What Do You Need To Predict Somebody's Salary?

Introduction

The data science and machine learning (DS/ML) field has been growing exponentially during the past 10 years. New positions have been open in basically any industry, and Data Scientists are in large demand. But not all candidates and positions are the same. The field includes experienced analysts, programmers and researchers switching into this growing field, data scientists who started in the field from the beginning and have now significant DS/ML experience, and a large number of young people just coming out of school. At the same time, the DS/ML field is composed of a large variety of tools developed in different languages, with language specific packages used in different environments, allowing to process and analyze any data from any field. So, if you are going to hire a new recruit, how much should you offer them? We can also ask, if you want to enter the Data Science job market, what skills or characteristics will help you land those high paying jobs?

In this short article we explore and compare ML models that can be used to predict salaries, and we identify the key pieces of information needed to make a good prediction.

The Data

Kaggle launches a Data Science and Machine Learning (DS & ML) Survey every year as a way to learn about the DS field, with questions on respondent's demographics characteristics (age, gender, education level, working status), their data science and machine learning knowledge and experience, and methods and tools they use or would like to get familiar with. The questionnaire also asks about working status and compensation level. The survey was distributed to the entire Kaggle community through the Kaggle (opted-in) email list, and promoted on the Kaggle website and Kaggle Twitter channel.

We use the survey to predict salaries in the community. There are 20,036 answers to the survey in 2020 from 171 different countries, including active workers, unemployed individuals and students. The number of active non-student workers in the survey is 13,213, and after dropping observations with missing compensation, the sample size is reduced to 10,729 individuals.

We identify a subset of survey questions that could be relevant to predict salaries, and we transform survey answers so they will be easier to process. Multiple answer questions are transformed into a set of indicator variables for each of the possible answers. Answers that are categorical, but numeric in nature (e.g. age group) are transformed into continuous variables using range midpoints.

Observations with missing values are dropped from the analysis, with the exception of the use of ML/DS tools, with a list to choose from (one or many). Each question has an option for "Other" or "None", but given the high number of non-responses in these questions, we assume that a none response implies "None".

Our target variable is defined with the question: "What is your current compensation (approximate $USD)?" Note that here we use compensation and salaries indistinctly. In order to model salaries

we want to consider five types of variables: demographic characteristics, geographic region, job position, experience, and DS/ML tools. Demographic variables considered are age level, education, and gender. To account for salary disparities around the globe, we created indicators for top 10 respondent countries (India, US, Brazil, Japan, Russia, UK, Germany, Nigeria, Spain, and Canada), and indicators for low income, medium-low income, medium-high income, and high income countries using data from the World Population Review. We identified two variables that indicate the type of position and employer for the individual: the job title, and the number of employees at the company. Variables considered to reflect experience are years of experience programming, years of experience using machine learning methods (ranges converted to continuous variables using range midpoint). We also consider the number of languages the person uses to program, indicators for specific languages (Python, R, SQL, C, C++, Java, Javascript, Julia, Swift, Bash, Matlab), the use of visualization methods, cloud computing platform, deep learning workstation or a personal computer/laptop, the use of advance statistical software (SPSS, SAS, etc.), basic statistical software (Excel, Google Sheets, etc), business intelligence software, or local development environments.

Household median income by country was downloaded from the World Population Review website, and countries were classified into four income levels using household median income quartiles.

Modeling Salaries

We try two different ML modeling approaches to predict salaries, linear models and random forest. The survey is very extensive and we have a long list of potential features that can be used to predict salaries that can lead to model overfitting. To mitigate this problem we will use cross validation methods to evaluate model fit and select the best method to predict salaries. We also use two regularization models, the Ridge Regression and the LASSO Regression. For the random forest fit we first do a grid search to optimize the parameters before modeling salaries. Given the high level of skewness in the distribution of the outcome variable we transform it using natural logs. We also normalize the matrix of covariates using the Scikit Learn scaler.

