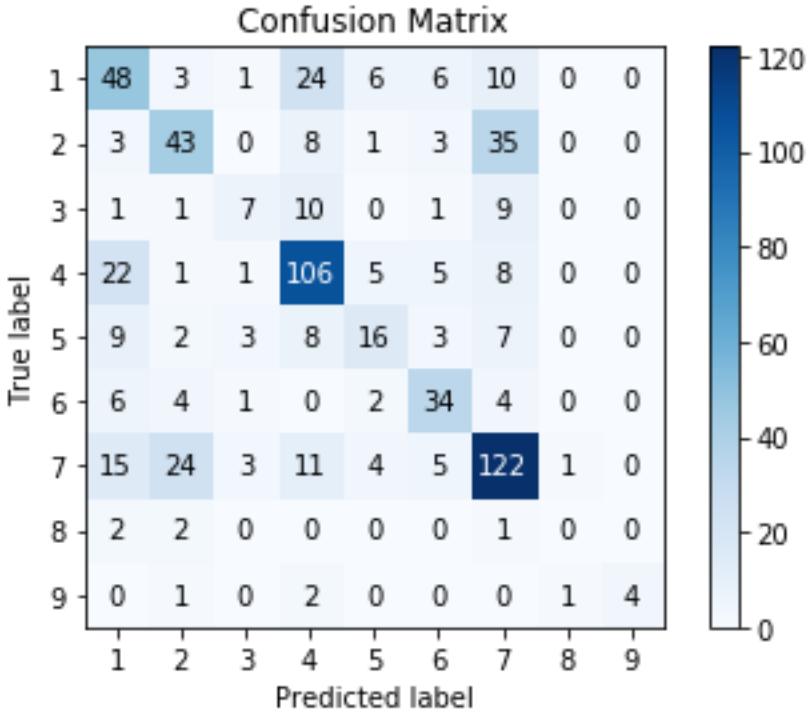
F-1 score: 0.540



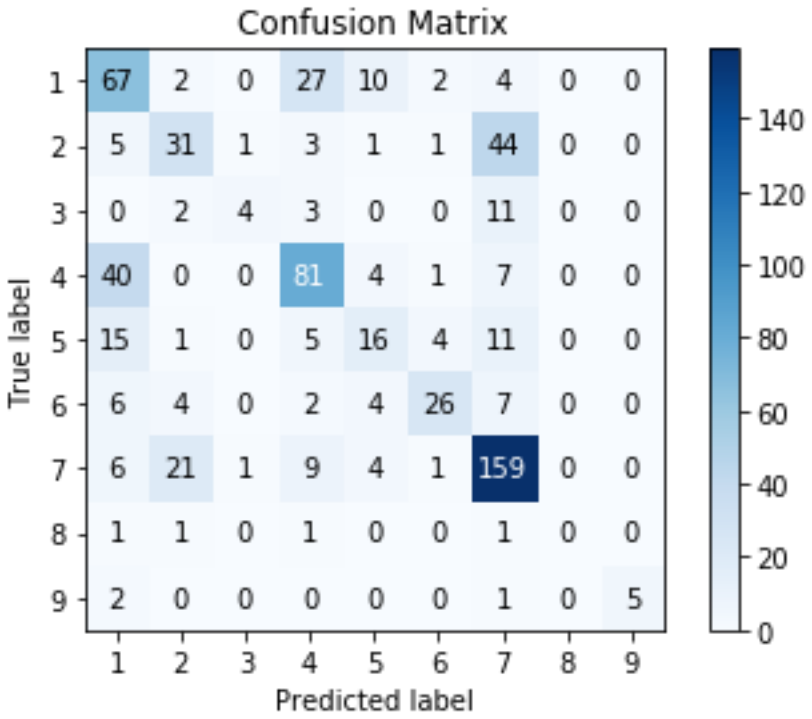
SVC:
C: 10
Gamma: 1
F-1 score: 0.597



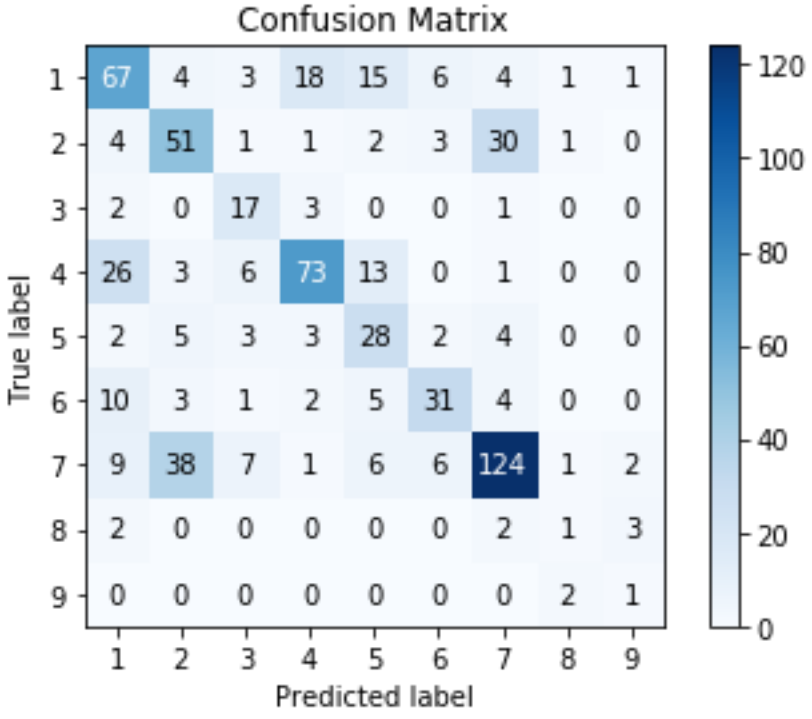
