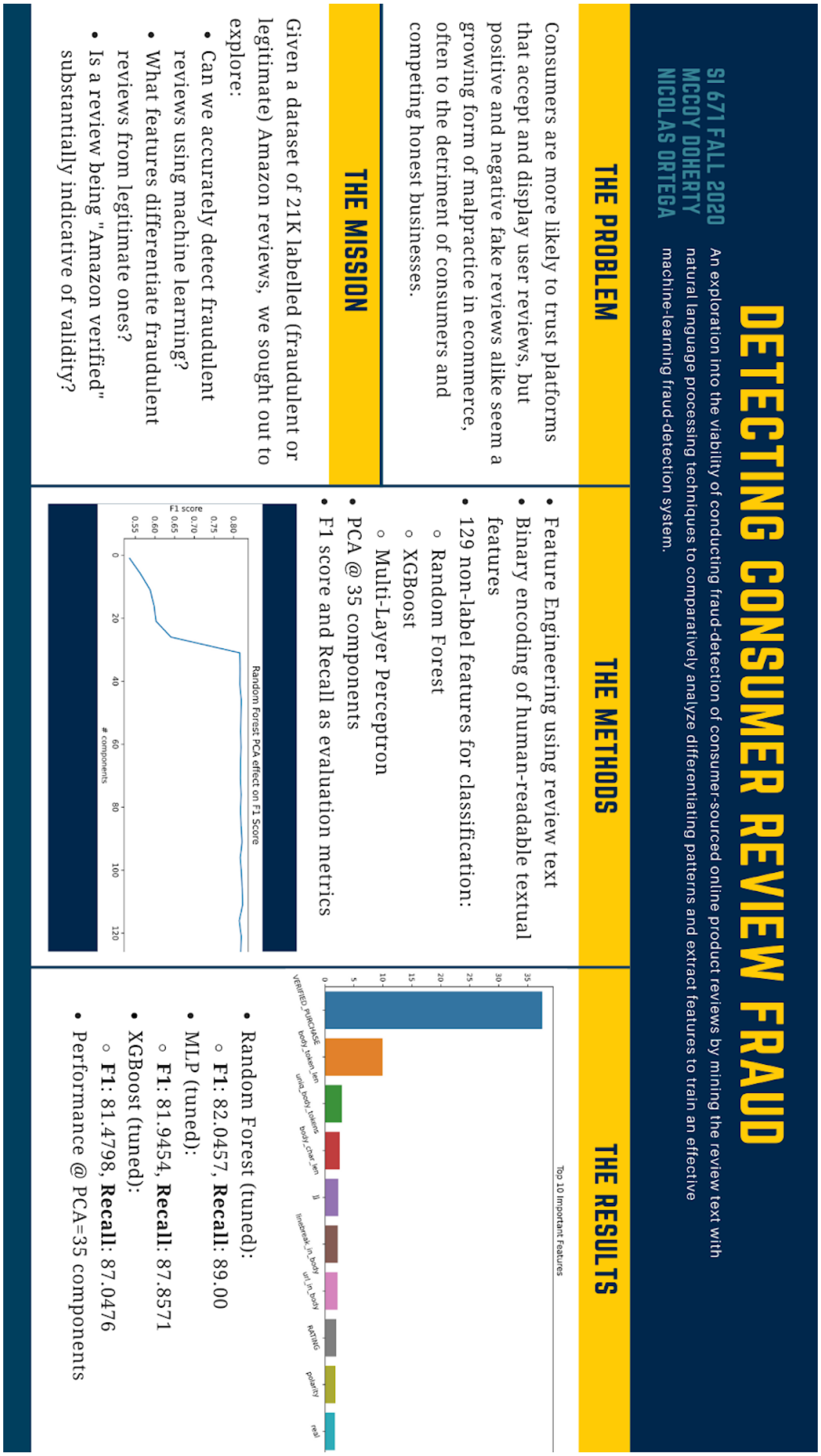
McCoy Doherty & Nicolas Ortega -- SI671 - Fall 2020 -- Consumer Review Fraud w/ Amazon



DETECTING CONSUMER REVIEW FRAUD

SI 671 FALL 2020
MCCOY DOHERTY
NICOLAS ORTEGA

An exploration into the viability of conducting fraud-detection of consumer-sourced online product reviews by mining the review text with natural language processing techniques to comparatively analyze differentiating patterns and extract features to train an effective machine-learning fraud-detection system.

