Constructing a Predictive Model for Quota Fulfillment:

Implementing General Classification and Neural Network Models to Predict Whether a Garment Factory Team Meets a Productivity Quota

Siddharth Jaswal
Master's Student of Quantitative Economics
University of California, Los Angeles

Claire Zhang
Master's Student of Quantitative Economics
University of California, Los Angeles

Tanner Woods
Master's Student of Quantitative Economics
University of California, Los Angeles

Abstract — By using a dataset acquired from the UCI Machine Learning Repository concerning productivity of textile factory workers, this project focuses on implementing predictive classification models via logistic regression, the k-NN method, Keras deep-learning model and SVM method with or without Grid Search on transformed label outputs. To improve the generalizability of the model, our team decided to transform the raw discrete output labels of productivity ranges into a binary classification of (0, 1). Contextually, these new labels define a team whose actual productivity failed to meet their target, and a team whose actual productivity met or surpassed their target, respectively. Ultimately, our team determined that a simple logistic regression or grid-search SVM using SMOTE or ADASYN rebalanced data is likely to produce high-performance and reliable predictions.

Keywords — Garment, textiles, classification, k-NN, k-nearest neighbors, logistic regression, neural network, machine learning, Keras.

Nomenclature — Adaptive Synthetic Algorithm (ADASYN), area under curve (AUC), k-nearest neighbors (k-NN), mean absolute error (MAE), receiver operating characteristic (ROC), Support Vector Model (SVM), Synthetic Minority Oversampling Technique (SMOTE)

I. Introduction

As an industry born at the beginning of the industrial revolution, the textile and garment industry is a tried and true process of production. Improvements are undeniably still available — and innovations of manufacturing yet to be introduced still lie in the future — but the methods of productivity measurement in this industry are uniquely more available than others. Alongside this, the presence of the textile industry is concentrated in countries such as China, Bangladesh and India; these nations account for approximately 60% of worldwide production. The appeal for companies to focus their production efforts there can be derived from the staggeringly low cost of labor in areas requiring relatively simple manual work. While the real manufacturing process of textile production might be relatively proven, that does not necessarily entail that inputting human labor will result in a constant output of textile. Rather, and as human nature generally brings, a factory line will see fluctuations in output and productivity. Furthermore, the relatively low level of education amongst those employed in such factories can make ensuring worker productivity even more challenging. These factors — alongside many others left unlisted — can make it difficult for companies within the textile industry to ensure that output can meet demand; thereby creating the potential for unfulfilled orders and other forms of lost revenue.

II. Statement of Previous Work

Incidentally, our analysis and experiment runs parallel to the work performed by Abdullah Al Imran, Md Nur Amin, Md Rifautl Islam Rifat, and Shamprikta Mehreen. In their work, the authors attempt to construct a deep neural network in order to construct discrete predictions on worker productivity given a vector of input data. According to the authors, their team was successful in their experiments and produced predictions possessing a MAE of 0.086 in comparison to a baseline predictive error of 0.15.

III. Task Description

In this project, we intend to use four methods to predict productivity in the classification problem. Before beginning experimentation, we have to appropriately scale the dataset on a (0, 1) range; this is to smooth the process of the Keras model later on, as well as generally help prevent larger values in the set from overshadowing smaller ones. Following this, we will drop rows containing NaNs from the dataset; imputing values is not entirely necessary, as we will generate more values when balancing the data.

Following this step, we will assign binary productivity labels of (0, 1) to teams depending on the relationship of their productivity to their quota. After visualizing the productivity, we discovered that the class was imbalanced, so we implemented two oversampling techniques in order to ease potential issues of poor generalization. Over the course of model construction, we will fit models to the original dataset and both oversampled datasets to produce three distinct predictive engines.

