Predicting Global Supply Chain Outcomes for Essential HIV Medicines using Machine Learning Techniques

Author: Tichakunda Mangono
Capstone Project: Udacity Machine Learning Engineer Nanodegree
Date: September 27th, 2017

Synopsis: A combined "classification-then-regression" machine learning model can avoid the public health and economic costs associated with delayed deliveries of HIV medicines. An ensemble classification algorithm, Extra Trees, is able to detect **1 in 2** delayed item deliveries. This is a significant improvement from a null hypothesis model which would detect only **1 in 9** delayed items and a considerable improvement from benchmarked Random Forest classification algorithm which catches **1 in 3** delayed items. Once delayed items are identified, an Extra Trees regression algorithm can predict the length of delay to within **12 days** (RMSE) with an R-Squared of **0.86**, which is similar to the benchmarked Random Forest regression performance. So, while there was no significant improvement in the regression part, the combined classification-then-regression model for Extra Trees does significantly better than the benchmark.

Definition

Project Overview

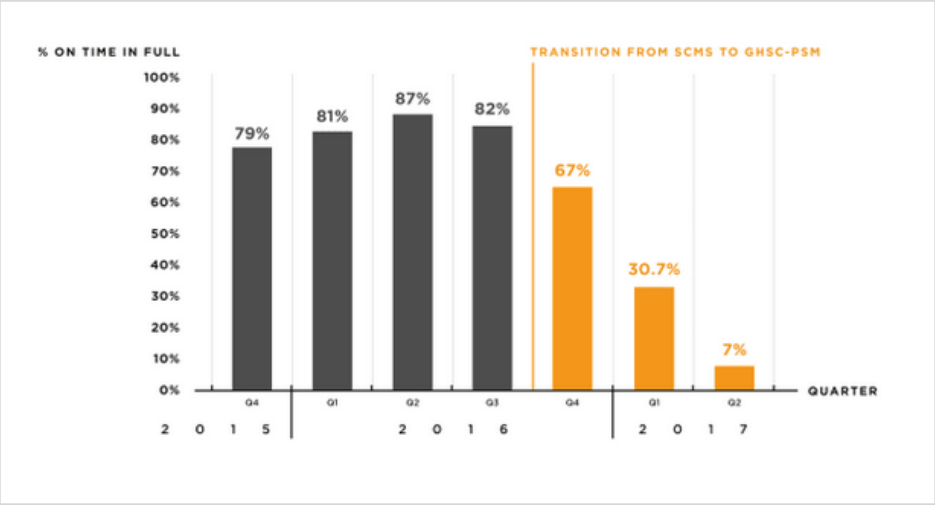
More than **36.7 million**[1] people in the world were living with HIV in 2016 and every year, about **1 million** people worldwide die from AIDS-related causes. While this death rate has decreased significantly (by 38%) since 2001 and continues to decline, about **1.8 million** people also became newly infected in 2016 alone. The epidemic disproportionately affects low income countries in Eastern and Southern Africa where women, adolescents and key populations like female sex workers and LGBTQ individuals are the most affected groups. There is currently no cure or vaccine for HIV and while several prevention methods exist, their efficacy is reduced by several factors, including economic and psycho-social factors. Fortunately, it has been shown that treatment can not only prolong life but also prevent the spread of HIV as it lowers the viral load of people living with HIV to a non-infectious level. However, of the **36.7 million** people living with HIV in 2016, only **19.5 million** were receiving this life-saving treatment.

The President's Emergency Plan for AIDS Relief (PEPFAR), a US government program is a key player in the procurement of drugs, testing and laboratory kits for HIV. One of its agencies spends more than **$9.5 Billion** per year on procurement of essential medicines to

[1] http://www.unaids.org/en/resources/fact-sheet

fight HIV/AIDS around the world. It is critical that these procurements arrive on time and in full to meet the needs of People Living with HIV (PLHIV) around the world, however this is not always the case. In fact, recent changes have resulted in drastic declines in supply chain performance as shown in the chart above. Thus, knowing whether or not HIV drugs are delivered on time and how long potential delays will be is very important. This study will use publicly available data from PEPFAR over the years 2006-2015 to determine the factors influencing timeliness of pharmaceutical deliveries as well as use these factors to develop a model that can predict if and by how long a particular HIV commodity will be delayed in delivery. While more and more supply chain analysis has begun to incorporate machine learning, it is especially aimed at demand forecasting as opposed to predicting the lead-time directly. However, the approaches taken in some academic studies[2] e.g. SVMs and RNNs have shown great promise. Similar problems like predicting flight delays[3] and improving flight efficiency have also been solved using machine-learning.

Figure 1: Quarterly Supply Chain Metrics for USAID's global supply chain for medicines, managed by Chemonics



Problem Statement

Timeliness of HIV procurement is critical to the efficiency and impact of the program in saving lives, controlling and eventually eliminating HIV. Delays in supply of commodities result in extra costs in terms of storage, coordination and most importantly, lost lives in the case of HIV medicines. This study will use publicly available supply chain data to determine the most important factors in predicting whether HIV drugs are delivered on time or not. It will then use these factors to predict how long delays are likely to be, thus allowing HIV/Supply Chain program managers to know **when and which products are likely to be**

[2] https://www.researchgate.net/publication/222928270_Application_of_machine_learning_techniques_for_supply_chain_demand_forecasting
[3] https://www.kaggle.com/c/flight

delayed, as well as **the extent of the delay** so that they can take mitigating action to save lives and avoid additional supply chain costs.

