Predicting the Severity of Adverse Drug Reactions

Katherine Erdman
kerdman@stanford.edu

Neal Patel
npatel21@stanford.edu

1. Introduction

Polypharmacy, co-prescribing multiple drugs, is incredibly common and often leads to drug interactions that can have adverse side effects. Currently, to aid doctors in prescribing treatments, clinical decision systems fire alerts when drug combinations are prescribed that have known reactions (Reis & Cassiani, 2010). Those alerts are based on drug interaction severity stored in databases such as Lexi-Interact. However, Lexi-Interact severity, which is based on clinical trials and literature reviews, does not include all drug interactions though there are many prescribed drug combinations that haven't been formerly covered by literature (Smithburger, et al., 2010). As an example, within a general oncology unit in Belgium, 60% of patients were taking at least one medication pair not within Lexi-Interact (Debruyne, et al., 2012). While a significant portion of drug combinations are not within Lexi-Interact, adverse side effects of such combinations have been reported to the FDA's Adverse Event Reporting System. We propose to use multinomial classification methods, SVM, Naive Bayes, Logistic Regression and Random Forests, to predict drug-drug interaction severity values from the adverse drug reactions in the FDA's database. If successful, this implies that severity values, and thus alerts, can be generated for drug combinations not formally studied.

2. Related Work

To address the widespread problem of unpredicted, adverse drug reactions, *in silico* methods of predicting if there will be an adverse effect (Chen, et al., 2016), as well as if a specific adverse effect will occur, have been created (Zitnik, et al., 2018). Specifically, Decagon, a convolutional network algorithm for link prediction, models protein-protein, drug-protein and drug-drug interactions (DDIs) and is state-of-the-art when predicting specific side effects (Zitnik, et al., 2018). However, Decagon is limited to drugs that are studied enough to have comprehensive interaction information. Unlike Decagon, past work indicates that the FDA Adverse Event Reporting System can be solely used to predict new drug interactions (Kuhn, et al., 2010). Given drug pairs and their DDI, graphical measures of similarity between two drugs can predict unseen side effects, as similar drugs tend to have similar drug interactions (Tatonetti, et al., 2012). However, while a list of potential side effects can be generated and is informative, the overall severity of an interaction, which has yet to be predicted *in silico*, would be more useful, as severity, such as that from the Lexi-Interact database, is currently integrated in clinical workflows.

Like other biological datasets, the FDA's Adverse Event Reporting System has a large class imbalance as few drug pairs with mild or extreme side effects are reported because minor events go unreported and life-threatening events are typically not repeated. Previous work to address such imbalance explores downsampling from the majority classes or upsampling the minority classes, such that the training set is balanced and the test set is untouched (Bone, et al., 2015). This upsampling approach duplicates full training examples rather than generating simulated data points. By doing so, it removes a bias towards the majority classes, while making no assumptions about the underlying data. Conversely, synthetic examples of the minority class can be created, and can be especially useful if they are near the decision boundary (Han, et al., 2005). However, by generating unseen data by averaging the five nearest neighbors of a data point, there is the possibility to create a feature vector that is unrealistic, especially if features are dependent.

3. Data

The presence or absence of 1,317 potential adverse drug reactions will be a sparse vector that represents each of the ~63,000 drug-drug interactions recorded in the FDA's Adverse Event Reporting System. Those reports have been parsed and stored in the TWOSIDES database (Tatonetti, et al., 2012). The true labels for these interactions are one of 5 classes from Lexi-Interact: Avoid combination, Consider Therapy Modification, Monitor Therapy, No Action Needed, and No Known Interaction (Up-to-date). As Lexi-Interact is a proprietary web-app, made available via Stanford's license, that does not have an API, in order to determine true labels, groups of 50 drugs were manually entered and resulting severity interaction scores then scraped. The intersection of drug-drug interaction records from TWOSIDES with the drug-drug severity scores from Lexi-Interact resulted in 3,646 drug pairs.

3.1 Class Imbalance

The majority class, a severity level of "Monitor Therapy," accounts for 71% of the data. This makes sense as, due to liability reasons, Lexi-Interact has incentives to be more cautious when assigning severity scores like "No Action Needed" or "No Known Interaction" and doctors have incentives to not prescribe drug-combinations labeled "Avoid combination" or "Consider Therapy Modification." As a first step to minimize class imbalance, Lexi-Interact severity scores were grouped by clinical relevance. Severities that required similar action by physicians were combined to eliminate some minority classes.