THE PROBLEM

Consumers are more likely to trust platforms that accept and display user reviews, but positive and negative fake reviews alike seem a growing form of malpractice in ecommerce, often to the detriment of consumers and competing honest businesses.

THE MISSION

Given a dataset of 21K labelled (fraudulent or legitimate) Amazon reviews, we sought out to explore:

- Can we accurately detect fraudulent reviews using machine learning?
- What features differentiate fraudulent reviews from legitimate ones?
- Is a review being "Amazon verified" substantially indicative of validity?

THE METHODS

- Feature Engineering using review text
- Binary encoding of human-readable textual features
- 129 non-label features for classification:
  - Random Forest
  - XGBoost
  - Multi-Layer Perceptron
- PCA @ 35 components
- F1 score and Recall as evaluation metrics

THE RESULTS

- Random Forest (tuned):
  - **F1**: 82.0457, **Recall**: 89.00
- MLP (tuned):
  - **F1**: 81.9454, **Recall**: 87.8571
- XGBoost (tuned):
  - **F1**: 81.4798, **Recall**: 87.0476
- Performance @ PCA=35 components

To view more directly, here's a URL -- no "great" way to embed a poster in a page like this :
https://docs.google.com/presentation/d/15SbScFwRgQhknB-s17gJ2U0PJ9UUWGwJKrWHrbvCek0/edit?usp=sharing

McCoy Doherty
Nicolas Ortega
SI671 - Fall 2020

Consumer-Review Fraud Detection on Amazon

On Origination, Motivation & Intention:

This project began as a conversation about where the authors wished we could mine every-day data for meaning, hoping we could identify hidden signals in data that could somehow actionably be interpreted in a way that could create a genuine positive impact in peoples' lives. In light of the COVID-19 epidemic still ravaging the nation and, consequently, our increased personal usage of e-commerce titan Amazon's services to minimize our risk of exposure while shopping, we mutually and informally observed an increase of reviews that "feel off" and reviews that levied accusations that the product seller had purchased fake reviews. Fake reviews seek to deceive consumers, damage the reputations of honest brands, and fraudulently establish consumer acclaim for brands and products undeserving, negatively impacting consumer trust and their potential return-on-investment if inflected towards a lesser product fraudulently.

This led us to consider ways in which we could attempt to apply data mining and machine learning techniques we had learned throughout the semester to attempt to gauge if there were signals that could be detected in the composition of consumer review text to meaningfully discriminate between legitimate and fraudulent reviews. We were especially interested in working with data sourced from Amazon due to its inarguable market dominance in both e-commerce and modern commerce in general, a titan that would only become increasingly ubiquitous given the plague of circumstances.

Driving motivations for this project were born of the realization that, given reasonable success of detecting fraudulent reviews accurately, such a process could be utilized as a back-end system for an eventual consumer-facing application to assist modern consumers with identifying when reviews from "everyday users" they may be inclined to trust could actually be deceitfully intended. This work, if successful, would empower everyday consumers and bolster confidence in making more thoroughly-informed purchasing decisions, able to incorporate not only the reviews presented, but if there seems an apparent dishonestly veiled therein.

With the ever-growing ubiquity of consumer-sourced reviews in consumer-facing transactive platforms, we also believe work done in this project could be indirectly applicable in other general sales platforms (such as eBay) or curtailed further to identifying more domain-specific signals of legitimacy or lack thereof on platforms such as GrubHub for food-services or AirBNB for lodging services.

Discussing the Dataset:

The dataset we are using is one initially curated by Amazon itself, containing a collection of reviews for a relatively diverse set of products across various prevalent categories. The dataset consists of 21,000 reviews in total, each with data expressing a binary label of legitimate or illegitimate, the number of "stars" granted by the reviewer, the review title, the body text of the review, as well as data about the product itself, including product title and category. Out of an acute familiarity with the data requisites involved in fraud detection in other domains of service, it is also worth explicitly noting that our dataset does not contain any data about the user accounts of those posting these reviews, nor any data about the user sessions involved in the review origination process.