Solution Statement

This study used a combined model which uses **classification machine learning algorithms** to predict whether a particular product is delayed or not and then use **regression analysis** to predict the length of the delay using the subset of the data which the classification predicted will be delayed. This will maximize the utility of the complete model since it follows the natural decision-making process – a supply chain program manager would normally care about the products that will be delivered late and within those, focus on the ones that will likely have the longest delays first, thus allowing them to prioritize supply chain/logistics management and solve the biggest problems first.

To select the best model, both the classification and regression versions of the following models will be explored evaluated against predetermined benchmarks of Random Forest model with default parameters in SciKit-Learn : i) ExtraTrees ii) XGBoost iii) Support-Vector Machines (SVM) and iv) Multi-Layer Percepton (MLP). Random-Forests, ExtraTrees and XGBoost are proven **high-performing ensemble** algorithms which can do **automatic feature extraction** while SVMs perform very well with **high-dimensional data** and can **detect non-linear relationships** if the right kernel is used. Finally, MLPs are useful for high-dimensional time-series data. The above advantages of these algorithms are well-suited to the selected dataset which has several categorical columns that will increase dimensionality and potentially be non-linearly related to the target variable after data transformation. Finally, the data is well-suited for this overall approach since our target variables is well-defined on the data i.e. delay occurrences and duration can be determined by data on scheduled versus actual delivery dates, allowing clear quantification and measurement of the problem and solution. This study's results will be applicable to future instances of supply chain orders, and thus it is applicable to future occurrences of similar supply chain data observations and useful for planning purposes.

Evaluation Metrics

The resulting combined models will be evaluated based on 4 metrics: **Recall** and **F1-Score** for classification, to balance the recall/precision trade-off, especially because the dataset is unbalanced with a ratio of **1:9** between the positive and negative class respectively. For the regression part of the model, the **R-squared** and **Root Mean-Squared Deviation (RMSD)** will be used to evaluate how well the regression model can predict the direction and length of delays in HIV medicine deliveries.

i) **Recall**: measures the success rate of correctly labeling the positive items i.e. what proportion of the positive labels did we successfully identify? This is important because it tells us what proportion of delays we can actually predict.

Recall = True Positives / (True Positives + False Negatives)

ii) **F1-Score** is an average (harmonic mean) of the recall and precision scores.

F1-Score = 2*(Recall * Precision) / (Recall + Precision)

where ***Precision = True Positives/ (True Positives + False Positive)*** *and* ***Recall = True Positives / (True Positives + False Negatives)***

iii) **R-squared** is the "coefficient of determination" which measures the amount of variation in the data that is explained by the model, again as a percentage/fraction of total variation.

$$r = \frac{n(\Sigma xy) - (\Sigma x)(\Sigma y)}{\sqrt{[\,n\Sigma x^2 - (\Sigma x)^2\,][\,n\Sigma y^2 - (\Sigma y)^2\,]}}$$

Here, r represents R-squared, n is the number of observations and x and y are the feature and target variables respectively.

iv) **RMSD** measures the average size (absolute value) of the error that the model makes when predicting continuous target variables e.g. days late/delay in this case.

$$\text{RMSD} = \sqrt{\frac{\sum_{i=1}^{n}(\hat{y}_i - y_i)^2}{n}}.$$

Here, "y-hat" are the predicted values and "y*i*" are true values of the target variable. "n" is the number of observations in the dataset.

Methodology & Analysis

Data Preparation

This study uses data from The President's Emergency Plan for AIDS Relief (PEPFAR) program's Supply Chain Management System (SCMS) data made publicly available online through the website: https://data.pepfar.net/additionalData . It has over 10,000 observations of unique HIV medicines/products with **33 feature columns** of product details, country, manufacturer and shipment details including - order, purchase and delivery dates. Two additional columns for **the target variables** ("on-time" a binary viable and "delay", a continuous variables will be derived from the existing delivery date-time columns). While the data has some limitations where products are sometimes consolidated into large shipments to save on costs, the availability of anticipated delivery and actual delivery dates makes this appropriate for this study. See appendix for list of dataset features/inputs.

Additional data sources were to consulted to develop new predictive features: **Logistics Performance Index** data from the World Bank **Fragile State Index** data from Fund for Peace data; and finally Factory location and continent from the googlemaps API:

Handling Missing Values and Mismatched data types: 3 variables: Dosage, Shipping Mode and Line Item Insurance had missing values (1736, 360, and 287 rows respectively) which were imputed using a combination of summary statistics at the appropriate level of granularity e.g. the mean or mode at the year-country-item level. In addition to missing values, some features also had the wrong data types assigned to them: some numerical or date features were classified as strings because of the type of placeholder used in data entry. These were converted and imputed as follows:

(i) **Purchase Order dates** (used derivation from delivery date scheduled by subtracting appropriate average time from order to delivery)
(ii) **Purchase Quotation dates** (derived from purchase order dates once imputed)
(iii) **Weight** – used shipping numbers to extrapolate topline vs. bottom (bundled) and unbundled shipments. Filled in resulting missing values with most appropriate average e.g. item, molecule, or dosage form level. Calculated average weight for each item, molecule or dosage group then multiplied by line item quantity to recover the weight. Assumption is that product unit weights should remain the same.
(iv) **Freight Cost** – required similar treatment as weight, but this time calculated as a proportion of Line Item Weight at the right level.