Table 1: Frequency of classes before and after combination

Lexi-Interact Label	Frequency	Combined Label	Frequency
No Known Interaction	7	No Action	225
No Action Needed	218		
Consider Therapy Modification	2604	Consider modification	2604
Monitor Therapy	645	Action required	817
Avoid Combination	172		

Further, upsampling was utilized to remove class imbalance within the training set, rather than downsampling or class weighting, as it resulted in better accuracy. After splitting the data as appropriate, full feature vectors from the minority classes were randomly chosen with replacement to be repeated within the training set until the size of all three classes was equivalent.

4. Multinomial Classification Models

Define y as an integer classification that corresponds to a severity and x is a vector of features.

4.1 Naive Bayes

The naive Bayes model applies Bayes' theorem and the assumption of independence between all of the features for a specific instance to states that the estimated classification, $\hat{y}$ is:

$$\hat{y} = \arg\max_{y} P(y) \prod_{i=1}^{n} P(x_i \mid y),$$

4.2 Logistic Regression

The logistic regression model attempts to approximate $P(y|x)$. The difference values of theta are determined by the training data using stochastic average gradient optimization to maximize the function:

$$LL(\theta) = \sum_{i=1}^{n} y^{(i)} \log(\sigma(\theta^T \cdot x^{(i)})) + (1 - y^{(i)}) \log[1 - \sigma(\theta^T \cdot x^{(i)})]$$

where θ is the vector of trained parameters, $x^{(i)}$ is the ith example, and $y^{(i)}$ is the corresponding label (as a one-hot vector).

4.3 SVM

The multi-class SVM model uses a "one-against-one" approach for classifying. If there are n potential classes, it trains $n \cdot (n-1)/2$ classifiers as each classifier trains data from two distinct classes. Each SVM classifier, parameterized by w and b, makes a classification based on the output of g($w^T x^{(i)} + b$) and has a functional margin of $y^{(i)}(w^T x^{(i)} + b)$ for each example i. Ultimately, SVM works to optimize:

$$min_{w,b} \frac{1}{2} ||w||^2 \; given \; y^{(i)}(w^T x^{(i)} + b) \geq 1 \text{ for all i}$$

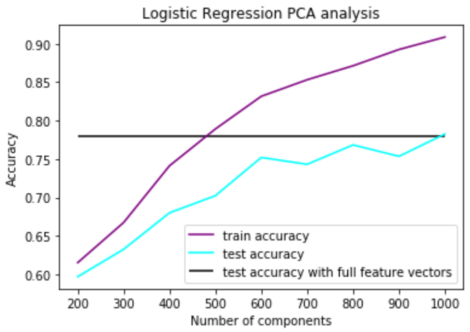
4.4 Random Forests

Random forests create numerous decision trees and make a classification by picking the majority class across regions. Gini loss is minimized where Gini loss is $\Sigma q_m \Sigma p_{mk}(1 - p_{mk})$ where p_{mk} is the proportion of class k in region m and q_m is the proportion of samples in region m.

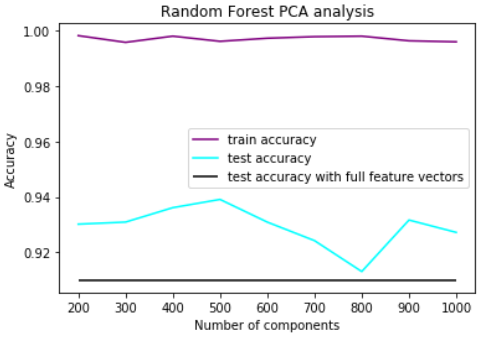
5. Features

Given that there are only 3,646 unique drug pairs, we wanted to explore reducing our 1,317 feature vector using PCA. Accuracy using 10-fold cross validation and an 80/20 train/test split was used to determine the optimal number of features for each model to use in future experiments. Most notably, PCA almost doubles accuracy for Naive Bayes, maybe because Naive Bayes is hindered by the fact that the original feature vectors are very sparse.

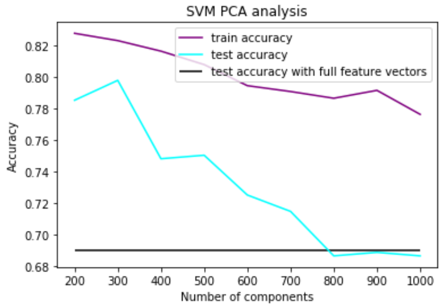
Table 2: Accuracy of multi-class classification algorithms vs. number of PCA components