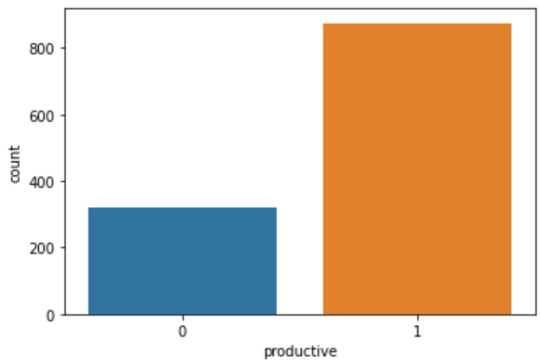
Finally, we will compare predictions from all four models — which will be trained on their own distinct training and validation samples — against their testing samples. From this comparison, we will select the most optimal predictive model by a measure of accuracy, precision, and recall scores; consideration will also be granted to AUC scores.

IV. Major Challenges and Solutions

Challenge of Unbalanced Data.

In a similar vein to problems involving the identification of financial or credit fraud, the dataset revealed that most factory teams do meet their assigned quotas. More precisely, early visualizations of the dataset revealed an approximate 2.6:7 ratio of quota failures to successes. The unbalanced nature of the dataset can lead to poor precision and recall statistics from the model: few examples of teams not making the quota inherently results in a model poorly trained on identifying such teams. This imbalance can be seen below in **Table 1**:

Table 1. *Comparison of textile factory teams flagged by their real productivity against their target productivity*



To rectify this we utilized two oversampling techniques: SMOTE and ADASYN. In brief, both methods implement a k-NN approach to constructing artificial instances similar to other minority points; more information can be found in Section VI.

Challenge of Optimal Hyperparameters.

Our second major issue centered on determining the appropriate optimal values for the hyperparameters across the three models. To this end, we decided to implement cross-validation techniques where appropriate to increase the likelihood of discovering the optimal hyperparameters.

V. Data description

For information on variables and constructed variables (e.g. Dummies, classification labels, etc.), please refer to **Appendix 1**.

VI. Data Processing

As mentioned above, the dataset was rescaled on a (0, 1) range via *sklearn* in order to standardize the dataset and attempt to normalize its distribution. The Z-scores for variables were calculated as follows:

$$Z = \frac{x - \mu}{\sigma}$$

where Z is calculated for every training sample.

Also discussed previously, the data was further processed via SMOTE and ADASYN techniques of resampling. In brief, SMOTE will generate one artificial instance for every original minority instance. Meanwhile, ADASYN will apply a weighted distribution to less artificially rebalance the dataset. In other words, ADASYN should produce fewer artificial samples and a distribution that is — while more balanced than the original distribution — less unnaturally balanced in comparison to SMOTE. Ultimately, both resampling techniques help construct a dataset more favorably distributed while remaining true to its features. The resampled datasets can be seen in the following **Table 2** and **Table 3**; note the marginal decrease between SMOTE and ADASYN for the productive flag of 0.

Table 2. *Productivity flags under SMOTE resampling*

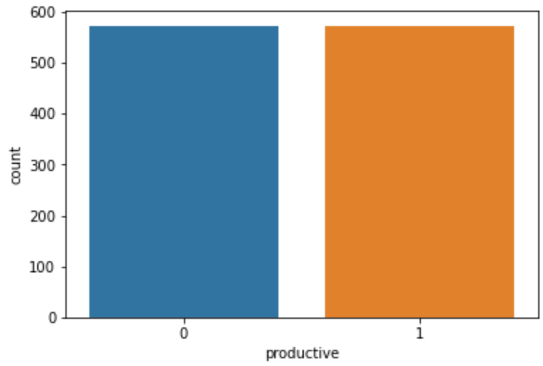
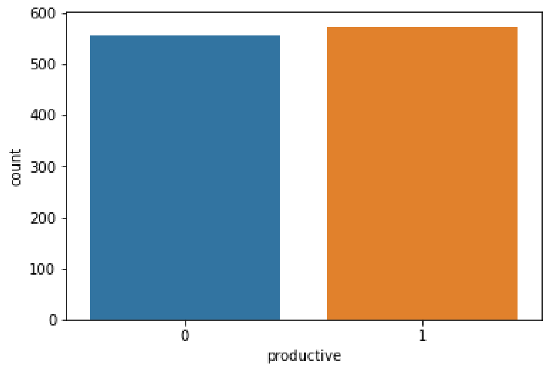


Table 3. *Productivity flags under ADASYN resampling*



VII. Algorithm Description

Logistic Regression.

