Annual Wage Prediction: Machine Learning Competition

Dhananjay Ghei*, Sang Min Lee†

May 20, 2020

Abstract

We use recent machine learning tools to predict annual wages in the Current Population Survey using a small subset of variables and an expanded set of variables. We deploy three recently developed machine learning techniques – XGBoost (Extreme Gradient Boosting), Light GBM (Light Gradient Boosting Machine), and Deep Learning. To achieve better prediction accuracy, we employ stacking to combine these models. For the small subset of variables, our best models show an improvement of 0.632%, 0.749%, and 3.803% in RMSE-log, RMSE-level, and absolute deviation, respectively, over the baseline Mincer regression. With the expanded set of variables, our best models show an improvement of 49.471%, 47.594%, 27.442% in RMSE-log, RMSE-level, and absolute deviation, respectively.

*Department of Economics, University of Minnesota
†Department of Economics, University of Minnesota

1 Introduction

Can machine learning algorithms outperform Mincer wage regression? This note explores this idea using state-of-the-art machine learning algorithms on the Current Population Survey dataset. We use three recently developed machine learning techniques – XGBoost, Light GBM, and Deep Learning and combine them using Stacking. We assess the performance of the machine learning models with the baseline Mincer regression using three different measures:

1. RMSE - log
2. RMSE - level
3. Absolute Deviation

Mincer regressions are OLS regressions of the logarithm of wages on individual characteristics such as age, education, gender, race, etc. Our baseline Mincer regression takes the following form:

$$y_i = \beta_0 + \beta_1 d_{\text{age},i} + \beta_2 d_{\text{educ},i} + \beta_3 d_{\text{age},i} \times d_{\text{educ},i} + \beta_4 d_{\text{gender},i} + \beta_5 d_{\text{state},i} + \beta_6 d_{\text{race},i} + \varepsilon_i \tag{1}$$

where, y_i denotes the logarithm of the wages for individual i, $d_{j,i}$ is a dummy variable of characteristic j for individual i.

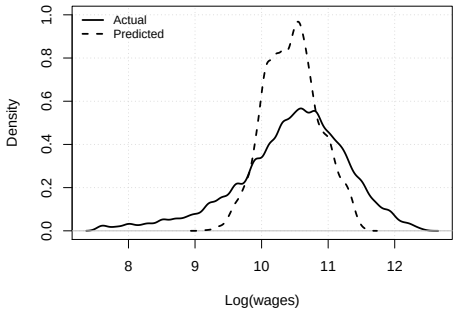
While the OLS estimates give a first pass at understanding the prediction of wages, these models typically have a poor predictive power. The low predictive power is visible in Figure 1. The solid line shows the actual data, and the dashed line shows the Mincer regression's prediction. The left column is on the training data, and the right column is on the test data. Two things strikeout: first, the Mincer regression over-predicts the density around the mean, and second, it does not capture the tails.

To preview the results in this regard, our best-performing Stacking model overcomes these two issues of Mincer regression, as in Figure 2.

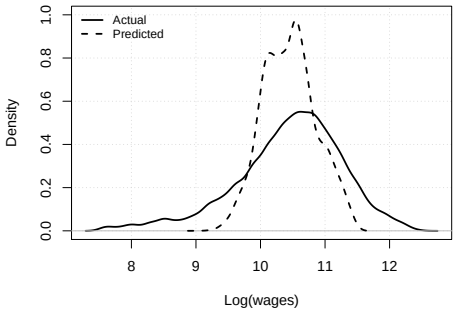
Figure 1 Baseline predictions from the Mincer regression

The figure shows two plots comparing the densities for the logarithm of wages of the training and the test dataset against their predictions from the Mincer regression. The solid black line shows the actual wages and the dashed line shows the predicted wages from the Mincer regressions

(a) Training Data



(b) Test Data



2 Part I

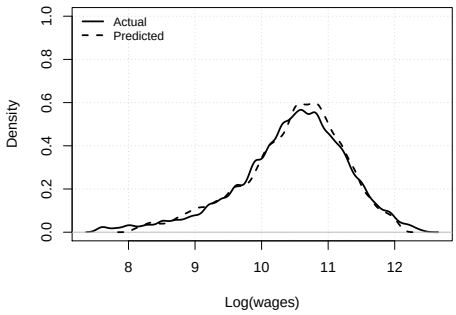
2.1 Machine Learning Algorithms

For this competition we utilized three algorithms: XGBoost (Extreme Gradient Boosting), LightGBM (Light Gradient Boosting Machine), and Deep Learning (multi-layer feedforward artificial neural network).