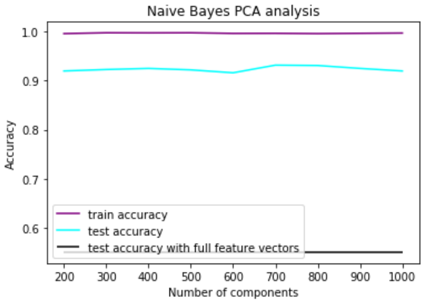
Optimal feature vector: 1000 PCA components



Optimal feature vector: 500 PCA components



Optimal feature vector: 300 PCA components



Optimal feature vector: 700 PCA components

6. Results/Discussion

For the following experiments, an 80/20 train/test split was used, unless tuning hyperparameters, in which case a 70/15/15 train/validation/test split was used. The average accuracy across 10-fold cross-validation is reported. When comparing the best performing models, precision, the ratio of true positives over all results, and recall, the number of labeled positives that are actually positive, are used in a discussion of model preference.

6.1 Baseline

As a baseline comparison, each drug-drug interaction was represented by five potential reactions which we thought would be indicative of severity: difficulty breathing, narcolepsy, diarrhea, AFIB, and emesis. The resulting logistic regression accuracy was 42%.

6.2 Logistic Regression

To tune logistic regression, different multi-class classification schemes were tested. Cross entropy loss had a higher accuracy than the one-vs-rest model, but seemed to suffer from overfitting, even with L2 regularization, as the train accuracy was over 10% greater than the test. Reducing the feature space though PCA helped with overfitting, the gap between train and test accuracy significantly decreased [Table 2], but when that occurred the accuracy is far below other classification methods. Logistic regression probably has poor performance because the decision boundary is not very linear, even in 1000-dimensional space.

Table 3: Accuracy for Logistic Regression w/ feature vectors of 1000 PCA components

Model	Train Accuracy	Validation Accuracy	Test Accuracy
One-vs-Rest (choose label with highest probability)	0.881	0.767	0.779
Cross Entropy Loss	0.910	0.803	0.797

6.3 Naive Bayes

Naive Bayes does not rely on any hyper-parameters and thus no tuning was necessary. The training accuracy was 0.997 and the test accuracy was 0.931. One interesting takeaway was how drastically PCA increased the performance of this algorithm. The number of PCA components has little effect, but the act of transforming the data through PCA alone increased the test accuracy from 0.55 to above 0.90. This is probably because the feature vectors were sparse and there were only 3,646 examples of length 1,317.

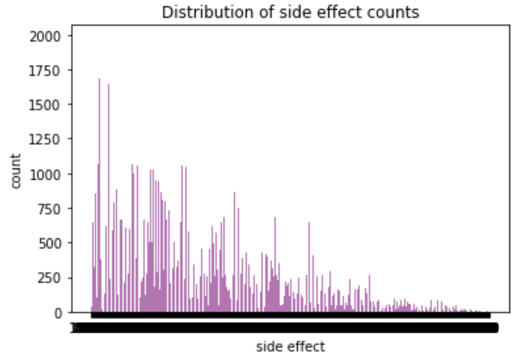


Figure 1: Distribution of side effect counts
For each of the 1,317 different side effects, the total number of examples that they appear in is defined as the count. There are many features with low counts, caused by feature vector sparsity, which when combined with the relatively low number of training examples, hinders Naive Bayes from learning true probabilities as there isn't enough information for a given class and feature pair. By transforming the data via PCA, the sparseness is lost and the data is also no longer binary. We hypothesize that this allows for better learning.

6.4 Random Forest

For the random forest model, a hyperparameter is the number of decision trees. By ranging the number of decision trees, the optimal validation accuracy was achieved using 20 decision trees. The train accuracy was 0.999, the validation accuracy was 0.950 and the test accuracy was 0.952. Train accuracy was higher than the test, but the difference isn't that large.

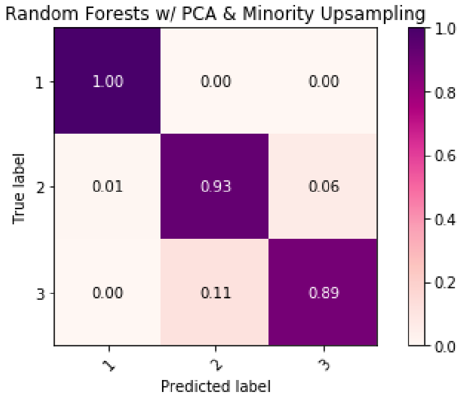


Figure 2: Normalized Confusion Matrix for optimized Random Forest Model
The numeric labels of 1, 2 and 3 correspond to the labels of "No Action", "Consider Modification," and "Action Required" respectively. The adjacent confusion matrix is for the Random Forest model trained on feature vectors transformed into the 800 PCA components that explain the most variance in the data. The model utilizes 20 decision trees within the random forest and represents the results of 729 test examples. Note that the precision and recall for labels 2 and 3 is less than 1. In fact, the recall for label 3 is just 0.88, implying that a significant number of serious drug complications are incorrectly labeled as *less* severe.

