Predicting the Unemployment Rate

Lauren Schmiedeler

Introduction

The unemployment rate (the number of unemployed individuals divided by the total number of individuals in the labor force) is a critical macroeconomic variable that gives significant insight into the condition of the overall United States economy. Every month, the unemployment rate is calculated using the Current Population Survey (or Household Survey), which is a monthly survey of households carried out by the Bureau of Labor Statistics. In my project, I attempt (1) to predict the unemployment rate in a given month given other economic variables from that month and (2) to predict whether the unemployment rate in a given month is increasing or decreasing (relative to the unemployment rate in the previous month) given other economic variables from that month. In short, I attempt to predict the unemployment rate and the change in the unemployment rate without using any data obtained via the Current Population Survey.

Aaron S. Kreiner at Oberlin College completed a similar project in May 2019 and reported his findings in a paper entitled "Can Machine Learning on Economic Data Better Forecast the Unemployment Rate?". He concluded that the machine learning methods of Lasso regression and especially neural networks forecast the unemployment rate four quarters ahead better than the Survey of Professional Forecasters (SPF). However, Kreiner's project is not identical to mine as he attempted to forecast the unemployment rate in a future period using data from previous periods while I attempt to predict the unemployment rate (and the change in the unemployment rate) in a given period using data from that same period.

Method

I obtained the data used in this project from FRED (Federal Reserve Economic Data), the online database of The Federal Reserve Bank of Saint Louis. I used the FRED API, which allows developers to retrieve data from FRED and incorporate this data into their programs, and the full_fred interface, which translates every type of request FRED supports to Python, to collect the data. FRED contains more than 800,000 time series of economic data split into 8 different categories, and of these 800,000 series I selected the 800 (in 5 different categories) that satisfy the following conditions.

1. The series must start on or before January 1980 and end on or after December 2020.
2. The series must not contain any missing values during the time period from January 1980 to December 2020.
3. The series must not be discontinued.
4. The series must be monthly (or convertible to monthly).
5. The series must belong to one of the five categories:

1. Money, Banking, & Finance (ID: 32991)
2. Population, Employment, & Labor Markets (ID: 10)
3. National Accounts (ID: 32992)
4. Production and Business Activity (ID: 1)
5. Prices (ID: 32455)

6. The series must not belong to the Current Population Survey (Household Survey) category (which is a subcategory of Population, Employment, & Labor Markets) because the data collected from this survey includes the unemployment rate.

The category breakdown of the 800 selected series is as follows.

- 108 series in Money, Banking, and Finance meet the above criteria.
- 157 series in Population, Employment, and Labor Markets meet the above criteria.
- 37 series in National Accounts meet the above criteria.
- 52 series in Production and Business Activity meet the above criteria.
- 446 series in Prices meet the above criteria.

After obtaining the time series from FRED that meet the above criteria, I first filtered each series so that it only includes data from January 1980 to December 2020 (a span of 732 months). Note that there is no missing data in these series during this time period because series with missing values during this period were excluded from selection. I then separated the feature variables from the output variables and created training and testing sets such that the testing sets each contain 20% of the full data. Note that there is a training set (and corresponding testing set) that is used in the regression models and another training set (and corresponding testing set) that is used in the classification models. Finally, I standardized both sets of training and testing features using sklearn's StandardScaler(). I fit the scaler using only the training features and then transformed both the training and testing sets using this scaler.

The final data includes 732 samples of 800 features. Input features are either float or integer values with some integer features including only values 0 and 1 (binary conditions). The output variable is either a float (the unemployment rate) or an integer (the change in the unemployment rate relative to the previous month where a decrease corresponds to the integer value 0 and an increase corresponds to the integer value 1). The distributions of the output variables are seen in Figures 1 and 2 below.