We primarily use the coefficient of determination (R2), the mean squared error, and mean absolute error (MAE) to evaluate and compare the models. A lower MSE and MAE implies a better fit. The coefficient of determination, or R2 is also commonly used, and it's based on squared errors and it can be interpreted as a proportion. The R2 compares the model fit with a prediction based on simple averages. A higher R2 implies a better fit. These two measures are part of the scikit-learn library. The last measure we will use is built here, the salary range score function (range_score_func). Since the original measure is presented in ranges, we measure how often the model predicts the right range. We take the continuous prediction from the models and we transform it into the compensation categories presented in the survey.

We first fit two linear models, a basic model with just demographic variables, experience, and country of origin, and a full model where we also include information on the type of position and tools used by the individual. This can help in evaluating how much the additional features add to the predictive power of the model. We fit the regular linear regression model, and also the Ridge

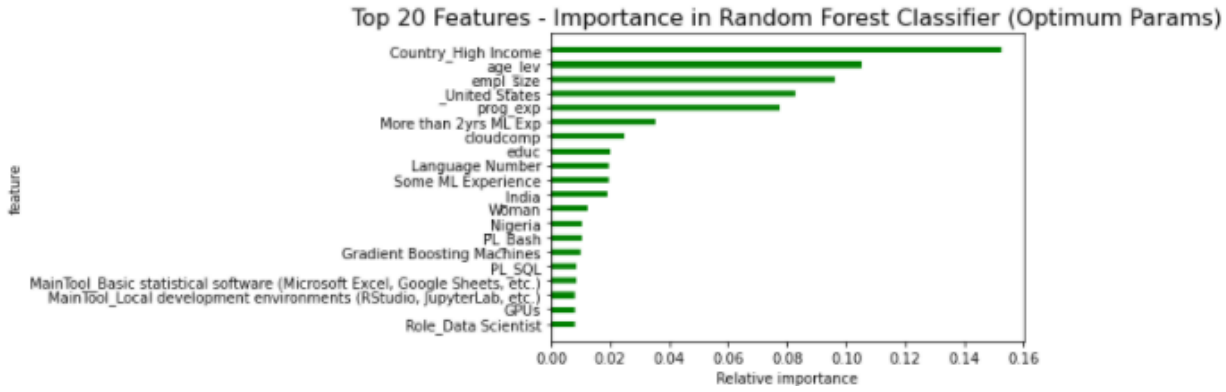
and Lasso regression models that can help the fit when the covariates are correlated. The model significantly improves when using the full set of covariates, with a shift from 1.29 to 1.21 in the MAE and from 0.36 to 0.42 in the R2 for the regular linear regression. The Ridge or Lasso regression don't improve the fit, so we run the regular regression on the training dataset and get a final evaluation using the test data.

The linear fit may not be the best approach given the non-linear nature of the salary and covariates relationship. We can model and predict salaries using the Random Forest, a type of bagging method that relies on the strength of fitting multiple trees and averages. The random forest randomizes not only samples but also the features used to build each of the trees. We first fit the model with some basic and default parameters, and then we run a grid search to identify the best parameters for the forest.