In addition to being balanced with equal counts of legitimate and fraudulent review in total, the proportions of legitimacy to fraudulency were balanced across all 700 reviews provided for all 30 shopping categories in the dataset, effectively translating to 350 instances of valid and invalid reviews per category.

Defining the Objectives:

Our primary objectives for the scope of the project were to

- develop a classifier that utilizes machine learning methods to accurately discriminate between fraudulent and legitimate Amazon reviews for various categories of products
- utilize natural language processing to engineer features that such a model could utilize to identify meaningful decision boundaries for its classifications

Vital Considerations on Metrics:

As our most central goal for the project was the training of a machine learning model, we had to consider which metrics would be of primary focus for setting the direction towards which our models would be continually optimized and, after much consideration, decided that those most effective for this project would likely be recall and F1 scores.

Retail platform providers, such as Amazon in our case, rely on consumer reviews largely being an altruistic effort of complementary service in rendering their reviews for the consideration of others, an effort that can increase consumer confidence in a potential transaction and help garner further sales. Depending on the direct consequence of a model labelling a review as fraudulent, using recall, a metric that optimizes finding the largest proportion of fraudulent reviews at the potential cost of false-positives, could result in legitimate reviews accidentally being penalized. If users were made aware of this, it could offend them and harm the reviewing ecosystem.

Out of interest of rendering more generalizable findings, our ultimate decision was to focus on utilizing F1 scores to guide our approach to tuning and optimizing our models, as they would represent the harmonic mean of precision and recall, and thus overall better represent a more balanced form of performance that could later be further curated towards precision or recall depending on the interest of a given application.

Feature Engineering // Methods:

After reading our dataset into a Pandas dataframe and converting boolean-like variables into 0/1 expressions for compatibility with the SKLearn library, we turned our attention to feature engineering the review-body text. Initial efforts focused on basic metrics like review character-length, word-length, stopword-free word length, and unique-word count. After considering bare-eye observations of visible irregularities in writing style, we additionally opted to build binary features around reviews being entirely in caps and being entirely in lower-case, as well as count-based features encapsulating total proportionality of review letters that are capitalized and what proportion of terms other than “I” are written in “title case,” having a capitalized initiating letter.

Other explicitly human-readable textual features we detected and chose to constitute binary features from were the occurrence of line-breaks (manifested as “<br />”), potentially indicative of a reviewer putting in more thorough effort we suspect unlikely of fake reviews, whether or not a review contained a URL, potentially indicative of intent to fish consumer traffic elsewhere, and whether or not a review contained what appeared to be a sort of Amazon-specific tagging syntax indicating either an embedding of a video to the review or, otherwise, an ASIN (Amazon Standard Identification Number) that appeared to be used for cross-site linkage to other products on the Amazon platform.

