An Analysis of Job Status and Compensation in San Francisco Salaries

Zac Taylor

Department of Data Sciences, College of Charleston

Introduction

The goal of this research was to develop an accurate model to predict job status for the city of San Francisco employees in 2014. Then use the same model to predict the job status of employees from the previous years. The R language and excel were used to create several predictive models and explore the dataset.

Dataset Information

The city of San Francisco has released names, job titles and compensations of their employees from 2011 – 2014. Kaggle.com has compiled and released this dataset for public use. It contains over 148,000 observations and 11 useful variables. The majority of this research was focused on data from 2014 because that was the only year where job status was recorded.

Methods

Data cleanup was the first major hurdle for this project. A number of variables were removed (name, agency, notes) because they provided no value. Several observations were also removed due to incomplete data. Additionally the Support Vector Machine (SVM) and neural network models required the data to be normalized.

Predictive modeling was started by implementing a neural network which caused several issues, making the convincing argument to start with something less complex.

A decision tree was chosen as a simple baseline model. At the default setting it only branched once based on benefits. Altering the complexity parameter allowed a more detailed tree and slightly more accurate model, however it prompted a closer analysis of the data.

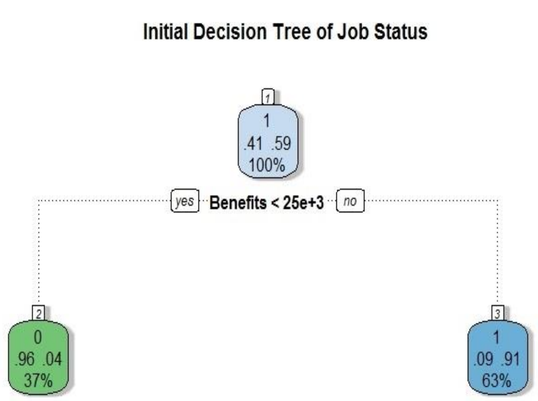
A correlation matrix was used to see how highly correlated the variables are. This determined that a majority of the variables used are all very highly correlated with each other (over 90%). Principal Component Analysis (PCA) was used to compliment the findings of the correlation matrix.

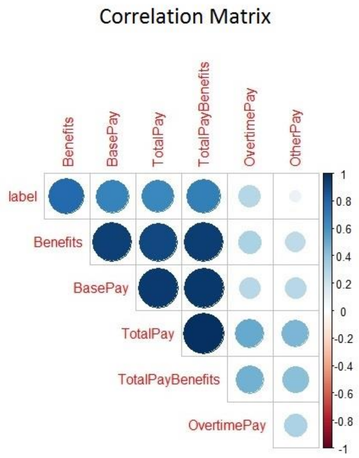
The original dataset was kept and compared to a minimal dataset where all the highly correlated variables except benefits were removed. This allowed for a much faster training time, especially with the neural network.

After the decision tree model was established as a baseline, more complicated models were implemented in an attempt to achieve more accurate results. These included random forests, support vector machines and neural networks.

All the data for the models was randomly sampled from 2014 and split into training and testing sets for validation.