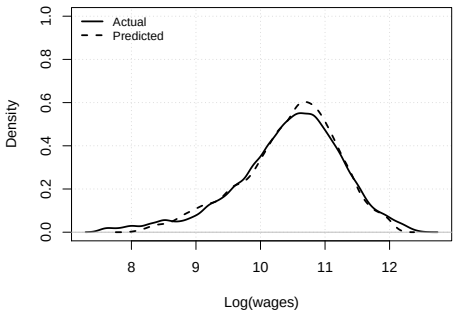
Figure 2 Baseline predictions from the Stacking (Part II)

The figure shows two plots comparing the densities for the logarithm of wages of the training and the test dataset against their predictions from Stacking on Part II. The solid black line shows the actual wages and the dashed line shows the predicted wages.

(a) Training Data



(b) Test Data



XGBoost (Chen and Guestrin, 2016) and LightGBM (Ke et al., 2017) are both tree-based gradient boosting algorithms which are known for their speed and prediction performance on structured data, winning many Kaggle competitions. Therefore, they were our natural choice for this competition, given the tabular nature of CPS data. The key difference between the two algorithms lies in how they grow trees. XGBoost grows tree depth-wise, whereas LightGBM does it leaf-wise.

Deep Learning (LeCun et al., 2015) is the other popular strand of ML tools. It also performs well for tabular data with fine-tuned neural network structure. Since there is an ongoing debate in the machine learning community over whether gradient boosting performs better than Deep Learning, we decided to include both for our prediction.

All key hyper-parameters, such as the number of trees for XGBoost and LightGBM, and the structure of hidden layers of Deep Learning, are tuned through cross-validation. We utilized the following random discrete grid search process:

1. Given a grid of hyper-parameters, choose one grid point randomly, estimate the model, and calculate the 5-fold cross-validation errors.

2. Repeat 1, until maximum training time of three hours is reached or the next ten models don't improve log-RMSE by 0.01.

3. Choose the grid point with the lowest log-RMSE.

The last row of Table 1 and 2 reports the number of grid points, or equivalently models, that the machine searched before reaching Step 3.

Furthermore, to check whether the ensemble of these three models could perform better, we used Stacking (Wolpert, 1992). Unlike Boosting, which combines weak learners of the same type to produce a strong learner, Stacking is a technique merging different types of strong learners to obtain better performance. Stacking uses various algorithms like GLM (Generalized Linear Methods) as a meta-learner to integrate the base models.

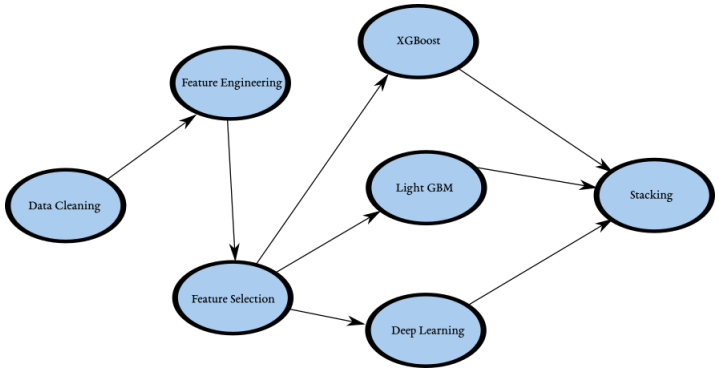
We chose one, three, or five models from each of XGBoost, LightGBM, and Deep Learning. Then, we stacked them through 6 different methods: GLM, GLM with nonnegative weights, GBM (Gradient Boosting Machine), DRF (Distributed Random Forest), Deep Learning, and XGBoost. This means that Stacking with 18 different specifications was conducted.

For brevity, we report the best performing stacked model in our report. For Part I, it is GLM Stacking with five models from each method, and for Part II, it is GBM Stacking of five best models from each of

three methods.