Feature Engineering

Feature Extraction by combining and transforming existing features as follows: i) **Date-time** to capture time aspect for purchase order dates, and scheduled delivery date (year, day, week quarter etc.); ii) **Time-Series** to capture trends and autocorrelation through lagged variables and rolling statistics iii) **Numeric**: counts, sums, proportions and measures of central tendency to estimate the impact of volumes, value and magnitude of activity; iv) **Categorical**: weight captured separately, shipment configuration, freight cost included commodities, or invoiced separately etc. iv) **Predicted variables** i.e. a categorical class variable "delayed" for when an item was delivered past its scheduled delivery date and a time-delta feature capturing number of delays delayed (i.e. actual delivery date less scheduled delivery date).

Feature Creation was done by sourcing and transforming data from external sources; data on Fragility State Index[4] (FSI) for country stability, Logistics Performance Index[5], and Factory location, country and continent[6].

Exploratory Data Analysis (EDA) & Feature Selection

General feature profiles

[4] http://fundforpeace.org/fsi/excel/
[5] https://lpi.worldbank.org/international/global?sort=asc&order=Infrastructure
[6] http://maps.googleapis.com/maps/api/geocode/json?

While there a lot of important features, there was some high multi-collinearity between several of them e.g. Line Item Weight, Value and Insurance are often highly correlated, which would be challenge for linear regression. However, methods such as Random Forests can easily deal with such a set of features. Below is a summary of EDA findings by type of feature/data.

i) Trends

- *Date-Time* - The years 2010, 2011, 2013 and 2014 clearly had above average proportions of delayed deliveries. Monthly, January had high rate of delays and more delays were seen over the weekend (Saturday and Sunday). However, the quarter delivery was expected had no noticeable signal.
- *Time-Series and Autocorrelation* – Moderate positive autocorrelations (up to 0.41) suggest that an item from a particular vendor is more likely to be delayed if that vendor has ever delayed delivery for several items (cumulative statistics) or if the number of recent delays from this vendor has increased (rolling statistics)

ii) Pairwise Correlations

- *Product-level:* High correlation between the values such as price, weight, quantity, value, insurance and freight costs indicated possible collinearity which would be an issue for a linear regression model.
    - Individually, these quantitative features have wide interquartile ranges and skewed distributions, so will need to be transformed to logarithmic scale and standardized/normalized
    - Additionally the correlations of e.g. weight and value/price of the products are bifurcated with 2 different slopes. This suggests two pricing or shipment systems e.g. bulk vs. single shipments or brand name vs. generic formulation.
    - The product-level features had some correlation with the delayed variable but they were generally on the lower side for any individual feature, suggesting the need to aggregate these to an entity level e.g. by country, vendor, factory, molecule test or brand
- *Entity level:* Once aggregated, the volumes, value and quantities of products for each entity are moderately correlated with delays, with the vendor metrics being particularly dominant in terms of correlation.

iii) Country Fragility and Logistics Performance Indices

- The higher the volumes and values of items delivered between origin and destination countries, the higher the rates of delays
- Deliveries from more fragile countries to other fragile countries are more likely to be delayed, with the origin country stability having more influence
- The better the logistics metrics of the origin country, the lower the rate of delays
- In summary, country fragility and logistics performance are important, with the supplier/origin country factors having more influence than destination factors.

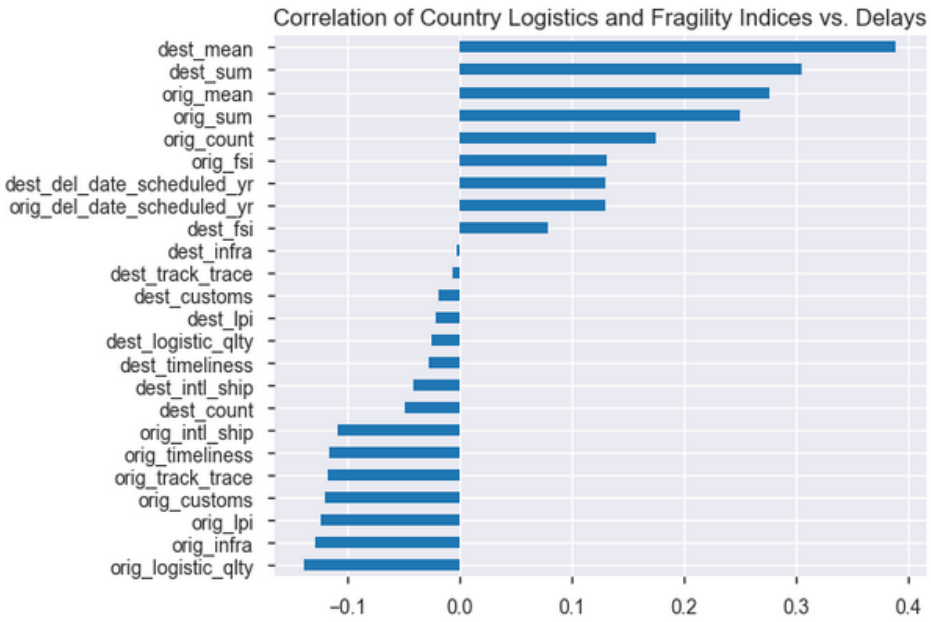


Figure 2: Plot of Correlation of Origin ('orig') and destination ('dest') country fragility and logistics indices with average delays

Dimensionality Reduction using Principal Components Analysis

Explained Variance - The first and second principal components explain **~ 31%** (see above) of the total variance in the data, which is quite significant. The first four principal components explained almost **43%** (see above) of the data which is not all of the variance in the data but significant enough to inform a useful model. Characteristics of the four principal components:

- *First Component:* This dimension explains about **17%** of the variance in the data. Looking at the composite features, one can see that these are the ***Supply-side factors*** which explain a lot of the variation e.g. origin geography characteristics like logistics quality and fragility of the state, vendor volumes, brand and prices.
- *Second Component:* This dimension explains about **14%** of the variance in the data. These factors are mostly about ***the Customer*** e.g. destination geography, logistics and volumes, recipient clients etc.
- *Third Component:* This dimension explains about **6%** of the variance in the data. It is mainly influenced by ***product/item level*** details, product value, and specifications as well as history of delay in delivery
- *Fourth Component:* This dimension explains about **5%** of the variance in the data. It is also mainly influenced by ***product/item level*** details in a slightly different way history and trends of being delayed etc.

