**Pilot Experiments on Feature Importance  
Using the Rutgers COVID 1,058-Subject Pilot Database**

We have conducted some pilot experiments designed to help us understand the Rutgers COVID data. Though there were a large number of features in the original data, many of the cells were missing values. Before we tackle the problem of missing data, we decided to select an 880-subject subset in which all subjects had valid data for 16 representative features. After removing two incorrectly labeled subjects, this became an 878-subject subset. These features are shown in Table 1:

|  |  |  |  |
| --- | --- | --- | --- |
| (1) Admission Status | (2) Emergency Room Patient | (3) Age | (4) Sex |
| (5) Race | (6) Ethnicity | (7) ICU days obs from ICU files | (8) Vizient Service Line |
| (9) Vizient Sub-Service Line | (10) ICD10\_Diagnosis1 | (11) ICD10\_Diagnosis2 | (12) ICD10\_ Diagnosis3 |
| (13) ICD10\_Diagnosis4 | (14) ICD10\_Diagnosis5 | (15) ICD10\_Diagnosis6 | (16) ICD10\_Diagnosis7 |

Table 1. A subset of features was selected that had no missing data

We assigned an arbitrary number as a label to each class based on the following list:

* Class 0: Neuro = No, COVID = No

|  |  |  |  |
| --- | --- | --- | --- |
| **Class** | **Train** | **Evaluation** | **Total** |
| 0 | 298 (45%) | 98 (45%) | 396 (45%) |
| 1 | 106 (16%) | 35 (16%) | 141 (16%) |
| 2 | 193 (29%) | 64 (29%) | 257 (29%) |
| 3 | 63 (10%) | 21 (10%) | 84 (10%) |
| TOTAL | 660 (75%) | 218 (25%) | 878 (100%) |

Table 2. A histogram of the pilot database is shown. A blind evaluation data set was created by partitioning the data using a 75% (train) / 25% (eval) split.

* Class 1: Neuro = No, COVID = Yes
* Class 2: Neuro = Yes, COVID = No
* Class 3: Neuro = Yes, COVID = Yes

We then sorted the data by class and produced the summary statistics shown in Table 2. The data was split into two sets: /train and /eval by assigning 75% to training set and 25% to the evaluation set. The data is considered unbalanced since there is a significant variation in the frequency of occurrence for each class – class 0 accounts for 45% of the total data. This necessitates doing some form of boosting to avoid the usual pitfalls during classifier training (which we will address when we have access to the entire data set). In building the evaluation set, we attempted to preserve the unbalanced distribution of the classes, so that the evaluation of algorithms using prior probabilities would be fair.

As can be seen by the data in Table 2, a simple baseline for performance can be achieved by always guessing the most likely class – class 0 in this case. This approach, often called the Zero Rule, would achieve an error rate of 55%. This is simply another example of the unbalance of the data. A valid question to ask in a baseline study is to what extent can an algorithm improve performance compared to the Zero Rule since this is often surprisingly difficult for unbalanced data sets.

Since most of our dataset is categorical, and the machine learning algorithm needs numerical data to do prediction, we used a hybrid encoding method based on frequency encoding to turn categorical data into numerical values. This is often called feature engineering. We rank-ordered simple categorical variables such as sex and ethnicity, by frequency of occurrence and then assigned integer indices to these values. For the diagnosis features, we sorted them in lexical order because the terms shared common prefixes (e.g., a diagnose mnemonic or a phrase like “neurology”) and assigned integers accordingly. There are a variety of methods used today for converting categorical variables such as on-hot encoding or label encoding. Good machine learning algorithms are usually somewhat insensitive to the nuances of these techniques. Pilot experiments on our data showed the encoding system did not make a significant difference and our approach seems to work as well as any.

We then used the statistical package JMP to generate a baseline analysis of our features. Using the Bootstrap Forest algorithm in JMP, we trained a model on the training data set described in Table 2, and then we used the feature importance analytic tool to determine how strongly each feature contributes to the overall prediction of a label in the evaluation set. The results of this analysis are shown in Figure1. ICD10\_Diagnosis1 is by far the most powerful feature.