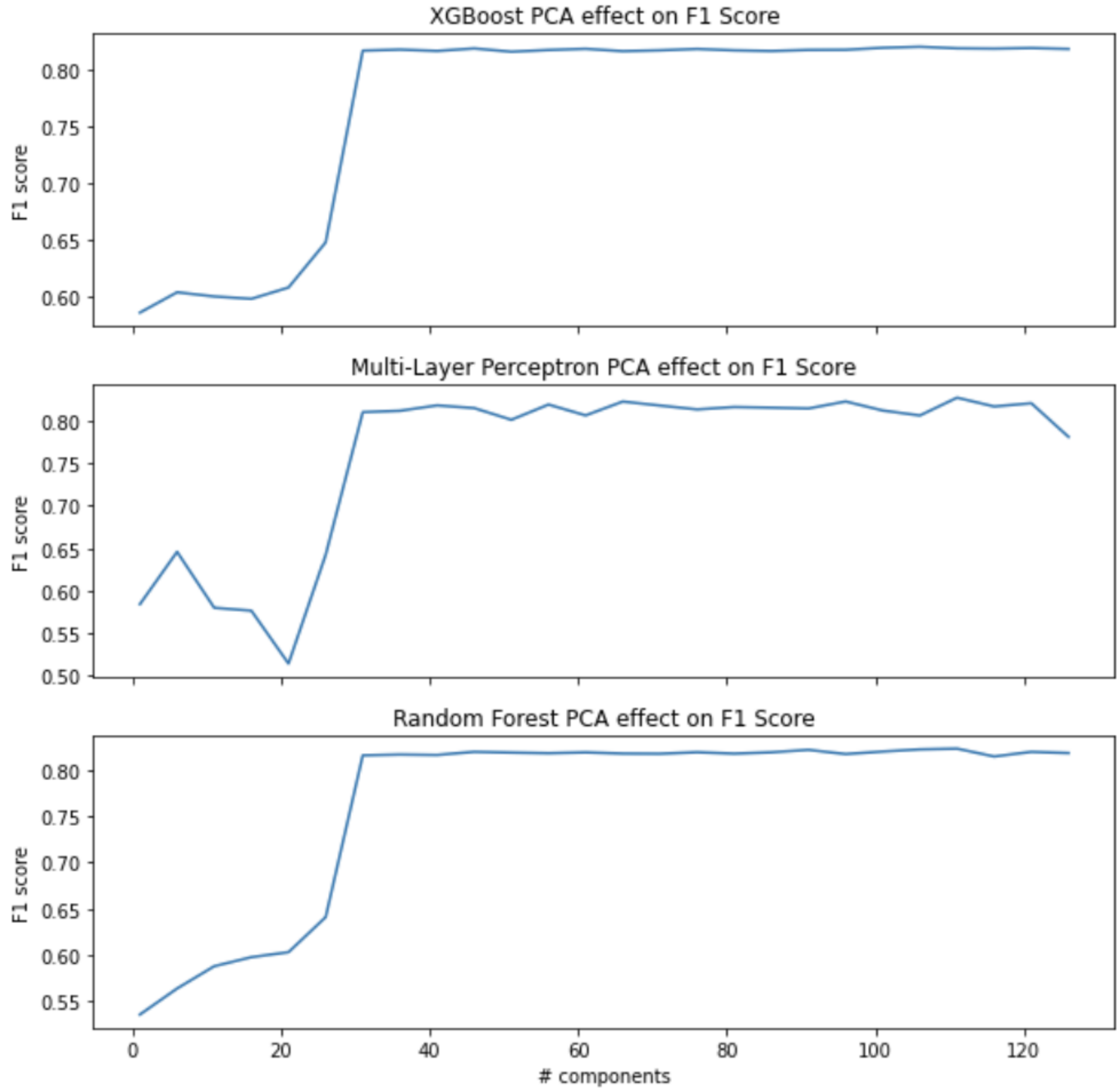
Our final set of features were derived from using the TextBlob library’s sentiment analysis to establish numerical representations of polarity and subjectivity, using NLTK’s Part of Speech tagger on the full review strings to get frequency of occurrence for each type of English speech component, and binary indicators of whether or not the review contained specific terms we were curious about being potentially being used in intentionally-persuasive fraudulent language. These were most often selected to observe how reviewers describe the items in terms of quality or marketing terms (e.g. “great organic lotion, but too heavy”), how they relate them to other individuals in their lives (e.g. “His mother loved this gift”), and whether or not certain other terms of persuasive rhetoric were utilized to attempt to establish appeals through patriotism, religiosity, or political stance. Most of these terms are indicated in the table on the following page.

Binary "Contains" Features A reference table for many keywords we noted the binary state of being in a given review text or not. Additional unlisted variations of terms below may have also been used.					
Family Role & Age & Professional Expressions					
Family	**Ext. Family**	**Romantic**	**Age State (I)**	**Age State (II)**	**Profession**
Mother Father Daughter Son Brother Sister	Aunt Uncle Niece Nephew Grandma Grandpa	Boyfriend Girlfriend Husband Wife Fiance Partner	Baby Infant Kid Young Toddler	Young Teen Adult Parent Elderly	Army/Military Teacher Professor Doctor Lawyer Office Business
Quality Descriptors					
Class	**Tactile:**	**Marketing**	**Validity**	**Durability**	**Generic**
Cheap Affordable Poor Rich Luxury	Light Heavy Smeel Feel	Natural Organic Authentic Synthetic Real	Real Fake False	Broke Faulty Busted Durable	Good Bad
Referentials & "Other"					
Referentials:	He/Him/His	She/Her/Hers	I/my/we/our	you/your	they/their
Other:	Church, God, Faith, Trust, MAGA, Patriot, America, China, Chinese				