Feature Importance using Random Forests

- *Overall:* individual "importance" for each feature is relatively small since several of them are also correlated with each other and thus have to share the importance along a particular common dimension/axis (see PCA analysis above).
- This would be an issue with linear regression, which would require combination of the importance/variance into one feature. However, tree-based ensemble methods like Random Forest can deal with this very easily.
- *Importance rankings*: are consistent with PCA findings as well as the general profiling of individual features and pairwise correlations above:
    - The lagged, cumulative sums, minimum as well as the lagged rolling mean of delays as well as delay length(days)
    - Supply factors like vendor volumes and quantities follow. This is very consistent with the Principal component analysis findings. Brand elements/influence is also quite visible
    - Product-level characteristics like value, quantity and price come up afterwards, mixed with some client-side factors like destination country stability and logistics, and origin logistics and quality

Model Benchmark

The solution model is combination of two algorithms working together sequentially; thus, the benchmark model will also require a two-part benchmark. In order to make clear objective comparisons, the same model, **Random Forest will be used as the benchmark for both classification and regression**. The study will use the default versions of the Scikit-Learn implementation of these models. Results:

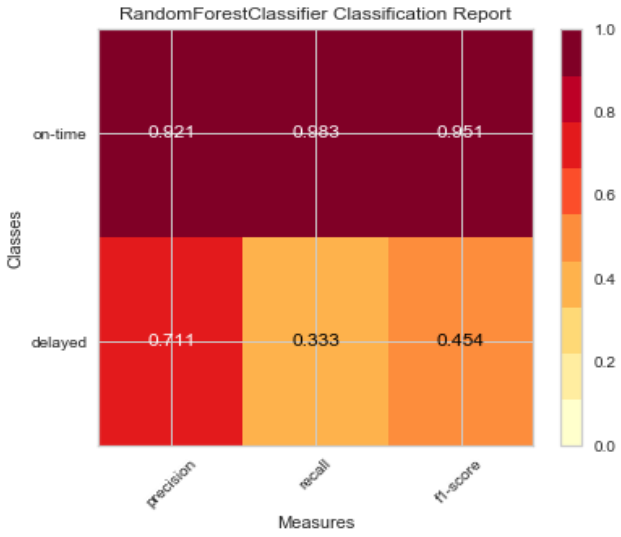


Figure 3: Classification benchmark results for default SciKit-Learn Random Forest algorithm.

- *Classification results with benchmark Random Forest Classifier.*
    - *Recall:* **0.33**

    - *F1-score*: **0.45**
    - *Total:* **134** *instances of delayed delivery correctly identified*
- *Regression results with benchmark Random Forest Regressor:*
    - *R-squared:* **0.85**
    - *RMSE:* **13 days**

Model Selection

After pre-processing the data through a pipeline for logarithm, standard scaling, one-hot and label encoding as well as oversampling to balance the classes, the following models were compared:

i) ***Classification***: *LinearSVC, SVC, KNeighborsClassifier, LogisticRegressionCV, LogisticRegression, SGDClassifier, BaggingClassifier, ExtraTreesClassifier, RandomForestClassifier, MLPClassifier ,GaussianNB, LinearDiscriminantAnalysis*

ii) ***Regression***: *LinearSVR, SVR, KNeighborsRegressor, LinearRegression , SGDRegressor, BaggingRegressor, ExtraTreesRegressor, RandomForestRegressor, MLPRegressor*

Final Models selected: ***Extra Trees Classifier*** for classification and ***Extra-Trees Regressor*** for regression.