Figure 3 Schematic overview of our workflow

The diagram shows an overview of our workflow for the machine learning competition.



2.2 Other algorithms

We implemented more traditional machine learning algorithms as well for this exercise to see if they outperformed the above models. In particular, we implemented Partial Least Squares, Non-parametric regressions, Random Forests, Neural Networks, Principal Component Regression, Generalized Additive Models, Quantile Regression, Splines, B-Splines, Isotonic regression, etc. Hyper-parameters were tuned for each of these models (when required) using the appropriate methods.

Our results for each of these models suggest that they did not perform well over the Mincer regression for the test dataset. To obtain any gain in prediction performance with these limited variables was a difficult task. The results for these methods are, therefore, omitted from our analysis.

3 Part II

3.1 Data merging and cleaning

We download the March Current Population Survey, 2012, ASEC supplement from IPUMS (Flood et al., 2020) and merge it with the training and test dataset provided to add additional variables.

Next, we remove the variables that are not allowed in the procedure as follows: first, we remove the INCOME and TAX modules as mentioned in the rules, second, we remove all the variables in the WORK module with the exception of occupation and industry of employment, third, we remove all child support variables, fourth, we remove the SPM module except for the variables mentioned in the last email about the competition, and finally, we remove variables in other modules that are related to hours worked, annual earnings, hourly wages, and subcomponent of earnings. We follow a conservative approach in eliminating the variables. If we are not sure of any variable, we remove it. Table 4 shows the complete list of 140 CPS variables that we included in our feature engineering, feature selection, and model building process.

As the industry and occupation classification in the original dataset is too narrow, we reclassify occupation and industry of employment more broadly from the CPS classification. The broad classification can be accessed here and here.

3.2 Feature engineering

Feature engineering is an important part of machine learning that makes a huge difference in model's performance. Before running the models, we construct additional features in the dataset using the domain

knowledge of the data. We construct these summary features on the training dataset and input those features in the test dataset without modifying or touching the test dataset. Here, we follow these steps to create additional features:

1. We first take the health variables and construct their principal component on the training dataset. These variables are DIFFHEAR, DIFFEYE, DIFFREM, DIFFPHYS, DIFFMOB, DIFFCARE, DIFFANY, and DISABWRK.

2. Next, we create averages of continuous variables by broad occupation, broad industry, metro area, state, and combinations of them. This captures the heterogeneity in these variables across occupations, industries, cities, states, and their combination.

3. We also create averages of continuous variables by education and race. This captures heterogeneity in these variables across education levels and race.

4. Subsequently, we create additional variable on the number of subsidies received. We calculate the total number of subsidies received by the individual from HEATSUB, FOODSTMP, GOTWIC, LUNCHSUB, and RENTSUB. These are indicator variables and do not capture any information related to the value of subsidies. Our understanding of the email was that we cannot use the value of subsidies and therefore, we dropped them from our analysis. We only use these binary variables for our analysis.

5. Also, we create variables to identify mobility of individuals' parents. We construct dummy variables to see if the person was born in the same place as their father, mother, or both. This captures the mobility of the parents over their life-cycle and can have strong correlations with the wage of the person.

6. Finally, we use one hot encoding for all the dummy variables. This is done in order to be able to run feature selection models on dataset that contains dummy variables.

This gives us a total of 217 variables (2088 one-hot encoded variables).

3.3 Feature selection

The final step before running machine learning algorithms is to select features of the dataset. As we have more than 2000 variables, dumping all of them in to the model will increase the run time of the training algorithm significantly. Also, it is also not clear how these variables interact and hence, putting all the features could lead to either increased complexity of the model or over-fitting.

To overcome this problem, we use LASSO (Tibshirani, 1996) with 10-fold cross-validation on the training dataset. If LASSO selects any of the one-hot encoded variables, we select the entire variable from the dataset. For example, if LASSO selects broad occupation category "Agriculture, Forestry, and Fisheries" then, we select all the broad occupation categories for our model.

This gives us a total of 92 variables (1889 one-hot encoded variables).

3.4 Machine learning

With the expanded set of variables, we now train the algorithms in a similar way as in Part I. That is, we use XGBoost, Light GBM, and Deep Learning, hyper-tune our parameters using grid search, and conduct Stacking on the best models.