6.5 SVM

For the SVM model, initial experiments showed the the rbf kernel outperformed the linear, polynomial and sigmoidal. Thus, we tuned SVM by adjusting gamma, a measure of how far a single example's influence can spread, and C, a way of regularization by determining the relative weight of accuracy versus widening the decision boundary. With the optimal hyperparameters, the training accuracy was 0.999, the validation accuracy was 0.973 and the test accuracy was 0.965.

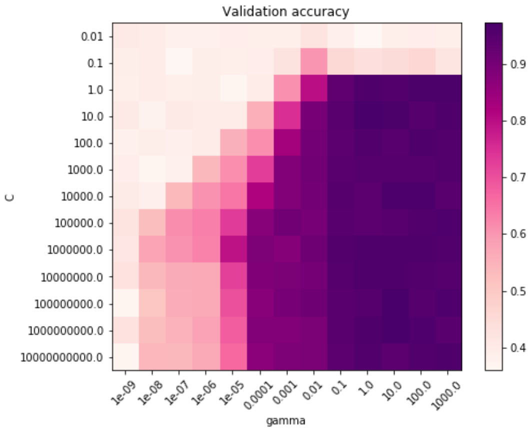


Figure 3: Validation accuracy for varying levels of C and gamma
Using 2-fold validation, the highest validation accuracy of .973 was obtained using a gamma value of 10 and a C value of 10^{10}. Given the incredibly large value of C, the margin is really small, which favors a more complex model and puts the model at higher risk of over-fitting. However, the PCA-transformed feature vector reduces the space from 1317 features to 300 features, which may counteract the over-fitting that can be introduced from the large C value. Note that given a gamma of 1.0, the range of accuracies between a C of 1.0 and 10^{10} is 0.0107. Visually, as long as gamma is greater than 0.1 and C is at least 1.0, the difference in accuracy is minimal.

As SVM has an accuracy that is comparable to that of Random Forest, a comparison of errors can be informative. SVM had the same issue with misclassification of some DDI's labeled "Action Required," though the recall for this class was higher by

0.1. SVM had marked improvements with correctly predicting label 2, reflected in the perfect recall for label 2 and perfect precision for label 3.

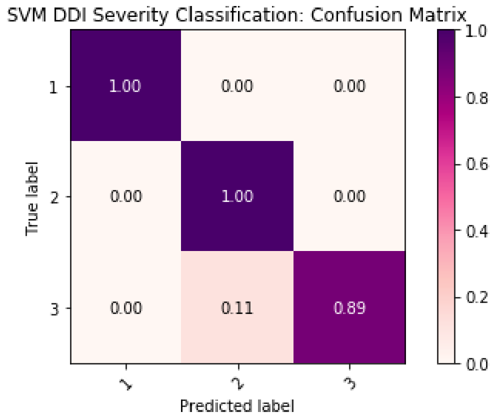


Figure 4: Normalized Confusion Matrix for optimized SVM Model
The numeric labels of 1, 2 and 3 correspond to the labels of "No Action", "Consider Modification," and "Action Required" respectively. The adjacent confusion matrix is for the SVM model trained on feature vectors transformed into the 300 PCA components that explain the most variance in the data. The value of C is 10 and the value of gamma is 10^{10}. The model has an 80/20 train/test split with the training set upsampled to remove class imbalance. As such, the confusion matrix represents the results of 729 test examples. The recall and precision for label 2 was 1 and 0.91 respectively, while that for label 3 was 0.89 and 1 respectively.

The areas of the confusion matrix that cause concern are when the predicted label is less than the actual. In these instances, the suggested severity would cause the physician to take more minor action than required, which can have life-threatening implications. In particular, the mis-labeling that is most concerning is when the label is predicted to be "No Action" when it is actually "Action Required." Thankfully that did not occur, but the mislabeling of a severity as "Consider Modification" when it should have been "Action Required" did. One of the drug pairs incorrectly labeled as such was clindamycin and erythrocin. Notably, clindamycin can be administered topically (on the skin tissue) or systemically (via blood circulation) and Lexi-Interact makes the distinction that the drug interaction only has a severity of "Action Required" when clindamycin is delivered systemically (Up-to-date). Erythrocin does not interact with clindamycin when delivered topically. However, the data from the FDA Adverse Event Reporting System makes no clarification on how clindamycin is delivered, which may account for the discrepancy in labeling. Similarly, Ciprofloxacin can be delivered ophthalmically (via the eye) or systemically (via blood circulation), but only when delivered systemically does it have a known DDI severity of "Action Required" with ibuprofen. The drug combination of ciprofloxacin and ibuprofen was another drug combination that was perhaps mislabeled due to unknown delivery mechanism.