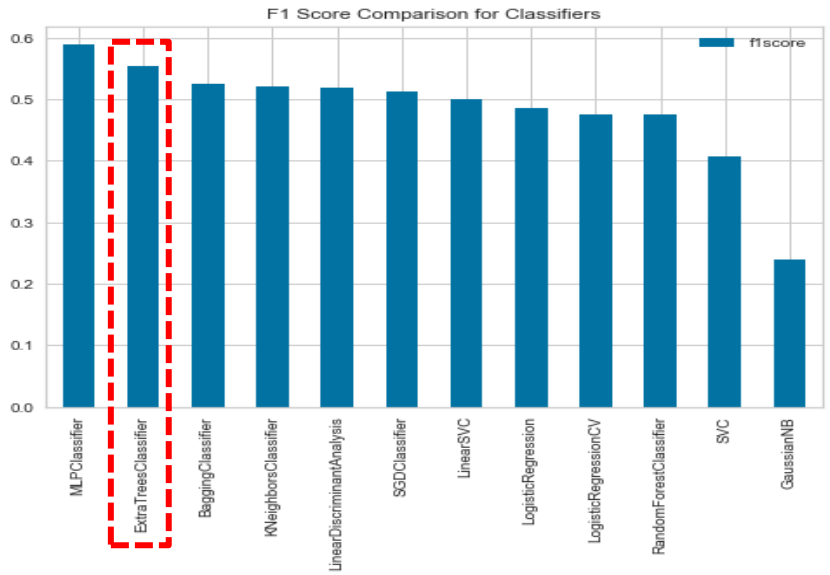


Figure 4: F1 Score comparisons for classification model selection

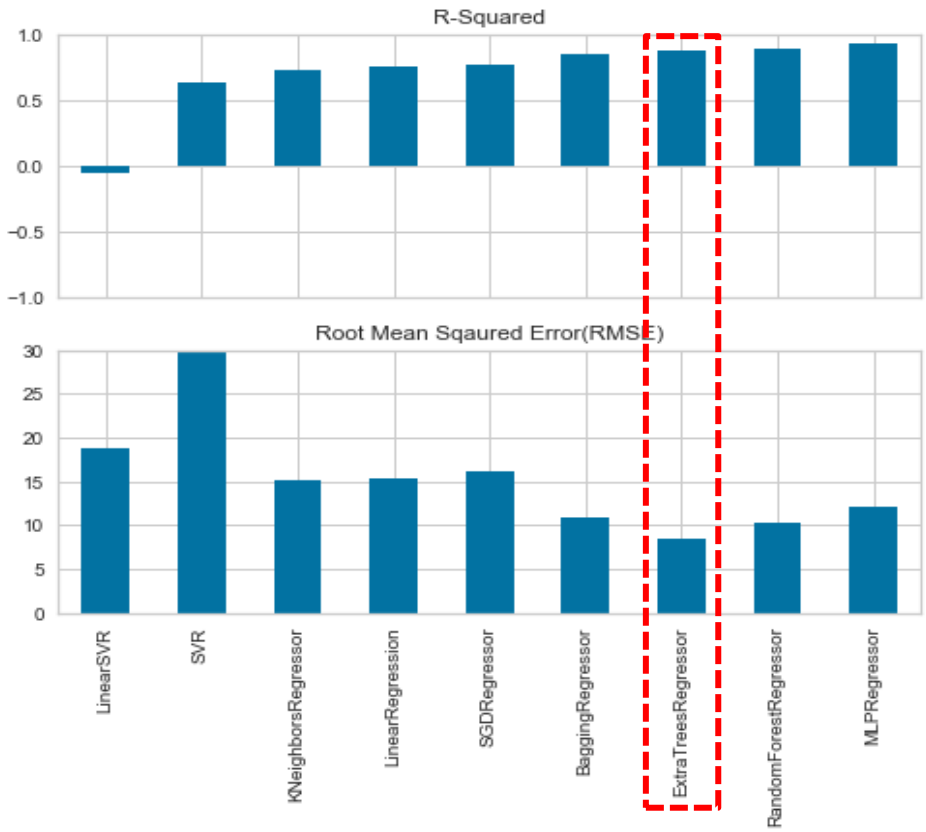


Figure 5: R-squared and RMSE comparison for model selection

Extra Trees Algorithm Specifics and Relevance

The Extremely Randomized Trees (Extra-Trees)[7] is supervised learning algorithm similar to the Random Forest algorithm in that they are both ensemble methods which used simple classification and regression decision trees as their building blocks. However, Extra Trees differs from Random Forests while Random Forests tries to find an optimal cut-point for each one of the *k* randomly selected features at each node (bootstrapping), Extra trees instead selects cut point at random and then averages the results. This results in "weak-learner" trees whose errors are uncorrelated, thus randomizing the cut-points and averaging result has a smoothing effect on the predictions of these trees. When combined with careful feature analysis to remove redundant features, this approach will simultaneously increase bias and reduce the variance, resulting in better model accuracy[8]. Additionally, it achieves higher computational efficiency by not trying to find optimal cut points.

The Extra Trees model is appropriate for this data and problem because of the following key features: i) it deals well with high dimensionality which is appropriate given the high

[7] http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.65.7485&rep=rep1&type=pdf
[8] http://www.montefiore.ulg.ac.be/services/stochastic/pubs/2006/WEG06/robust-trees.pdf

cardinality of most of the categorical variables in this problem. ii) it deals well with multi-collinearity and is in fact quite robust to this aspect iii) it can handle heterogeneity amongst the features as well as skewness within individual features iv) due to smoothing, it often leads to increased accuracy in the presence of a high number of continuously varying numerical features.

Implementation of Model, algorithm and techniques

The key steps to implementing this model are detailed in the accompanying python code and JuPyter notebooks, below is a brief summary of the nuances in the approach:

1. *Data cleaning* – loading the data was straightforward given it was stored in standard excel format. However, it was important to understand the column names, types and distributions as well as supply chain terminology. A reference dictionary with lookup functionality for each column was developed for this purpose. Imputing missing values was then done using either mean or mode methods. For dates that were missing, an estimate based on related date columns was made e.g. purchase order dates were extrapolated from expected delivery dates while purchase quote dates were in turn derived from purchase order dates. The code for this required some proficiency with pandas date time and indexing functionality. To estimate missing weight and freight cost, this study took advantage of the relationships of these variables to other item-level features i.e. *packet price = unit price * number of units*; *line item value = line item quantity*packet price*; average weight of standard items should stay constant, and thus missing weight values can be imputed using this *item weight * item quantity* and *freight cost is proportional to weight* and dependent on whether items where single or bundled in a particular shipment.