4 Results

Table 1 shows the results for Part I of the competition. The first column shows the performance measures for the Mincer regression which is our baseline model. The remaining columns show the performance measures and percentage reduction relative to the Mincer regression for XGBoost, Light GBM, Deep Learning, and stacking, respectively. The best out of sample prediction on the test dataset for RMSE-log and RMSE-level is achieved by stacking. The best out of sample prediction for absolute deviation is achieved by Deep Learning. Overall, we achieve the highest reduction of 0.632% in RMSE-log, 0.749% in RMSE-level, and 3.803% in Abs-Dev measure across all these models.

Table 1 Performance outcomes for Part I

This table shows the performance of our machine learning models against the baseline Mincer regression. The first column of the table shows the performance measures for the Mincer regression, the next four columns show the performance measures and percent reduction against the baseline model for XGBoost, Light GBM, Deep learning, and stacking, respectively. The last row shows the number of models trained for hyperparameter tuning.

	Measure	Baseline (Mincer)	XGBoost	Light GBM	Deep Learning	Stacking
Training	RMS Log	0.7164	0.7120	0.7116	0.7135	0.7108
	% reduction		0.610	0.664	0.403	0.7812
	RMS Level	30874.3054	30755.5700	30673.3300	31681.9900	30649.0100
	% reduction		0.385	0.651	-2.616	0.7297
	Abs Dev	3.7409	3.7634	3.7928	3.6622	3.7627
	% reduction		-0.602	-1.390	2.104	-0.583
Test	RMS Log	0.7200	0.7160	0.7164	0.7208	**0.7154**
	% reduction		0.545	0.491	-0.121	**0.6321**
	RMS Level	30762.6019	30608.0100	30565.3000	31562.3200	**30532.2400**
	% reduction		0.503	0.641	-2.600	**0.7488**
	Abs Dev	3.7670	3.6755	3.6856	**3.6238**	3.7096
	% reduction		2.429	2.162	**3.803**	1.525
Number of Models		1	78	58	34	

Table 2 shows the results for Part II of the competition with the expanded set of variables. As for the RMSE criteria, Stacking showed the best performance, reducing RMSE-log by 49.471%, RMSE-level by 47.594%. For Abs-Dev, Light GBM provided the most performance gain of 27.442%.

We think Stacking is performing the best, because it combines algorithms that are capturing different features of the data. For example, in Part II, None of the five most important variables for XGBoost and LightGBM were in the top five for Deep Learning, as you can see in Table 3. We conjecture that Deep Learning is complementing XGBoost and LightGBM by explaining the residuals of the two boosting algorithms.

5 Replication files and packages

The code for replication of the results is in the GitHub repository. Note that this is a private repository but we have already provided you with access to the repository. Go to https://github.umn.edu and log in using your UMN ID and password. You will see a repository called `ghei0004/mebdi-ml`. It contains all the codes along with the README files for replication.

Table 2 Performance Outcomes for Part II

This table shows the performance of our machine learning models against the baseline Mincer regression. The first column of the table shows the performance measures for the Mincer regression, the next four columns show the performance measures and percent reduction against the baseline model for XGBoost, Light GBM, Deep learning, and stacking, respectively. The last row shows the number of models trained for hyperparameter tuning.

	Measure	Baseline (Mincer)	XGBoost	Light GBM	Deep Learning	Stacking
Training	RMS Log	0.7164	0.2951	0.2323	0.3132	0.2477
	% reduction		58.801	67.574	56.275	65.428
	RMS Level	30874.3054	15216.9631	12768.0100	15300.9400	12974.2600
	% reduction		50.713	58.645	50.441	57.977
	Abs Dev	3.7409	2.6121	1.9322	4.0592	2.1711
	% reduction		30.174	48.350	-8.510	41.963
Test	RMS Log	0.7200	0.3742	0.3728	0.4214	**0.3638**
	% reduction		48.027	48.224	41.463	**49.471**
	RMS Level	30762.6019	16800.8900	16579.8428	18121.2300	**16121.5972**
	% reduction		45.385	46.104	41.093	**47.594**
	Abs Dev	3.7670	2.8089	**2.7333**	3.3090	2.8379
	% reduction		25.435	**27.442**	12.160	24.664
Number of Models		1	33	14	37	