To get a sense of what type of performance can be achieved on this data, we evaluated three standard algorithms that are known to give good performance across a wide range of applications:

* **PCA:** Principal Components Analysis (PCA), often referred to as discriminant analysis, uses a classic Bayes Decision rule under the assumption that the data can be represented as multivariate Gaussian distributions. We prefer to refer to this approach as class-dependent Principal Components Analysis with a maximum a posteriori decoding rule (PCA-CD-MAP).
* **SVM:** A Support Vector Machine (SVM) is a discriminative learning approach in which we directly model the decision surface between two classes using a mathematical model based on the actual data points near this hypothetical boundary. In practice, this technique, when combined with a nonlinear kernel function, performs well for data consisting of continuous-valued features.
* **RF:** Random Forests (RF) are a decision tree-based approach that is widely regarded as one of the most powerful machine learning algorithms for fixed-length data. This algorithm can achieve error rates close to zero on the training set but is prone to overtraining. Hence, controlling the complexity of the tree during training to maximize the generalization properties of the model is critical.

It is instructive to first look at the performance of RF as a function of the features to better understand which features are most important and to verify the feature importance analysis. In Figure 2, we show a comparison of performance for RF, in terms of error rate, using the features ordered as shown in Figure 1. We evaluate on both the training set (closed-set testing) and the evaluation set (open-set testing). The difference in performance of these two is often an indication of the limitations of the data and overtraining. We see that the first three features reach an error rate close to 20%. The remaining features don’t add much except perhaps the last feature (Emergency Room Patient).