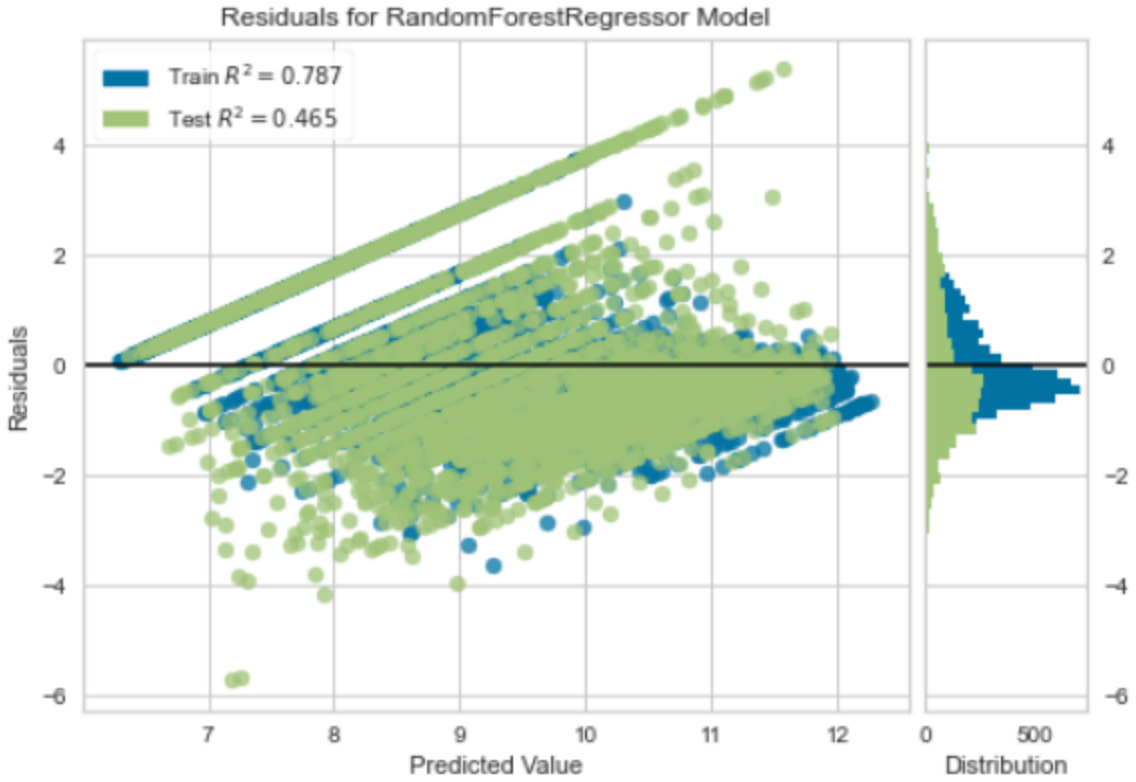
The random forest model improves the fit when compared with the linear model but before looking further, let's see if the model can be improved. We try an extensive search over all parameters in the random forest model. We first create a grid with points for each parameter, to fit the random forest and then we use the randomized search (RandomizedSearchCV) estimator, that will pick a random set of parameters from the grid and fit the model applying the cross validation method. Once the function finds the optimum set of parameters, we run a comprehensive search around them to refine the optimum point.

	Model	MSE	MAE	R2
0	Linear Regression	2.304	1.201	0.421
1	Ridge Regression	2.304	1.201	0.421
2	Lasso Regression	3.979	1.728	-0.000
0	Random Forest - Basic	2.138	1.159	0.436
1	Random Forest - Optimized	2.138	1.159	0.463

The optimized forest fit with the identified parameters from the grid search shows a large improvement in the model fit. Using the R2 measure it's a clear improvement not only on the linear model fit, but also on the Random Forest fit using the default parameters. Using the final identified parameters we train the Random Forest and evaluate it on the test set. Then we look into the most important features identified in this final model.



The Random Forest model using all covariates has the best fit. There is still a large amount of unexplained variation, with a significant error, not explained by the covariates. However, there are a few features that stand out in their explanation power. Of course country of residence is highly relevant. In particular, being from a high income country, or a US or Nigeria residence indicators, are among the top explanatory variables. The second set of variables with high prediction power are related to experience: age, programming experience, significant experience (more than 2 years) in machine learning methods. Employer size is another highly relevant variable. Finally, the knowledge and use of some specific tools or environment have also significant predictive power.



The errors from the fit are still clearly not random and still quite large. The Random Forest gives a decent fit but it seems to overestimate salaries when they are on the lower end of the spectrum, and underestimate them when they are on the high end of the spectrum.