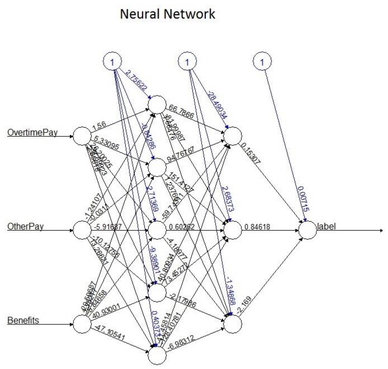
Results

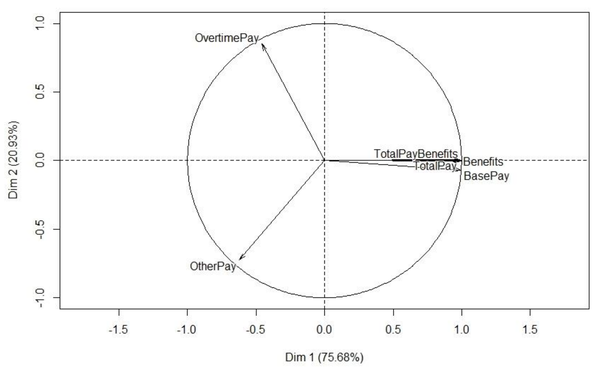




Numeric Values Of Correlation Matrix

	BasePay	OvertimePay	OtherPay	Benefits	TotalPay	TotalPayBenefits	label
BasePay	1.0000000000	0.2750969276	0.2787345116	0.9323154101	0.9567053397	0.9677359294	0.6644885615
OvertimePay	0.2750969276	1.0000000000	0.3102490669	0.3162700631	0.5087269046	0.4720370919	0.2869029036
OtherPay	0.2787345116	0.3102490669	1.0000000000	0.2507473567	0.4584217365	0.4172398748	0.0896779139
Benefits	0.9323154101	0.3162700631	0.2507473567	1.0000000000	0.9041941354	0.9427995199	0.7751197940
TotalPay	0.9567053397	0.5087269046	0.4584217365	0.9041941354	1.0000000000	0.9948592066	0.6450672524
TotalPayBenefits	0.9677359294	0.4720370919	0.4172398748	0.9427995199	0.9948592066	1.0000000000	0.6872383651
label	0.6644885615	0.2869029036	0.0896779139	0.7751197940	0.6450672524	0.687238365	1.0000000000





Initial Results

	Overall Accuracy	PT Accuracy	FT Accuracy
Decision Tree	0.9290099	0.9120269	0.9571767
Random Forest	0.9663964	0.9546928	0.9663964
SVM	0.9515714	0.9688549	0.9404823
Neural Net	0.9816030	0.9789056	0.9852908

Results on minimal Dataset

	Overall Accuracy	PT Accuracy	FT Accuracy
Decision Tree	0.9267550	0.9094957	0.9559446
Random Forest	0.9301480	0.9168121	0.9520421
SVM	0.9258106	0.9031567	0.9655870
Neural Net	0.9657292	0.9547920	0.9723041

Conclusions

The initial decision tree could predict with 91% accuracy whether a person is full time and with 96% accuracy on whether they are part time based on benefits being above or below $25,000. It had an overall accuracy of 92.9%

The correlation matrix and its output show that benefits, base pay, total pay and total pay benefits are all over 90% correlated.

PCA variables factor map shows a significant clustering of the same variables as the correlation matrix. With 75.68% of the variance explained by dimension 1 and 20.93% of the variance explained by dimension 2.

This knowledge prompted the creation of a simpler dataset for model testing. Removing the correlated variables greatly reduced the running time of all the algorithms, the neural network in particular, with minimal differences in accuracy.

The accuracy table at the bottom of the center column shows that the neural network was the most successful model followed by the random forest. The minimal dataset results show that the decision tree is barely affected by the removal of the variables while the other models lose 1.5-3.5% accuracy.

Next Steps

The next steps in this research will be to continue tuning the models for improved accuracy. Then apply the neural network or random forest model to add a job status label to years 2011-2013.

Job titles need to be processed into factors to allow them to be incorporated into future models.

The goal is to use these new variables to provide more comprehensive datasets that will produce more accurate models.

References

"Kaggle: The Home Of Data Science". *Kaggle.com*. N.p., 2016. Web. 20 Apr. 2016.

"The Comprehensive R Archive Network". *Cran.r-project.org*. N.p., 2016. Web. 14 Feb. 2016.

Alice, Michy. "Fitting A Neural Network In R; Neuralnet Package". *R-bloggers*. N.p., 2015. Web. 15 Mar. 2016.

"Correlation Matrix : A Quick Start Guide To Analyze, Format And Visualize A Correlation Matrix Using R Software - Documentation - STHDA". *Sthda.com*. N.p., 2016. Web. 19 Apr. 2016.

"Creating, Validating And Pruning The Decision Tree In R | Edureka Blog". *Edureka Blog*. N.p., 2015. Web. 19 Apr. 2016.

"Principal Component Analysis In R : Prcomp() Vs. Princomp() - R Software And Data Mining - Documentation - STHDA". *Sthda.com*. N.p., 2016. Web. 19 Apr. 2016.

"The Caret Package". *Topepo.github.io*. N.p., 2016. Web. 19 Apr. 2016.