Figure 1: Unemployment Rate Distribution

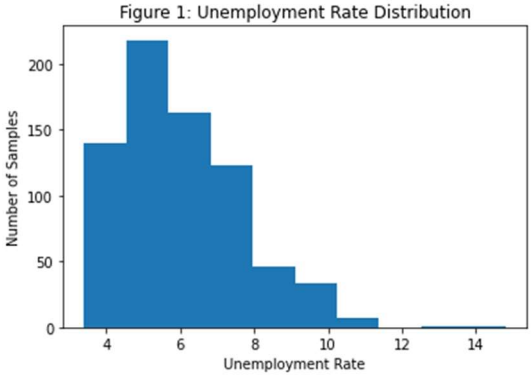
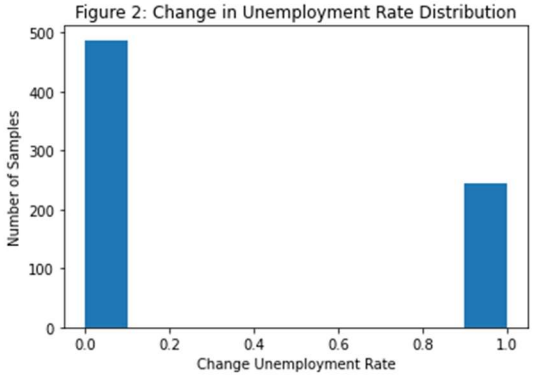


Figure 2: Change in Unemployment Rate Distribution



As seen in the above figures, neither output distribution is particularly balanced. The distribution for the unemployment rate is right skewed, and the distribution for the change in unemployment rate has about double the number of 0's as 1's (so the unemployment rate decreases much more than it increases during this time period).

To predict the unemployment rate I use linear regression, ridge regression, k-nearest-neighbors regression, random forests, and neural networks. Each of these models is described in detail below.

- Linear regression is the simplest of these models and makes predictions by computing a weighted sum of the input features plus a bias term. Training a linear regression model involves finding the model parameters (weights and bias term) that best fit the training data. This model utilizes a cost function, in this case the root mean squared error function, to measure how well a given set of model parameters fits the training data (Geron 112-113).
- Ridge regression is a regularized version of linear regression in which a regularized term is added to the cost function. This term forces the model weights to remain as small as possible while still fitting the data (Geron 135).
- K-nearest-neighbors regression requires minimal training. In order to predict the output value of a test sample, k-nearest-neighbors regression finds the k training samples closest (based on the distance between the feature values) to this test sample and predicts the average of the output values of these nearest neighbors.
- Random forests are ensembles of decision trees and introduce randomness when building trees by searching for the best splitting feature among a random subset of features (instead of search for the best feature among all the features). Random forest models trade higher bias for lower variance and generally result in better models than the traditional decision tree model (Geron 197).
- Neural networks are inspired by the networks of biological neurons found in human brains and include collections of nodes that make them adept at handling large and complex tasks. The neural networks I used in this project are relatively simple and include only (relatively few) Dense layers (Geron 279).

To evaluate the results of each of these regression models I examine the mean squared error between the actual values and the values predicted by each model.

To predict the change in the unemployment rate I use Decision Trees, Logistic Regression, KNN Classification, Random Forests, and Neural Networks.

- Decision trees are easy to visualize models that infer simple decision rules from the given data. Classifying a particular sample using a decision tree involves starting at the root node and moving down the tree until you eventually reach a leaf node that includes a predicted class (Geron 176-177).
- Logistic regression is similar to linear regression in that it computes a weighted sum of the input features and a bias term, but unlike linear regression, it outputs the logistic of this result. This model estimates the probability that an instance belongs to a particular

class (in this case, the model estimates the probability that the unemployment rate increases in a given month) (Geron 143).

- The regression forms of k-nearest-neighbors, random forests, and neural networks are described above, and their classification forms function similarly.

To evaluate the results of these classification models I examine the accuracy score, precision, recall, and F1 score.

Results

I perform hyperparameter tuning using sklearn's RandomizedSearchCV() with 10-fold cross-validation on the following models with the following tuning parameters (the selected parameters are bolded).