2. *Feature Engineering:* Dates: year, month, day, weekday, quarter, week-of-year were extracted using pandas date-time functionality to capture time aspects as categorical variables. Numeric: counts, sums, proportions and measures of central tendency were calculated at country-year, factory-year, vendor-year, molecule-year, brand-year levels an then these were merged with the item-level data. Categorical: after cleaning up the weight and freight cost columns, there was additional information from which the following labels were extracted: weight captured separately, shipment configuration, freight cost included commodities, or freight invoiced separately. For shipment configuration, it was challenging to separate into single vs. bundled shipments but using string functions and regex, it was possible to isolate the id of each top-line item and then group all the items with which it was bundled. The remaining items constituted the single shipments. Time series variations were also captured at the vendor-date level using pandas group-by, rolling and cumsum functions to capture the short term and long term autocorrelation of delayed items and number of days of delay. The predicted variable itself, "delayed" was derived as the difference, in days, of the date delivered at client site and the scheduled delivery date. Extra care was taken to remove the predictor variable from the feature. External features on logistics and country fragility which turned out to be strongly predictive were also obtained from other sources. This data

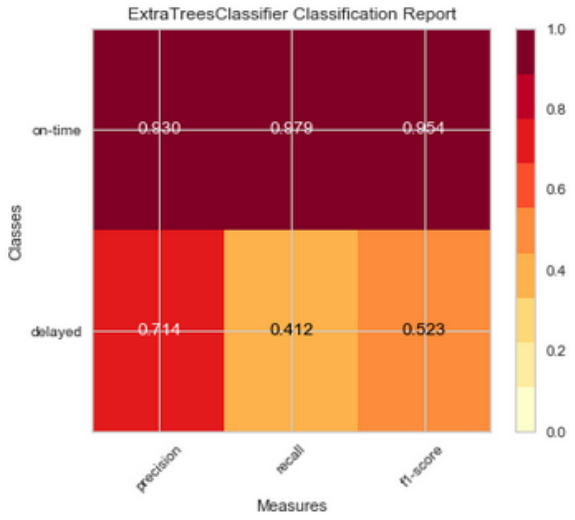
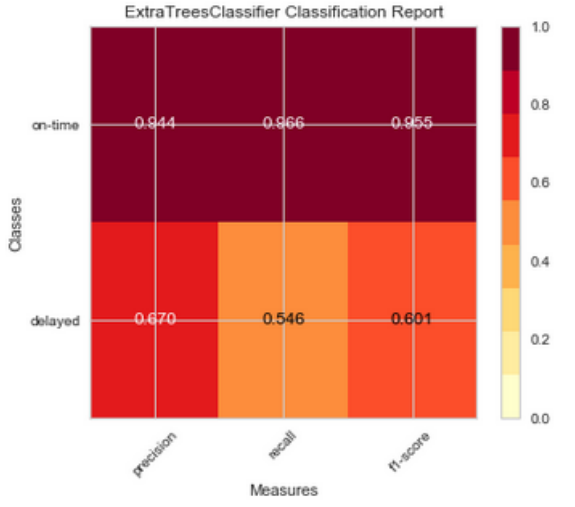
required cleaning, entity resolution (e.g. country names) and minimal missing value imputation before joining to the main item-level dataset.

3. *Feature selection* – Several methods, from univariate, bivariate and trends analysis to dimensionality reduction and feature importance was used to select the most predictive, least collinear, scaled and transformed features. Using several methods allowed complementary and robust approach to the features selected. These methods are already outlined in detail in the EDA & Feature Selection section above and in the accompanying code.

4. *Model benchmarking* – see the Model Benchmarking section already described.

5. *Model Selection* – see Model Selection section already described, and Discussion of key challenges section.
.
6. *Final Model results* – this is detailed in the Final Model Results section below along with key findings and challenges.

Model Improvement and Fine-tuning

	Extra Trees Initial Results	**Extra Trees Final Results**
CLASSIFICATION		
REGRESSION	*R-squared:* **0.92** *RMSD:* **8.5 days** *Total delays captured:* **171**	*R-squared:* **0.86** *RMSD:* **11.8 days** *Total delays captured:* **221**

For classification, a clear improvement was observed in the Recall and F1-Score metric for the selected model. For regression, both the R-squared and RMSD fared a little worse in the final model than initial model due to the change in denominator, the final model was classifying and regressing on more items and as such the additional items may have slightly

higher internal variation than the items picked up by the initial model. Note, however, that the final model still outperforms the benchmark Random Forest model.

The additional features introduced from external sources as well as those extracted from the existing features were critical to model improvement. Especially logistics performance, country fragility indices, total/mean vendor quantities and shipment values as well as rolling and cumulative statistics of delays and length of delay. The details on this are outlined in previous sections. Another key element in improving performance is the use of oversampling technique to overcome the imbalance in the data set. Oversampling, resamples the minority class to produce similar data points in that class and equalize the weighting of the classes in the algorithm. The SMOTE implementation of imblearn library was used to accomplish this.