Table 3 Variable Importance

XGBoost	
1	Employer Contribution to Insurance (EMCONTRB)
2	Mean Employer Contribtuion to Insurance by Occupation (O1)
3	Indicator if a person is 150% and above the low income level (POVERTY.23)
4	Indicator if a person is a full-time worker (FULLPART.1)
5	Indicator if a person is a part-time worker (FULLPART.2)
LightGBM	
1	Employer Contribution to Insurance (EMCONTRB)
2	Mean Employer Contribtuion to Insurance by Occupation (O1)
3	Indicator if a person is 150% and above the low income level (POVERTY.23)
4	Indicator if a person is a full-time worker (FULLPART.1)
5	Indicator if a person is included in pension plan at work (PENSION.3)
Deep Learning	
1	Indicator if a person did not receive a food stamp (FOODSTMP.1)
2	Indicator if a person did receive a food stamp (FOODSTMP.2)
3	Indicator if residency of one year ago is Oregon (MIGSTA1.41)
4	Indicator if a person is living in East South Central Division (REGION.32)
5	Broad Industry is Retail (INDBROADLY.Retail)

The table shows the list of variables used in our analysis. The first column shows the variable name in the CPS, second column shows if the variable is a continuous variable or a dummy variables, and the last column gives the description.

Table 4: Variables used for analysis

Variables	**Continuous**	**Description**
wageinc	Y	Wage and salary income (LHS variable)
age	Y	Age
educ_cat		Educational attainment (recoded by MEBDI)
male		Male
statefip		State (FIPS code)
wtsupp	Y	Survey weight
asecwth	Y	Annual Social and Economic Supplement Household weight
region		Region and division
metro		Metropolitan central city status
metarea		Metropolitan area
county		FIPS county code
cbsasz		Core-based statistical area size
individcc		Individual principal city
ownershp		Ownership of dwelling
pubhous		Living in public housing
rentsub		Paying lower rent due to government subsidy
heatsub		Received energy subsidy
foodstmp		Food stamp recipiency
stampno	Y	Number of persons covered by food stamps
stampmo	Y	Number of months received food stamps
atelunch	Y	Number of children who ate complete school lunch
lunchsub		Government school lunch food subsidy
frelunch	Y	Number of children with government school lunch subsidy
unitsstr	Y	Units in structure
phone		Telephone availability
nfams	Y	Number of families in household
ncouples	Y	Number of married couples in household
nmothers	Y	Number of mothers in household
nfathers	Y	Number of fathers in household
relate		Relationship to household head
race		Race
marst		Marital status
vetstat		Veteran status
momloc		Person number of first mother (from programming)
poploc		Person number of first father (from programming)
sploc		Person number of spouse (from programming)
famsize	Y	Number of own family members in hh
nchild	Y	Number of own children in household
nchlt5	Y	Number of own children under age 5 in hh
eldch	Y	Age of eldest own child in household
yngch	Y	Age of youngest own child in household

Continued on next page

Table 4 – continued from previous page

Variables	Continuous	Description
nsibs	Y	Number of own siblings in household
aspouse		Spouse line number (self-reported)
pecohab		Cohabiting partner line number (self-reported)
pelnmom		Mother's line number (self-reported)
pelndad		Father's line number (self-reported)
pemomtyp		Mother's relationship to child (self-reported)
pedadtyp		Father's relationship to child (self-reported)
ftype		Family Type
famkind		Kind of family
famrel		Relationship to family
bpl		Birthplace
yrimmig		Year of immigration
citizen		Citizenship status
mbpl		Mother's birthplace
fbpl		Father's birthplace
nativity		Foreign birthplace or parentage
empstat		Employment status
classwkr		Class of worker
whyunemp		Reason for unemployment
wnftlook		When last worked full time 2 consecutive weeks (looking last week)
wkstat		Full or part time status
educ		Educational attainment recode
diffhear		Hearing difficulty
diffeye		Vision difficulty
diffrem		Difficulty remembering
diffphys		Physical difficulty
diffmob		Disability limiting mobility
diffcare		Personal care limitation
diffany		Any difficulty
classwly		Class of worker last year
fullpart		Worked full or part time last year
pension		Pension plan at work
firmsize		Number of employees
wantjob		Want regular job now
whyptly		Reason for working part-time last year
usftptlw		Usually work full time (part time last week)
payifabs		Paid if absent from work last week
numemps	Y	Number of employers last year
strechlk		Stretches of looking for work last year
actnlfly		Activity when not in labor force last year (part-year workers)
poverty		Original poverty status (PUMS original)
schllunch	Y	Family market value of school lunch
spmpov		SPM unit's poverty status
spmmort		SPM unit's tenure/mortgage status
spmmedxpns	Y	SPM unit's medical out-of-pocket and Medicare B subsidy
spmchxpns	Y	SPM unit's child care expenses - not capped