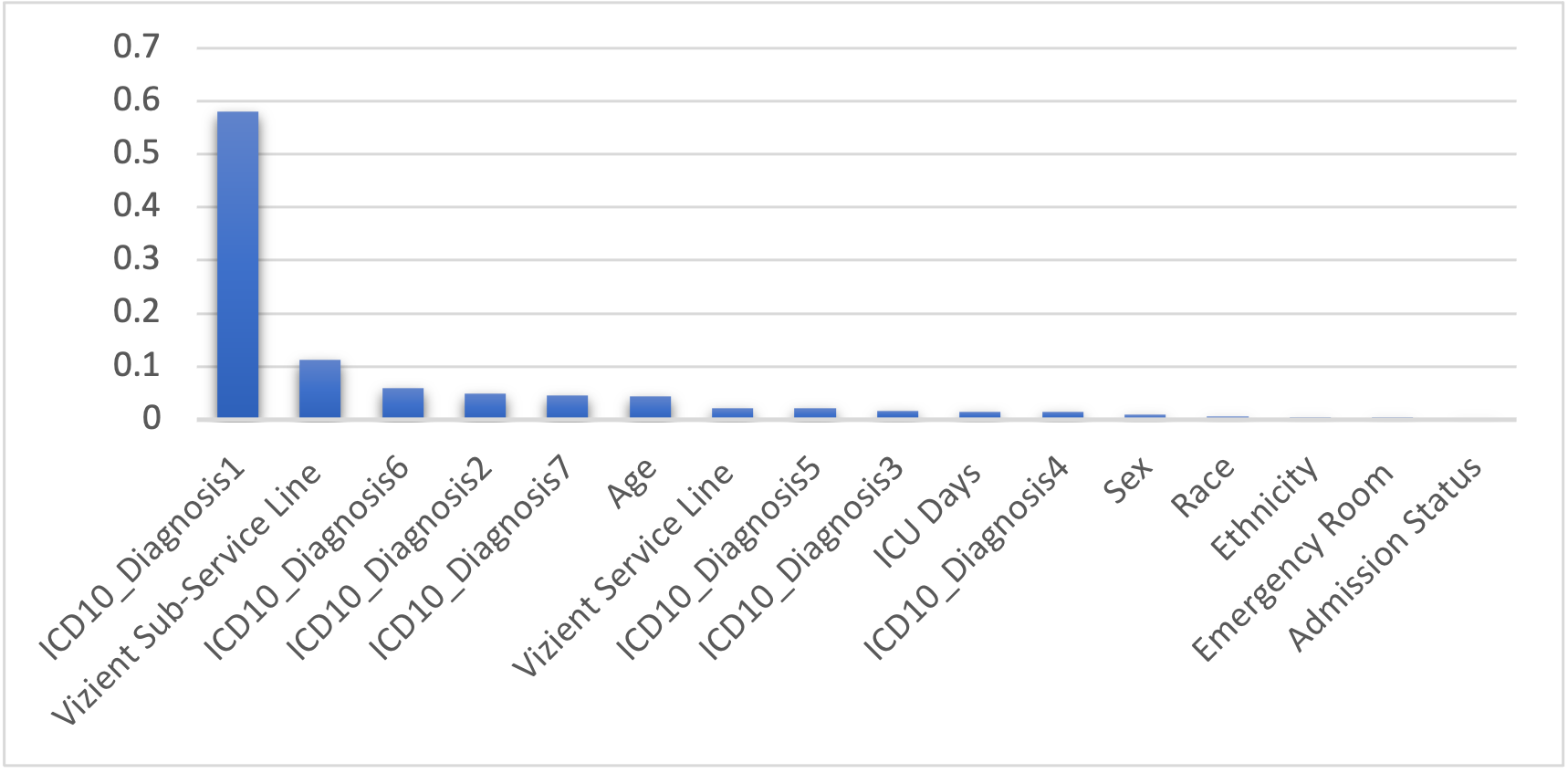


Figure 1. Feature importance for the 16 selected features using the Bootstrap Forest algorithm in JMP.

On the training data, we achieve an error rate close to zero, which is expected for the RF algorithm. We adjusted the parameters of this algorithm to make sure it wasn’t overtraining, but performance on the blind evaluation set didn’t vary significantly.

Chart, line chart

Description automatically generated

Figure 2. An analysis of RF as a function of the features ordered by importance

On the evaluation data in Figure 2, the error rate hovers around 20%, which most likely simply a reflection of the limitations of the data. The fact that performance get worse on the training data for the second feature, while it improves slightly on the evaluation data, indicates some degree of mismatch in the two data sets. Again, this is most likely simply an indication that the amount of data is too small to draw any definitive conclusions.

As we mentioned, RF is a very reliable benchmark. In fact, performance of modern deep learning systems will not greatly exceed this level of performance on this type of task. Nevertheless, we evaluated the three previously mentioned algorithms, as shown in Figure 3, to get an idea of the range of performance that was achievable on this data. Error rates are plotted as a function of the importance of the feature as shown in Figure 1. The RF algorithm, due to its ability to integrate numerical and categorical data, performs better than both PCA and SVM.