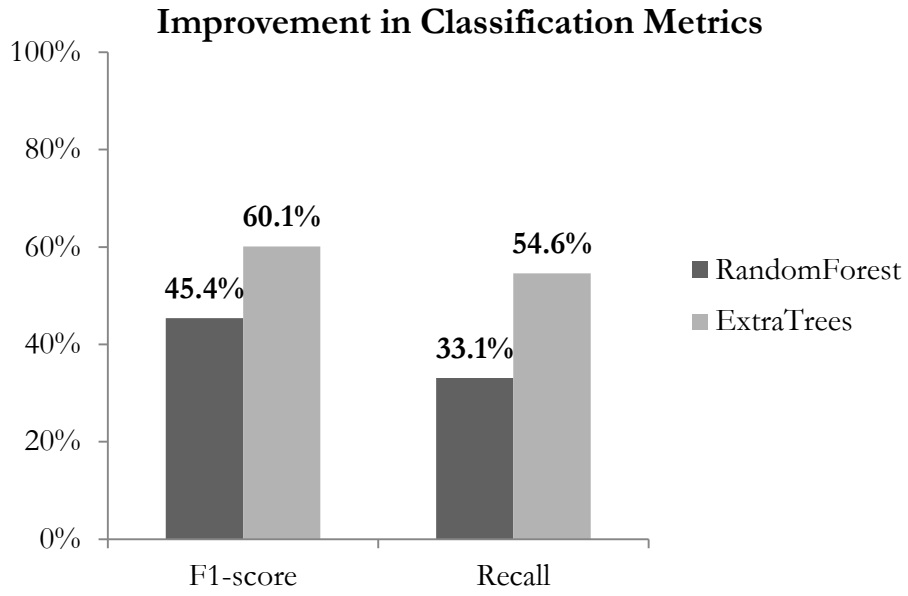
Finally, the hyper parameters were obtained through trial and error method as opposed to extensive grid search. This was deemed sufficient after narrowing down a few of the most important hyper parameters that often lead to model improvement and try some order of magnitudes for those values manually and observing the effect on the results. In general, the higher the n_estimators, max_features and max_depth, the more able the Extra Trees model is to predict accurately. Since it works with randomly-chosen cut points, it requires more trees to reduce the variance, a significant (but not too large) number of features on which to split (so as to capture most of the signal available in the data), as well as a good number for maximum depth to allow the model to capture all the nuances in a particular feature (thus increasing bias). All these features increase the accuracy of the model (see explanation of model algorithm above along with why it works well for this type of problem).

Final hyper parameters chosen:
Classification → ExtraTreesClassifier (n_estimators=900, max_features= 50
, criterion= 'entropy',max_depth= 50, random_state=121)
Regression → ExtraTreesRegressor(n_estimators=900 ,max_features= 50,max_depth= 50, random_state=121)

Final Model Results

Summary: Supplier-side factors such as origin country stability, vendor and brand volumes as well origin country logistics environment explain significant variation in the data. Together with customer/receiver-side factors, they explain about a third of the variation in the data. The rest of the variation is due to product level characteristics like volumes, value/price, and weight as well as how they evolve over time (time-series). Of note is the significant auto-correlation where vendors who have delayed items in the past as well as more recently are more likely to delay deliveries again in the future. These insights were used for feature selection for the final classification and regression models.

The Extra Trees Classifier and Extra Trees Regressor were selected as the best algorithms for the classification and regression tasks respectively. Both algorithms outperformed the benchmark Random Forest and several other algorithms. The Extra Trees Classifier improved the Recall by **65%** (from **33.1% to 54.6%**) and the F1-score by **32%** (from **45.4% to 60.1%**).



Metric	Random Forest	Extra Trees	Improvement
F1-score	45.4%	60.1%	**32%**
Recall	33.1%	54.6%	**65%**
R-Squared	85.8%	86.3%	**0%**
RMSE (days)	12.96	11.97	**8%**

Final Model Robustness

In general, the final model selected is quite robust to outliers and has optimized parameters. However, there are several concerns to be wary of when considering the use of this or a similar model for real life predictions. Below is a discussion of the final model qualities, parameters and robustness.

Parameters: A discussion of each parameter and how it helps the model is handled in a previous section "Implementation of Model, Algorithm and Techniques".

Outliers: Tree-based algorithms such as Extra Trees are inherently robust to outliers, oversampling is used to generate artificial samples to help the model learn better and enforce

the true class boundaries, log transformations and standard scaling used on the data/feature space increase the models robustness to outliers as well. Another aspect which this study did not do but could improve the robustness to outliers, is the use of another metric e.g. mean absolute error instead of mean squared error.

Unseen Data: Supply chain characteristics may change. The model relies on some features like past trends and although it includes cumulative and rolling statics to capture this, major changes such as drastic improvements or declines in supply chain performance can throw off the model's accuracy and its inherent historical trend assumptions. It may not easily/quickly pick up on the effect of such changes, requiring more data and time to re-train. Another important feature is that this model currently benefits from having the annually aggregated features but if used in real life, it would have to rely on say, the last 6 months of data depending on how far the year has progressed, and this is exacerbated by the seasonal element of demand-based processes such as product supply chains. Thus, it will take time to learn new trends in the data for that year and will increase in accuracy as the year proceeds. This is becomes obvious when considering a scenario where one country switches to new vendors for which no historical data exists. All this highlights the need to evaluate the model before deciding or continuing to use it as it may be vulnerable to drastic changes.

Discussion – Key Challenges, Lessons and Potential Improvements

While this study presented some initial data cleaning/analysis challenges, feature-selection and model-selection processes presented the most difficult challenges; the solutions to which led to important lessons about the machine learning process as this study evolved.

The major hurdles to seamless **feature selection** were high dimensionality, heterogeneity and multicollinearity in the feature space. *High dimensionality* was caused by high cardinality of categorical variables such as number of vendors, number of factories, number of countries and the number of weeks in a year etc. which increase the total list of features available (713 after one-hot-encoding). *Heterogeneity* of the features resulted from the combination of several metrics measured by different metrics at different scales and *multi-collinearity* was caused by the high correlations amongst variables such as weight, price, quantity and insurance for line items delivered as these tend to be closely related in the supply chain context. To address the "*curse of dimensionality*" this study identified the most important features through careful dimensionality reduction (principal component analysis) and confirming these findings by studying the relative feature importance using ensemble tree-based techniques. Data transformations such as logarithms and standard-scaling addressed feature-heterogeneity as well as skewness in individual features. While dimensionality

reduction helped to deal with some of the multi-collinearity (by clustering related features), pairwise correlation matrices were also helpful in this regard. Thus, highly correlated features were easily identified and eliminated if necessary. An important lesson to note is that several of the most important features were the result of feature creation and extraction as opposed to the original features.