Continued on next page

Table 4 – continued from previous page

Variables	Continuous	Description
spmcapxpns	Y	SPM unit's capped work and child care expenses
spmwkxpns	Y	SPM unit's work expenses - not capped
migsta1		State of residence 1 year ago
whymove		Reason for moving
migrate1		Migration status, 1 year
disabwrk		Work disability
health		Health status
quitsick		Quit job or retired for health reasons
inclugh		Included in employer group health plan last year
paidgh		Employer paid for group health plan
emcontrb	Y	Employer contribution for health insurance
himcaidly		Covered by Medicaid last year
himcarely		Covered by Medicare last year
hichamp		Covered by military health insurance last year
covergh		Covered by group health insurance, last year
coverpi		Covered by private health insurance, last year
phinsur		Reported covered by private health insurance last year
phiown		Private health insurance in own name last year
mocaid		Months of Medicaid coverage last year
moop	Y	Total family medical out of pocket payments (in dollars)
hipval	Y	Total family payments (in dollars) for health insurance premiums
verify		Verification: Did individual actually have health insurance
grpdeply		Dependent covered by employment-based insurance last year
grpownly		Policyholder for employment-based insurance last year
grpoutly		Employment-based insurance covered non-household member
grptyply		Type of employment-based coverage last year
grpwho1		Line number of first policyholder of employment-based insurance
dpdeply		Dependent for direct-purchase insurance, previous year
dpownly		Policyholder for direct-purchase insurance, previous year
dpoutly		Direct-purchase private coverage for non-hh member last year
dptyply		Type of direct-purchase insurance plan, previous year
dpwho1		Line number of first policyholder of direct-purchase insurance
trccovly		Covered by Champus/Tricare last year
militva		Covered by VA or Military health care last year
inhcovly		Covered by Indian Health Service last year
out		Covered by policy of person outside the household
hiurule		HIU pointer rule
hcovany		Any insurance, public or private (summary)
hcovpriv		Any private insurance (summary)
hinsemp		Employer-sponsored insurance (summary)
hinspur		Individually purchased insurance (summary)
hcovpub		Any public insurance (summary)
hinscaid		Any Medicaid/SCHIP/other public insurance (summary)
hinscare		Medicare coverage (summary)
hinsmil		Any military insurance (summary)
gotwic		Received WIC

Continued on next page

Table 4 – continued from previous page

Variables	Continuous	Description
paidhour		Paid by the hour
union		Union membership
eligorg		(Earnings) eligibility flag
occ2010		Occupation, 2010 basis
occ10ly		Occupation last year, 2010 basis
ind1990		Industry, 1990 basis
ind90ly		Industry last year, 1990 basis

References

Chen, T. and C. Guestrin (2016). Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, pp. 785–794.

Flood, S., M. King, R. Rodgers, S. Ruggles, and J. R. Warren (2020). Integrated public use microdata series, current population survey: Version 7.0.

Ke, G., Q. Meng, T. Finley, T. Wang, W. Chen, W. Ma, Q. Ye, and T.-Y. Liu (2017). Lightgbm: A highly efficient gradient boosting decision tree. In *Advances in neural information processing systems*, pp. 3146–3154.

LeCun, Y., Y. Bengio, and G. Hinton (2015). Deep learning. *nature 521*(7553), 436–444.

Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society: Series B (Methodological) 58*(1), 267–288.

Wolpert, D. H. (1992). Stacked generalization. *Neural networks 5*(2), 241–259.