7. Conclusion/Future Work

Overall, logistic regression was outperformed by all other algorithms, probably because of over-fitting, as seen by the discrepancy between train and test accuracy. Naive Bayes performed well, with a test accuracy of 93% and was greatly improved by the transformation of the input data by PCA as the initial data was incredibly sparse. Both Random Forests and SVM had test accuracies above 95%. Interestingly, Random Forest was minimally influenced by parameter tuning, while for SVM, it allowed for an increase in accuracy from ~80% to 96%. For both algorithms, the largest source of error came from mis-classification of the severity of certain drug pairs as "Consider Modification" when the correct label is "Action Required." This may be due to differing delivery methods, which change drug interaction presence and severity.

Now that we have trained a model that has relatively high accuracy and recall with known Lexi-Interact severity values, the goal would be to test it on drug interactions with known severity. Specifically, a drug pair would have a predicted severity and then a panel of clinical pharmacists, individuals familiar with clinical outcomes of drug interactions, would rate the validity of that predicted severity. This is a crucial step as there may be a selection bias when only training on drug pairs found within Lexi-Interact. Lexi-Interact is based on published literature, so these drug pairs are well-studied, perhaps because they are widely-used or treat a common disease. Doctors may also be more inclined to prescribe a drug pair if it has a known severity. As such, drug pairs with severity values in Lexi-Interact may have more associated event reports because they are given more often and thus the feature vector may be more informative than for drug pairs not found within Lexi-Interact.

8. Appendix

8.1 Code - https://github.com/kerdma6777/ddi-severity-classification

8.2 Contributions

Neal: data collection from Lexi-Interact, class imbalance experiments, wrote final paper

Katherine: data collection from Lexi-Interaction, data processing pipelines, feature engineering experiments

Works Cited

Bone, D., Goodwin, M. S., Black, M. P., Lee, C. C., Audhkhasi, K., & Narayanan, S. (2015). Applying machine learning to facilitate autism diagnostics: pitfalls and promises. *Journal of autism and developmental disorders*, *45*(5), 1121-1136.

Chen, D., Zhang, H., Lu, P., Liu, X., & Cao, H. (2016). Synergy evaluation by a pathway–pathway interaction network: a new way to predict drug combination. Molecular BioSystems, 12(2), 614-623.

Debruyne, P. R., Pottel, L., Lycke, M., Boterberg, T., Ketelaars, L., Pottel, H., ... & Rottey, S. (2012). Experience with Lexicomp® Online Drug Database for medication review and drug-drug interaction analysis within a comprehensive geriatric assessment in elderly cancer patients. Journal of Analytical Oncology, 1(1), 32-41.

Han, H., Wang, W. Y., & Mao, B. H. (2005, August). Borderline-SMOTE: a new over-sampling method in imbalanced data sets learning. In *International conference on intelligent computing* (pp. 878-887). Springer, Berlin, Heidelberg.

Kuhn, M., Campillos, M., Letunic, I., Jensen, L. J., & Bork, P. (2010). A side effect resource to capture phenotypic effects of drugs. *Molecular systems biology*, *6*(1), 343.

Reis, A. M. M., & Cassiani, S. H. D. B. (2010). Evaluation of three brands of drug interaction software for use in intensive care units. Pharmacy world & science, 32(6), 822-828.

Smithburger, P. L., Gill, S. L. K., Benedict, N. J., Falcione, B. A., & Seybert, A. L. (2010). Grading the severity of drug-drug interactions in the intensive care unit: a comparison between clinician assessment and proprietary database severity rankings. Annals of Pharmacotherapy, 44(11), 1718-1724.

Tatonetti, N. P., Patrick, P. Y., Daneshjou, R., & Altman, R. B. (2012). Data-driven prediction of drug effects and interactions. *Science translational medicine*, *4*(125), 125ra31-125ra31.

Up-to-date [online]. Lexi-interact online. www.uptodate.com/crlsql/ interact/frameset.jsp (accessed 3 June 2019).

Zitnik, M., Agrawal, M., & Leskovec, J. (2018). Modeling polypharmacy side effects with graph convolutional networks. Bioinformatics, 34(13), i457-i466.