Logistic regression is a process of modeling the probability of a discrete outcome given an input variable. It serves as a simple and efficient method for binary classification and regression problems under conditions with linearly separable classes. Classes are assigned depending on the fulfillment of the following conditions, where *x* is defined as the input feature:

$$g(x) = \frac{1}{1+e^{-x}} = \{> 0.5 = 1, \ <= 0.5 = 0\}$$

k-NN Method.

The k-NN method is a nonparametric supervised learning method, where inputs consist of *k* closest training instances in a dataset. After calculating the distances of the input vector from a selected feature, a classification is assigned to this feature based on the distance-weighted plurality of the input vector classifications; weighting is not necessary, but can improve results.

Keras Neural Network.

In the context of this paper, the Keras neural network implemented is a relatively simple sequential model composed of a linear two-layer stack. The model is then fitted over 100 epochs containing batch sizes of 32 instances against pre-prepared validations samples.

Baseline SVM and SVM with Grid Search CV.

SVM models are supervised learning models that can function well in high-dimensional input spaces and require data rescaling for proper function. In this paper, we implemented two SVMs across the three datasets. The first processes the datasets with default hyperparameters, while the latter implements a grid search cross-validation to produce more optimal hyperparameters.

VIII. Evaluation Metrics

The success of constructed models was measured as a combination of their accuracy, precision, and recall scores; confusion matrices and AUC-ROC scores will be provided where possible.

IX. Major results and Analysis

Logistic Regression.

The results for Logistic Regression are reported in the following **Tables 4**, **5**, and **6**:

Table 4. *Confusion matrix and performance metrics for logistic regression on original data*

```
Confusion Matrix
==========================================================
[[ 16   3]
 [  0 120]]

Classification Report
==========================================================
              precision    recall  f1-score   support

           0       1.00      0.84      0.91        19
           1       0.98      1.00      0.99       120

    accuracy                           0.98       139
   macro avg       0.99      0.92      0.95       139
weighted avg       0.98      0.98      0.98       139


AUC-ROC
==========================================================
0.9706140350877193
```

Table 5. *Confusion matrix and performance metrics for logistic regression on SMOTE data*

```
Confusion Matrix
==========================================================
[[107   7]
 [  4 112]]

Classification Report
==========================================================
              precision    recall  f1-score   support

           0       0.96      0.94      0.95       114
           1       0.94      0.97      0.95       116

    accuracy                           0.95       230
   macro avg       0.95      0.95      0.95       230
weighted avg       0.95      0.95      0.95       230


AUC-ROC
==========================================================
0.9816999395039323
```

Table 6. *Confusion matrix and performance metrics for logistic regression on ADASYN data*

```
Confusion Matrix
==========================================================
[[ 99   7]
 [  3 117]]

Classification Report
==========================================================
              precision    recall  f1-score   support

           0       0.97      0.93      0.95       106
           1       0.94      0.97      0.96       120

    accuracy                           0.96       226
   macro avg       0.96      0.95      0.96       226
weighted avg       0.96      0.96      0.96       226


AUC-ROC
==========================================================
0.9849056603773585
```

Analyzing the matrices and metrics above, we can see that SMOTE and ADASYN produce higher recall rates on the testing data in terms of identifying the minority class. ADASYN does perform marginally better than SMOTE in terms of weighted averages and AUC-ROC score. Meanwhile, the original data produces superior accuracy, but drastically under identifies the minority class relative to the resamples.

k-NN Method.

Testing and training accuracy for k-NN over increasing values of k can be found in **Tables 7**, **8**, and **9** below:

Table 7. *Testing accuracy and training accuracy of original data from k-NN*

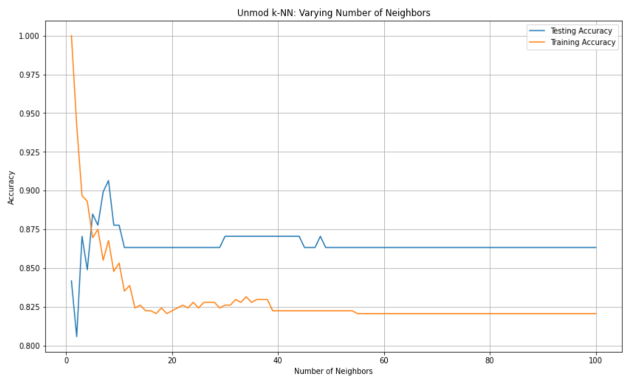
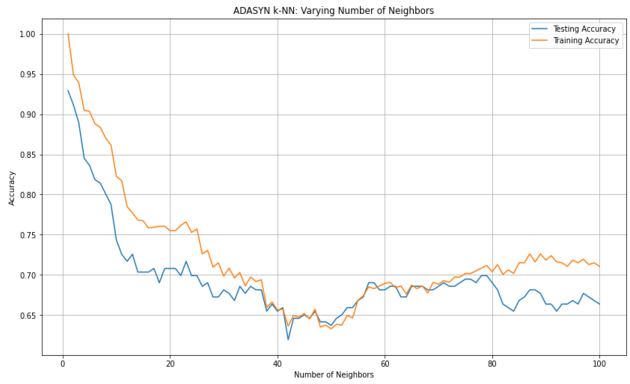


Table 8. *Testing and training accuracy of SMOTE resampled data from KNN*

Table 9. *Testing and training accuracy of ADASYN resampled data from k-NN*



From the increasing values of k plots above, we can derive an interesting result: the original unsampled data provides superior overall performance compared to the resampled variants. This may be a consequence of overfitting on the data in the resamples, which could exist as evidenced by consistently superior training performance relative to testing performance. It may also be an artifact of k-NN's characteristic of nonparametric operations: balancing the data is irrelevant to k-NN predictions. Regardless, the convergences that do occur within the resamplings are at significantly lower accuracies than in the original data.

Keras Neural Network.

For results of predictions from a fitted Keras neural network model, refer to **Table 10** below:

Table 10. *Loss and accuracy over validation and testing between baseline and resampled data*

Data	Loss	Acc.	Val. Loss	Val. Acc.
Original	0.1874	0.9295	0.2726	0.8962
SMOTE	0.1492	0.9462	0.1632	0.9373
ADASYN	0.1625	0.9321	0.1911	0.9195

Here, we can determine that SMOTE produces the most optimal predictions within the circumstances of Keras. However, it is important to note that all data variants report high-performing metrics — that is, save for an approximate 27.26% validation loss seen within a model fitted on the original data.

Baseline SVM.

Under default hyperparameters, performance metrics for predictions on a fitted non-grid search CV SVM can be found in the following **Table 11**:

Table 11. *Select performance metrics for SVM without grid search CV across all data variants*

Data	Accuracy	Recall
Original	0.8993	1.0000
SMOTE	0.9217	0.9397
ADASYN	0.9159	0.9250

Astonishingly, the original data under non-grid search SVM predictions produce a perfect recall rate. However, it does appear that this recall comes at a slight trade-off for general accuracy. As a further note, this result does promote some doubt about the validity of the non-grid search SVM; further investigation into this is required. While this paper will mark the original data as superior in this instance, it must be noted that a flaw in experimental design may have produced this result.

SVM with Grid Search CV.

For SVM with a grid search CV, test results and accompanying parameters are shown in **Tables 12**, **13**, and **14** below:

Table 12. *Precision, recall, f1-score, AUC-ROC score and confusion matrix of unsampled data from SVM with Grid Search*

```
SVC(C=100, gamma=0.1)
[[ 14   5]
 [  3 117]]
              precision    recall  f1-score   support

           0       0.82      0.74      0.78        19
           1       0.96      0.97      0.97       120

    accuracy                           0.94       139
   macro avg       0.89      0.86      0.87       139
weighted avg       0.94      0.94      0.94       139
```

Table 13. *Precision, recall, f1-score and confusion matrix of SMOTE resampled data from SVM with Grid Search*

```
SVC(C=100, gamma=0.1)
[[108   6]
 [  2 114]]
              precision    recall  f1-score   support

           0       0.98      0.95      0.96       114
           1       0.95      0.98      0.97       116

    accuracy                           0.97       230
   macro avg       0.97      0.97      0.97       230
weighted avg       0.97      0.97      0.97       230
```