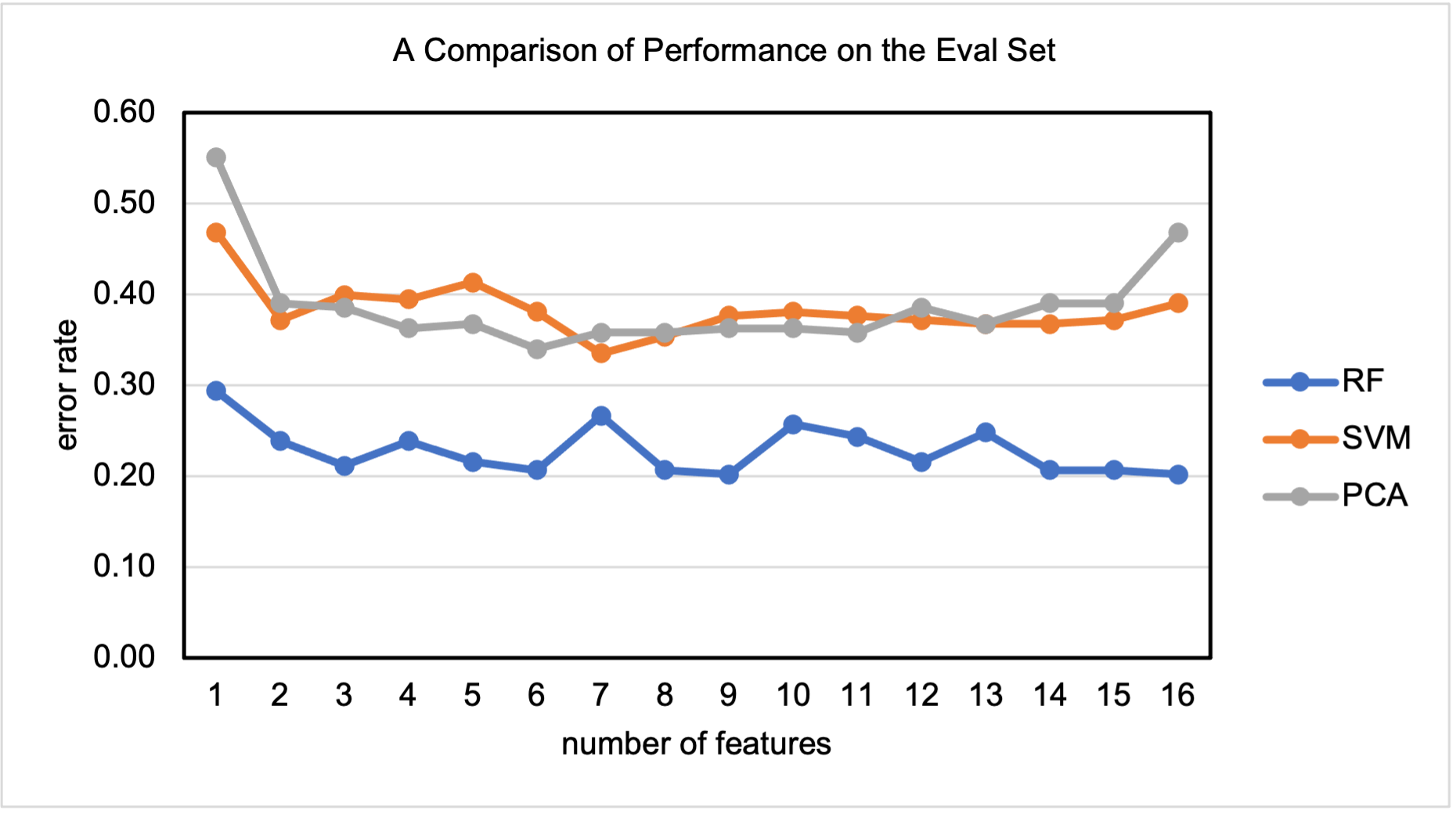


Figure 3. A comparison of three standard classification algorithms on the COVID pilot data set is shown. Note that the features are ordered by importance as shown in Figure 1.

We also examined the confusion matrix to analyze the types of errors being made. This analysis is shown in Table 3. The dominant error modality is class 3 predicted as class 1. Class 3 in general is poorly predicted while class 1 is frequently causing confusions. This is probably simply another indication that the data set is too small since these classes are disproportionately low in the training data set.