Pruned Decision Tree:
Max depth: 160
Minimum samples split: 8
F-1 score: 0.564



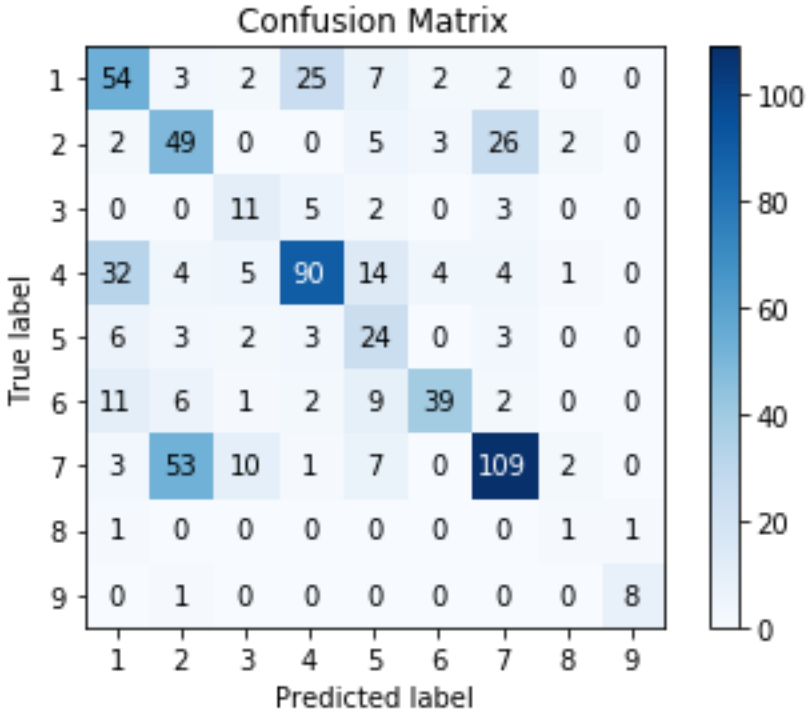
Evaluation of Balancing Methods
Undersampling:
F-1 score: 0.573