After all features were built, our data-frame contained 129 non-label features that were run through assorted "dummy" (static-strategy) classifiers and default-parameter instantiations of SKLearn's Random Forest, K-Nearest Neighbors, Logistic Regression, Linear SVC, XGB, SGD, GaussianNB, and MLP classifiers, the most effectively-performant of which were evaluated under principal-component transformations and grid searches, as discussed in the following section.

Performance Evaluation:

To establish a baseline for performance, all static strategies available as part of SKLearn's "dummy" classifiers were tested against the dataset, the most strongly performant of which was the stratified strategy, which exhibited 0.5038 recall and 0.5048 F1 scores. All models were then tested under their default parameter configurations. Of the eight models tested, only three achieved F1 scores greater than 0.80, those models being Random Forest, XGBoost, and Multilayer Perceptron, scoring 0.8226, 0.8242, and 0.8175, respectively. These three models were then explored further for how their performance would respond with the dataset undergoing dimensionality reduction in the form of projection into principal components.



We observed that we could reduce dimensionality to approximately thirty-five principal components with relatively little negative impact to the model outcomes and significantly decreased runtime requirements to train our models. Average loss when contracting the dataset into thirty-five components was approximately 1% for Random Forest models and 1.5%-2% for Multilayer Perceptron and XGBoost models.

The thirty-five-dimensional translation of our dataset was then utilized to grid-search these three models to attempt to find a more optimal combination of parameters for each, compared to the default parameters, and ultimately identify the best-performing model with respect to F1 score.

Results:

After having run the grid searches for each model, the strongest model performance was from a Random Forest model with criterion set to “gini,” maximum depth set to None, maximum features set to “auto,” and 250 estimators. This model demonstrated an F1 score of 0.82 and a recall score of 0.89, leading performance in both of these metrics of interest. While the model had very comparable F1 performance prior to undergoing dimensionality reduction, a loss offset by the gridsearch-tuning efforts, the pre-PCA recall score of the model was 0.8786, meaning that in addition to achieving near-equivalent F1 performance, even after PCA we had increased model recall by a full percentage point.

Both the XGBoost and MLP models performed within 1% of one another in both F1 and recall scoring, the XGBoost with F1 and recall of 0.8705 and 0.8148 respectively and the MLP model with 0.8786 and 0.8195 F1 and recall scores, respectively. It is worth noting that grid-search tuning of the XGBoost model offset the F1 and recall performance losses resulting from dimensionality reduction quite substantially, albeit not completely. The Multilayer Perceptron model, however, appeared to demonstrate ability for equivalent-or-better F1 scoring when properly tuned, despite dimensionality reduction, although to the net-loss of some recall that could potentially be recovered given a grid-search tuned to optimize that measure.

Future Work:

Given more time to continue building forward on our work, there are a few different avenues through-which we believe there is potential for improved performance. These future engagements would focus on expanding features, expanding into deep learning, and potentially even expanding the dataset.

While our work utilizes machine learning methods with SciKit Learn to train our model, we believe that there is room to also explore utilizing neural networks and deep learning methods to explore the effectiveness of unsupervised learning approaches in this space, but elected to leave that out of the scope of the current implementation due to time constraints and relative inexperience in utilizing neural networks.

In terms of feature expansion, we believe it could have been worth also mining for meaningful features in the headers of reviews, although they are brief enough for mutual skepticism of likelihood of there being much value there. More importantly, however, we believe we could also consider pulling in strategies from our information retrieval course to consider utilizing term-frequency-inverse-document-frequency and cosine similarity in further work.

Regarding the notion of expanding the data, one of our members coincidentally encountered another academic yesterday who reported having a similar dataset of 50,000 records rather than 21,000 records and, pending response, agreed to share their dataset with our team as well. It's said more data can trump better algorithms and, in this case, we believe we could see gains in both model accuracy and generalizability given a larger sample of training data. Given more data per category, we would also be interested in mining for indicators potentially more specific to particular categories of consumer retail, such as health products, electronics, and clothing.