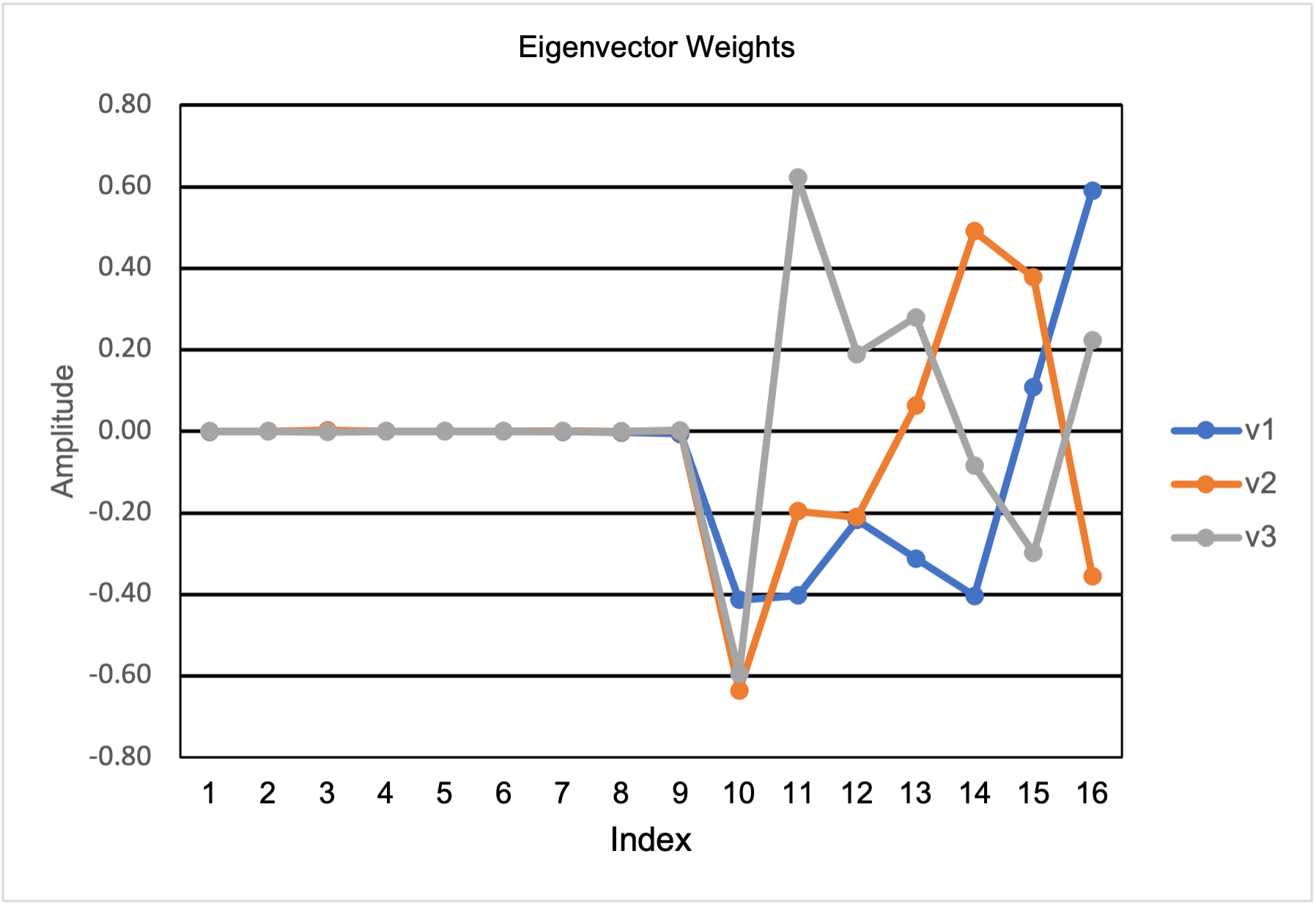


Figure 4. A plot of the three most significant eigenvectors that shows the first 9 features are plot playing a significant role in the prediction.

We can also look at the weights of the three eigenvectors associated with the largest eigenvalues to gain some insight into what these models have learned. A plot of the first three eigenvectors is shown in Figure 4. From this plot it is clear that the first 9 features do not play a significant role in classification because the weights associated with those features are close to zero.

**Recommendations:**

This report represents our first set of baseline experiments on the Rutgers COVID data. These experiments were designed to help us understand the data and remove any bugs in our experimental set up. Our baseline performance is well above what we can achieve with the Zero Rule or random guessing, so we feel the algorithms are able to successfully learn from the data.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Class** | **0** | **1** | **2** | **3** |
| **0** | 90 | 3 | 5 | 0 |
| **1** | 4 | 29 | 0 | 2 |
| **2** | 12 | 0 | 52 | 0 |
| **3** | 5 | 13 | 0 | 3 |

Table 3. A confusion matrix for the RF algorithm on the evaluation data set is shown. The dominant error modality is class 3 misrecognized as class 1.

We are now at a point where we can do some meaningful experiments on the data. There are four main issues to be addressed:

1. The unbalanced data can lead to overly optimistic results.
2. We need to identify inputs and outputs – what label are we trying to predict? From what inputs?
3. A large portion of the data have incomplete entries. To deal with the full data set, we will need to decide what we should do about the missing data.
4. Some of the features seem redundant or at least are not useful in predicting outcomes.

If we examine performance in Figure 3 carefully, we observe a few interesting issues. Let’s focus on the RF performance. The 7th feature increases the error rate. This corresponds to “Vizient Service Line.” Similarly, the 10th feature, which corresponds to “ICU Days”, increases the error rate, as does the 12th feature, which is “Sex.” The small fluctuations in error we see going from feature 5 to 6 and 8 to 9 and 15 to 16 might indicate some issues with the data, such as data entry errors or a lack of understanding of the meaning of some of these terms. Optimal performance on the eval set can probably be achieved by combining features 1 (ICD10\_Diagnosis1), 2 (Vizient Subservice Line), 3 (ICD10\_Diagnosis6), 6 (Age), 9 (ICD10\_Diagnosis3) and 16 (Admission Status). But this kind of tweaking will be better considered when we have the complete database.

Therefore, the team needs to meet to discuss these results and the implications of them on our proposed project. As with all projects of this nature, understanding the data is an important step towards the successful application of machine learning algorithms.