Oversampling:
F-1 score: 0.591

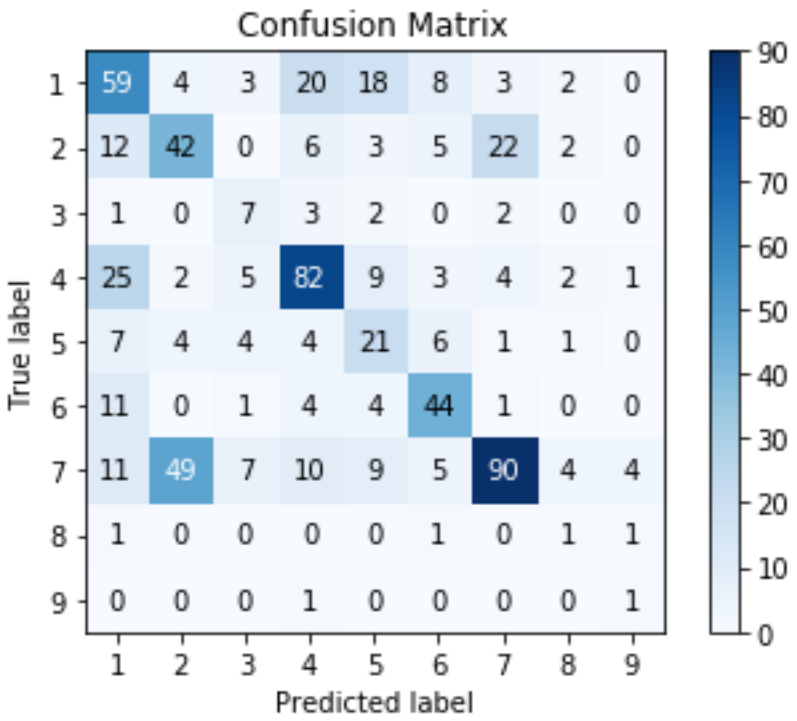


SMOTE:
F-1 score: 0.597

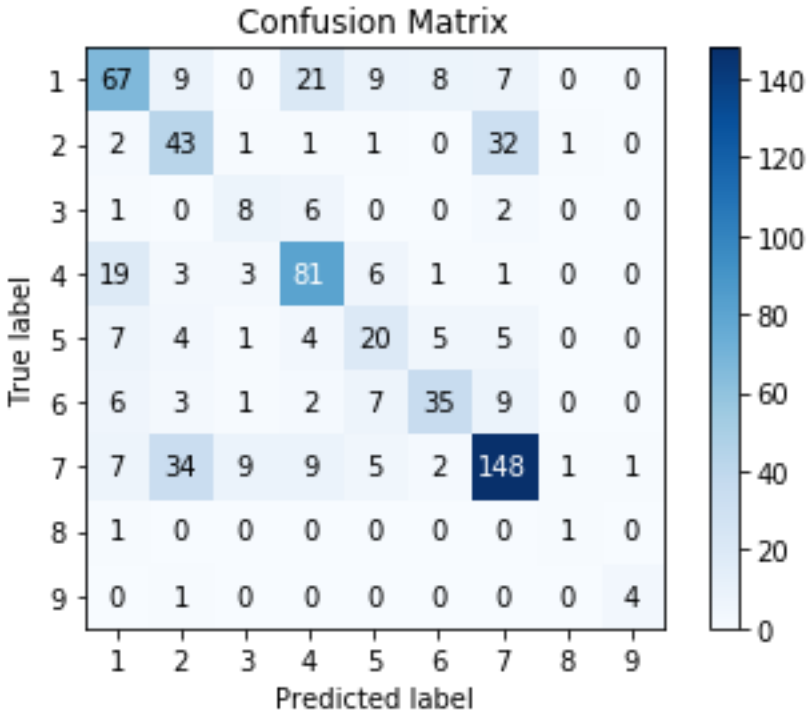


Classifier Evaluation for Balanced Data

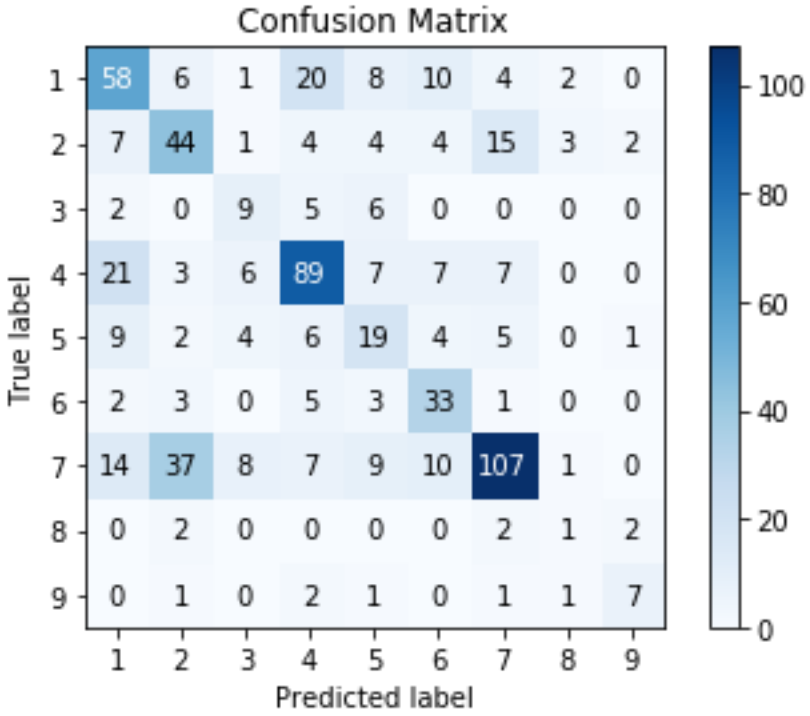
KNN:
K: 3
F-1 score: 0.532



SVC:
C: 10
Gamma: 10
F-1 score: 0.615



Pruned Decision Tree:
Max depth: 160
Minimum samples split: 8
F-1 score: 0.558



Conclusions

After plotting the frequency distributions of mutations in the training datasets, we see that the data is highly unbalanced, with most of the mutations being assigned to Class 7 and very few being assigned to Class 8. As a result, we implement techniques to balance the data and use the F-1 metric in order to develop a more accurate classification model and evaluate its accuracy appropriately.

First, we evaluate the three chosen classifiers: KNN, SVC, and a pruned decision tree, using the unbalanced data. From the results above, we can see that SVC with a radial basis function kernel produced the highest F-1 score of 0.597 with parameters C = 10 and Gamma = 1. Consequently, we use this model to evaluate the three balancing methods: undersampling, oversampling, and SMOTE. From these results, we can see that SMOTE produced the highest F-1 score of 0.597. As a result, we reevaluate the same three classifiers, using SMOTE to balance the subset of the data for training the model. The results show that SVC produces the highest F-1 score again of 0.615.

Limitations and Future Work

One limitation of the dataset is that it includes the same text evidence for multiple mutations since a single peer-reviewed journal article can include information for multiple mutations. This convolutes the dataset because each mutation does not contain distinct attributes. In addition,

the dataset is biased because the mutations are manually classified by researchers who parse information from the literature, and different individuals may classify a mutation differently based on their personal interpretation of clinical evidence. Thus, if a model is trained on a dataset annotated by one researcher and tested on a dataset annotated by another, the model may not perform as accurately, because it is biased toward a factor that is not included as an attribute in the training data.

The model can be improved by using other classification methods such as deep learning. We can also combine multiple classifiers via ensemble methods such as bagging and/or boosting.