In terms of **model selection**, it was important to select a model that would be robust to the challenges mentioned above i.e. high-dimensionality, multicollinearity and heterogeneity amongst the features, so naturally tree-based ensemble models offered clear advantages over other simpler models such as linear or logistic regression which are easily affected by these problems. Another complication was that the target space for this problem was highly imbalanced (**1:9**), so this required careful selection of the metrics to measure e.g. recall instead of accuracy to focus on the actual performance at classifying the low-representation class of delayed items. Additional techniques such as oversampling were also explored to deal with the imbalance and improve the final model's metrics. This study also highlighted the highly iterative nature of model selection, necessitating several repeated processes with slightly changed parameters. In order to streamline the process it was critical to use robust, reusable code e.g. generalized functions, classes and pipeline objects to speed up the iterations.

Finally, by changing a couple more aspects of this model such as the scope and the feature set that can be changed to potentially **improve the results**. Changing the scope by focusing on one country with a high delayed item rates, for example, may lead to a more localized hence accurate model for that country. Since vendors/supply side factors seem to explain a significant portion of the model, future studies may add more features such as vendor ratings over time (e.g. from Yelp ratings or Twitter sentiment analysis) to determine how customers feel about a vendor at a given time point. Finally, a supply chain professional would probably have more domain knowledge to add to this study especially around explanation of trends and other hidden variables.

Conclusion

A combined "classification-then-regression" machine learning model can avoid the public health and economic costs associated with delayed deliveries of HIV medicines. An ensemble classification algorithm, Extra Trees, is able to detect slightly more than **1 in 2** delayed item deliveries. This is a significant improvement from a null hypothesis model which would detect only **1 in 9** delayed items and a considerable improvement from benchmarked Random Forest classification algorithm which catches **1 in 3** delayed items. Once delayed items are identified, an Extra Trees regression algorithm can predict the length

of delay to within **12 days** (RMSE) with an R-Squared of **0.86**, which is similar to the benchmarked Random Forest regression performance. So, while there was no significant improvement in the regression part, the combined classification-then-regression model for Extra Trees does significantly better than the benchmark.

Table 1: Available Input Features in the PEPFAR Supply Chain Data Set

#	FieldName	FieldDescription	DataType
1	**ID**	Primary key indentifer of the line of data in our analytical tool	Number
2	**Project Code**	Project code	Text
3	**PQ #**	Price quote (PQ) number	Text
4	**PO #**	Order number: Purchase order (PO) for Direct Drop deliveries, or Sales Order (SO) for from Regional Delivery Center (RDC) deliveries	Text
5	**ASN/DN #**	Shipment number: Advanced Shipment Note (ASN) for Direct Drop deliveries, or Delivery Note (DN) from RDC	Text
6	**Country**	Destination country	Text
7	**Managed By**	SCMS managing office: either the Program Management Office (PMO) in the U.S. or the relevant SCMS field office	Text
8	**Fulfill Via**	Method through which the shipment was fulfilled: via Direct Drop from vendor or from stock available in the RDCs	Text
9	**Vendor INCO Term**	The vendor INCO term (also known as International Commercial Terms) for Direct Drop deliveries	Text
10	**Shipment Mode**	Method by which commodities are shipped	Text
11	**PQ First Sent to Client Date**	Date the PQ is first sent to the client	Date/Time
12	**PO Sent to Vendor Date**	Date the PO is first sent to the vendor	Date/Time
13	**Scheduled Delivery Date**	Current anticipated delivery date	Date/Time
14	**Delivered to Client Date**	Date of delivery to client	Date/Time
15	**Delivery Recorded Date**	Date on which delivery to client was recorded in SCMS information systems	Date/Time
16	**Product Group**	Product group for item, i.e. ARV, HRDT	Text
17	**Sub Classification**	Identifies relevant product sub classifications, such as whether ARVs are pediatric or adult, whether a malaria product is an artemisinin-based combination therapy (ACT), etc.	Text
18	**Vendor**	Vendor name	Text
19	**Item Description**	Product name and formulation from Partnership for Supply Chain Management (PFSCM) Item Master	Text
20	**Molecule/Test Type**	Active drug(s) or test kit type	Text
21	**Brand**	Generic or branded name for the item	Text
22	**Dosage**	Item dosage and unit	Text
23	**Dosage Form**	Dosage form for the item (tablet, oral solution, injection, etc.).	Text
24	**Unit of Measure (Per Pack)**	Pack quantity (pills or test kits) used to compute unit price	Number
25	**Line Item Quantity**	Total quantity (packs) of commodity per line item	Number
26	**Line Item Value**	Total value of commodity per line item	Currency (USD)
27	**Pack Price**	Cost per pack (i.e. month's supply of ARVs, pack of 60 test kits)	Currency (USD)
28	**Unit Price**	Cost per pill (for drugs) or per test (for test kits)	Currency (USD)
29	**Manufacturing Site**	Identifies manufacturing site for the line item for direct drop and from RDC deliveries	Text
30	**First Line Designation**	Designates if the line in question shows the aggregated freight costs and weight associated with all items on the ASN/DN	Binary
31	**Weight (Kilograms)**	Weight for all lines on an ASN/DN	Number
32	**Freight Cost (USD)**	Freight charges associated with all lines on the respective ASN/DN	Currency (USD)
33	**Line Item Insurance (USD)**	Line item cost of insurance, created by applying an annual flat rate (%) to commodity cost	Currency (USD)