Table 14. *Precision, recall, f1-score and confusion matrix of ADASYN resampled data from SVM with Grid Search*

```
SVC(C=1, gamma=1, kernel='poly')
[[102   4]
 [  7 113]]
              precision    recall  f1-score   support

           0       0.94      0.96      0.95       106
           1       0.97      0.94      0.95       120

    accuracy                           0.95       226
   macro avg       0.95      0.95      0.95       226
weighted avg       0.95      0.95      0.95       226
```

In this instance, the original data produces weaker predictions relative to the resamples. Between SMOTE and ADASYN, the results are again approximately identical with ADASYN again carrying marginally superior recall metrics.

X. Conclusion and future work

Initially, the desired product of these experiments was to produce a singular triumphal model. However, such a model was not so forthcoming. In reality, we have determined that nearly all of the models produced high-performance statistics — with the exception of k-NN, which is only relative as predictions on original data under k-NN still boast an approximate 88% accuracy. Overall, SMOTE and ADASYN resampled datasets generally produced more optimal predictions than the original. Between the two methods, results were approximately identical: superiority of one resampling over another occurred within extremely tight margins.

However, there was one notable exception to this: the non-grid search SVM and the Keras neural network. For the former, a model fitted on the original data produced a perfect recall rate. As mentioned previously, this is likely a result of experimentational error and requires further investigation. In general, implementing a logistic regression or grid-search SVM using SMOTE or ADASYN rebalanced data is likely to produce high-performance and reliable predictions.

XI. References

[1] A. A. Imran, M. N. Amin, M. R. Islam Rifat and S. Mehreen, "Deep Neural Network Approach for Predicting the Productivity of Garment Employees," *2019 6th International Conference on Control, Decision and Information Technologies (CoDIT)*, 2019, pp. 1402-1407, doi: 10.1109/CoDIT.2019.8820486.

[2] Haibo He, Yang Bai, E. A. Garcia and Shutao Li, "ADASYN: Adaptive synthetic sampling approach for imbalanced learning," *2008 IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence)*, 2008, pp. 1322-1328, doi: 10.1109/IJCNN.2008.4633969.

[3] L. Pelayo and S. Dick, "Applying Novel Resampling Strategies To Software Defect Prediction," *NAFIPS 2007 - 2007 Annual Meeting of the North American Fuzzy Information Processing Society*, 2007, pp. 69-72, doi: 10.1109/NAFIPS.2007.383813.

XII. Contributions

Data cleaning and transformation: *Siddharth Jaswal*, *Claire Zhang*

Logistic regression: *Claire Zhang*

k-NN method: *Siddharth Jaswal*, *Tanner Woods*

Keras neural network: *Tanner Woods*, *Claire Zhang*

SVM variants: *Siddharth Jaswal*, *Tanner Woods*

XIII. Appendix

Appendix 1. *Variables and Descriptions from the Dataset*

Variable	Dummies Created?	Variable Dropped?	Description
date	N	Y	MM/DD/YYYY of entry observation.
quarter	Y	N	Quarter of entry observation; five quarters observed.
department	Y	N	Associated department with the instance.
day	Y	N	Day of entry observation.
team	N	Y	Associated team number with the instance.
targeted_productivity	N	Y	Productivity quota set by management.
smv	N	N	Standard Minute Value, it is the allocated time for a task.
wip	N	N	Work in progress. Includes the number of unfinished items for products.
over_time	N	N	Represents the amount of overtime by each team in minutes.
incentive	N	N	Represents the amount of financial incentive in Bangladeshi Taka (BDT) that enables or motivates a particular course of action.
idle_time	N	N	The amount of time when the production was interrupted due to several reasons.
idle_men	N	N	The number of workers who were idle due to production interruption.
no_of_style_change	N	N	Number of changes in the style of a particular product.
no_of_workers	Y*	N	Numbers of workers assigned to the team.
actual_productivity	N	Y	Measure of real productivity at the end of shift.
productive	N	N	‘1’ indicates actual productivity of this team is equal or greater than targeted productivity; ‘0’ indicates actual productivity of this team is less than targeted productivity.