Last but not least, knowing that platforms have greater breadth of data access when internally researching to construct fraud-detection systems, namely session data and account data surrounding the posting of the review, we believe that if deploying a consumer-insights application that takes in a product-page URL and runs review fraud detection, it would be likely ideal to additionally scan the review-author account pages. While we wouldn't gain session data to mine, this would allow us to mine their review history for features integrating, what their average ratings are like, and how concentrated their reviews are into specific categories or product ranges ("is this person reviewing a suspicious amount of blenders?"), sentiments, and so on.

<u>Conclusion:</u>

We started this project with the intention of finding out the extent to which detecting fraud in consumer crowd-sourced reviews can be viable through the features of the review alone and ultimately developed models capable of achieving stable F1 scores of approximately eighty percent, as well as recall scores at ninety percent, a testament to how much can be done without adjoining user-session data or account data at the disposal of e-commerce platforms have available when attempting to develop these systems internally. Through this project, we have effectively reaffirmed the viability of our interest in developing a strictly-to-help-consumers analytics application for detecting fraudulent reviews with a platform-external team, especially with our willingness to consider optimizing instead towards recall due to relatively lower stakes involved if accidentally false-positive labelling a review as fraudulent.

Ultimately, however, I predict that in coming decades, the sustained effort of businesses to find ways to render machine and deep learning more accessible to the untrained consumer may result in the proliferation of neural-network-based text generation into the world of review fraud and usher in an era of a small group of service providers emerging who abuse deep learning to offer clients bulk amounts of fake reviews that appear more organic, expressive, and human than ever before, and thus more convincing to humans and, potentially, even current fraud-detection models.

Appendix:

Because we made lots of spreadsheets while doing this project to be able to document and easily compare circumstantial model performance, and we may as well share them:

Model Performance -- Standard Dataset -- Default Model Parameters					
Classifier Model	**Accuracy**	**Precision**	**ROC AUC**	**Recall**	**F1 Score**
Dummy Stratified	50.57%	50.57%	50.57%	**50.38%**	**50.48%**
Random Forest	81.05%	77.33%	81.05%	**87.86%**	**82.26%**
KNeighbors	59.38%	60.03%	59.38%	56.14%	58.02%
Logistic Regression	78.60%	75.98%	78.60%	83.62%	79.62%
LinearSVC	71.52%	79.05%	71.52%	58.57%	67.29%
XGBoost	81.07%	76.95%	81.07%	**88.71%**	**82.42%**
MLP	79.88%	74.81%	79.88%	**90.10%**	**81.75%**
Gaussian NB	55.17%	67.82%	55.17%	19.67%	30.49%
SGD	50.90%	88.00%	50.90%	2.10%	4.09%

Model Performance -- 35 Principal Components -- Default Model Parameters					
Classifier Model	**Accuracy**	**Precision**	**ROC AUC**	**Recall**	**F1 Score**
Random Forest	79.52%	75.12%	79.52%	**88.29%**	**81.17%**
XGBoost	79.05%	75.08%	79.05%	**86.95%**	**80.58%**
MLP	76.17%	77.94%	76.17%	73.00%	75.39%

Model Performance -- 35 Principal Components -- GridSearch-Tuned Model Parameters						
Classifier Model	**Accuracy**	**Precision**	**ROC AUC**	**Recall**	**F1**	**Best Parameters**
Random Forest	80.52%	76.10%	80.52%	**89.00%**	**82.05%**	*{'criterion': 'gini', 'max_depth': None, 'max_features': 'auto', 'n_estimators': 250}*
XGBoost	80.21%	76.58%	80.21%	87.05%	81.48%	*{'learning_rate': 0.1, 'max_depth': 5, 'min_child_weight': 5, 'n_estimators': 100, 'objective': 'binary:logistic'}*
MLP	80.64%	76.78%	80.64%	87.86%	81.95%	{'activation': 'relu', 'alpha': 0.0001, 'learning_rate': 'adaptive', 'solver': 'adam'}