- Ridge Regression: alpha = [0, **0.01**, 0.02, 0.03, …, 0.97, 0.98, 0.99, 1]
- KNN Regression: n_neighbors = [**1**, 2, 3, 4, 5, 6, 7, 8, 9, 10]
- Neural Network (Regression): n_layers = [1, 2, **3**], n_neurons = [10, 20, **30**, 40, 50]
- Decision Tree (Classification): max_depth = [1, 2, 3, **4**, 5, 6, 7, 8, 9, 10, None], max_leaf_nodes = [**2**, 3, 4, 5, 6, 7, 8, 9, 10, None], criterion = [**gini**, entropy]
- KNN Classification: n_neighbors = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, **15**, 16, 17, 18, 19, 20], metric = [**euclidean**, minkowski, cosine]
- Neural Network (Classification): n_layers = [**1**, 2, 3], n_neurons = [10, 20, 30, 40, **50**]

The results for the tuned regression and classification models are shown in Tables 1 and 2 below, and these results show that the best regression model is ridge regression, and the best classification model is a decision tree. Additionally, linear regression is by far the worst of the regression models, and the accuracy scores for the classification models have much less variation than the mean squared errors for the regression models. Furthermore, all of the classification models have relatively low recall scores, which indicates that none of the models are classifying positive (increasing unemployment rate) samples particularly well.

Table 1: Regression Results

	Model	MSE
0	Linear Regression	0.285064
1	Ridge Regression	0.040110
2	KNN	0.104490
3	Random Forest	0.079080
4	Neural Network	0.069116

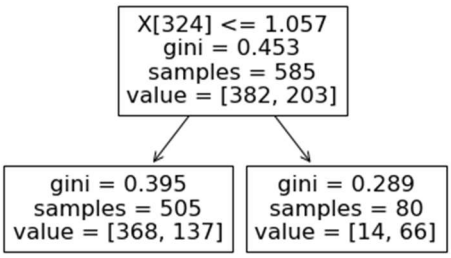
Table 2: Classification Results

	Methods	Accuracy	Precision	Recall	F1-Score
0	Decision Tree	0.768707	0.666667	0.380952	0.484848
1	Logistic Regression	0.707483	0.486486	0.428571	0.455696
2	KNN	0.727891	0.555556	0.238095	0.333333
3	Random Forest	0.687075	0.437500	0.333333	0.378378
4	Neural Network	0.707483	0.488889	0.523810	0.505747

The decision tree produced by the tuned parameters specified above is pictured below in Figure 3. This tree splits only on the binary factor X[324], which represents the FRED series with id 'USRECM' and name 'NBER based Recession Indicators for the United States from the Peak

through the Trough.' This tree predicts 1 (unemployment rate increasing) if the economy is in a recession and 0 (unemployment rate decreasing) if the economy is not in a recession. As the decision tree is the best classification model, this indicates that this recession indicator is an important classification feature.

Figure 3: Decision Tree



The 8 most important features for ridge regression, random forest regression, logistic regression, and random forest classification are listed below in Tables 3-6. 6 of the 8 most important features for ridge regression are price indicators (either consumer price index or producer price index), and 6 of the 8 most important features for random forest classification are recession indicators. This fact about random forest classification is not surprising considering the optimal decision tree splits only once on a recession indicator. When retraining the logistic regression and random forest classification models using only the 8 most important features, accuracy remains about the same at around 0.70 for both models (the accuracy for random forest classification actually increases slightly). Retraining the random forest regression model using only the important features listed below actually decreases the mean squared error from about 0.08 to 0.06. Both the random forest regression and classification models are improved by considering only the important features. However, retraining the ridge regression model using only the important features listed below leads to a significant increase in the mean squared error from about 0.04 to about 1.64 (which is well above the mean squared error for linear regression, which was by far the worst regression model).

Table 3: Ridge Regression Feature Importance

	Series ID	Series Name	Importance
4	CWUR0000SA0R	Consumer Price Index for All Urban Wage Earners and Clerical Workers: Purchasing Power of the Consumer Dollar in U.S. City Average	-1.727205
3	CUUR0000SA0R	Consumer Price Index for All Urban Consumers: Purchasing Power of the Consumer Dollar in U.S. City Average	-1.691057
6	WPU1597	Producer Price Index by Commodity: Miscellaneous Products: Brooms and Brushes	-1.545716
0	DDURRG3M086SBEA	Personal consumption expenditures: Durable goods (chain-type price index)	-1.277709
5	WPU125	Producer Price Index by Commodity: Furniture and Household Durables: Home Electronic Equipment	1.267154
2	CWSR0000SAD	Consumer Price Index for All Urban Wage Earners and Clerical Workers: Durables in U.S. City Average	1.262876
1	MVGFD027MNFRBDAL	Market Value of Gross Federal Debt	1.259814
7	PCU311422311422	Producer Price Index by Industry: Specialty Canning	1.253810

Table 4: Random Forest (Regression) Feature Importance

	Series ID	Series Name	Importance
5	CES3000000034	Indexes of Aggregate Weekly Hours of Production and Nonsupervisory Employees, Manufacturing	0.744695
2	CES0600000034	Indexes of Aggregate Weekly Hours of Production and Nonsupervisory Employees, Goods-Producing	0.733757
3	CES1021000001	All Employees, Mining	0.211996
6	CEU3100000001	All Employees, Durable Goods	-0.150672
4	USMINE	All Employees, Mining and Logging	0.129658
1	BAAFFM	Moody's Seasoned Baa Corporate Bond Minus Federal Funds Rate	0.049083
7	HOUST	New Privately-Owned Housing Units Started: Total Units	-0.018861
0	AAA	Moody's Seasoned Aaa Corporate Bond Yield	0.004836

Table 5: Logistic Regression Feature Importance

	Series ID	Series Name	Importance
2	CES0600000007	Average Weekly Hours of Production and Nonsupervisory Employees, Goods-Producing	1.004804
5	DPCERGM1M225SBEA	Prices for Personal Consumption Expenditures: Chained Price Index	-0.967929
0	BAA10YM	Moody's Seasoned Baa Corporate Bond Yield Relative to Yield on 10-Year Treasury Constant Maturity	-0.872860
7	WPU0113	Producer Price Index by Commodity: Farm Products: Fresh and Dry Vegetables	-0.865740
3	CEU3000000009	Average Weekly Overtime Hours of Production and Nonsupervisory Employees, Manufacturing	-0.772361
6	DSERRGM1M225SBEA	Prices for Personal Consumption Expenditures: Chained Price Index: Services	0.737637
1	NONREVSLAR	Percent Change of Total Nonrevolving Consumer Credit	0.727584
4	DGDSRGM1M225SBEA	Prices for Personal Consumption Expenditures: Chained Price Index: Goods	0.677677

Table 6: Random Forest (Classification) Feature Importance

	Series ID	Series Name	Importance
6	USRECM	NBER based Recession Indicators for the United States from the Peak through the Trough	0.661386
2	USREC	NBER based Recession Indicators for the United States from the Period following the Peak through the Trough	0.213447
3	USRECD	NBER based Recession Indicators for the United States from the Period following the Peak through the Trough	0.213447
5	USRECDP	NBER based Recession Indicators for the United States from the Peak through the Period preceding the Trough	-0.111532
7	USRECP	NBER based Recession Indicators for the United States from the Peak through the Period preceding the Trough	-0.111532
1	FLNONREVSLA	Nonrevolving Consumer Credit Owned and Securitized, Flow	0.033500
4	USRECDM	NBER based Recession Indicators for the United States from the Peak through the Trough	-0.013230
0	TB6SMFFM	6-Month Treasury Bill Minus Federal Funds Rate	-0.008370

Regarding future work, taking advantage of the time series nature of this data and using data from previous months to predict the unemployment rate and the change in the unemployment rate in a given month would hopefully improve the mean squared error for regression and the various classification metrics. This could involve using more advanced neural network techniques such as RNN and LSTM. In his paper, Kreiner performed this analysis for the regression portion of this project but not for the classification portion. So, applying more advanced neural networks (especially to the classification portion of the project) that more effectively utilize previous data to predict future data would be a good next step.

References

Geron, Aurelien. *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow*. O'Reilly Media, 2019.

Kreiner, Aaron S., "Can Machine Learning on Economic Data Better Forecast the Unemployment Rate?" (2019). Honors Papers. 126.