**Implementing Machine Learning and Deep Learning techniques to predict and evaluate Overtime Awards**

Celestino B. Edaňo Jr.

January - March 2018

**Introduction**

The concept of big data has been around for years; most companies now understand that if they capture all the data that streams into their businesses, they can apply analytics and get significant value from it. Big data analytics helps organizations harness their data and use it to identify new opportunities.

The new benefits that big data analytics brings to companies are speed and efficiency. Whereas a few years ago a business would have gathered information, run analytics and unearthed information that could be used for future decisions, today that business can identify insights from immediate decisions. The ability to work faster and stay agile gives organizations a competitive edge they didn't have before.

In this project proposal, with the help of data I mined from one of Tanda's old client (food manufacturing industry), and with the use of different machine learning and deep learning models or algorithms, I intend to find the best model and set of features to predict the overtime pay rates or awards given to the employees.

**Dataset**

My dataset comes from one of Tanda's old client which is a food manufacturing industry. I used a software called Postman to perform data mining. After gathering one year worth of data, I do data cleansing using Tableau desktop. With the help of Tableau data visualization, I carefully examine the features I need for my project proposal.

I extracted a total of 6 attributes as numerical values, describing different aspects or ways how the employer grants overtime awards, from a total of almost 13,000 different overtime shifts. The 6 attributes or features I extracted are employees clock-in, clock-in day, break (some employees takes a break while others don't), clock-out, clock-out day, and the total working hours (excluding break). I split-up clock-in day from clock-in and clock-out day from clock-out because based on my data visualization, the employer usually gives lower pay rates during Fridays and higher pay rates for those employees who clock-out the next day.

**Methods**

*Software*

The following are the software I used during the project development:

* **Anaconda** - is an open - source distribution of Python and R programming languages for large-scale data processing, predictive analysis, and scientific computing.
* **Jupyter Notebook** - is an open-source web applications that allows the user to create and share documents that contain live code, equations, visualizations and narrative text. It also is also useful for data cleaning and transformation, numerical simulation, statistical modeling, data visualization, machine learning, etc.
* **Postman** - is a powerful HTTP client for testing web services allowing me to quickly put together both simple and complex HTTP requests. I also used this software for data mining.
* **Tableau** **Desktop** - is a data visualization software which comprises a complete business intelligence software solution. I also used this software for data cleansing and a little bit of data visualization.
* **WEKA (for Windows)** - a collection of machine learning algorithms for data analysis used for preliminary analysis and experiment.
* **Google’s Tensorflow** - or simply Tensorflow is an open - source software library for numerical computation using data flow graphs.

*Programming Language*

The following is the programming language I used during the project development:

* **Python** - is an object-oriented and high-level programming language with dynamic semantics. Over the years, using Python is critical in data analysis and data science. This is my first choice programming language to create and evaluate a model that predicts overtime awards.

*Machine learning & deep learning models*

The following are the machine learning algorithms or models I used during the project development:

* **Softmax regression** - a member of logistic regression use for multi-class classification assuming that the classes are mutually exclusive or disjoint.
* **Adam** - is an optimization algorithm mostly use for deep learning task. It updates network weights iterative based on training data.

**Performance measures**

*Objectives*

The main goal of this project proposal is to make use of a largely collected dataset with over 13,000 overtime shifts from the food manufacturing industry, first select informative features and then create a learner that can automate the giving of awards. Specifically, it aims to:

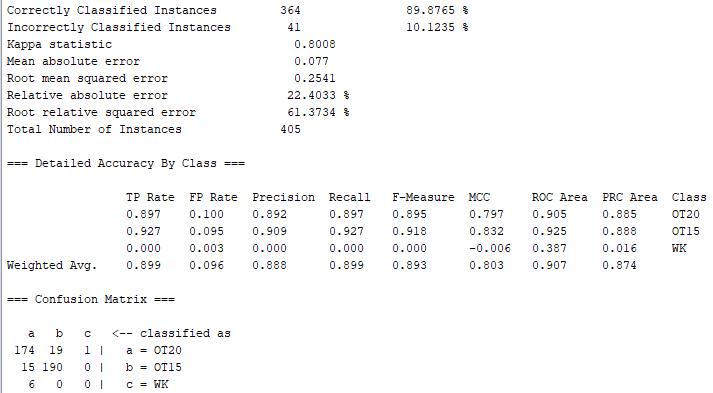
* Gather enough data from food manufacturing industry client.
* Select informative features.
* Apply Softmax regression model to predict and evaluate the overtime awards.
* Use Adam algorithm to optimize the Softmax regression model.

*Methods and tools*

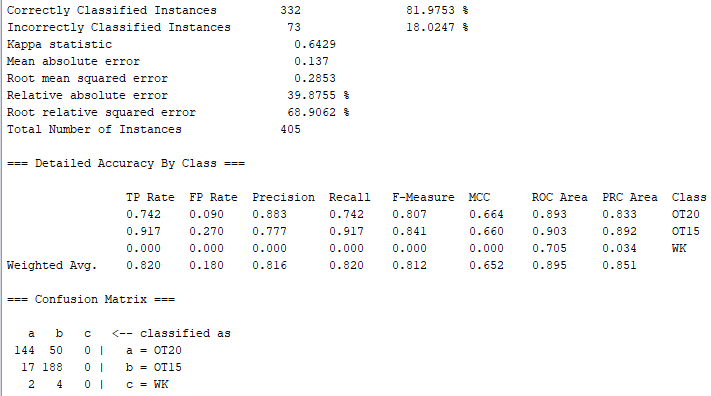
The following are the methods and tools I used during the project proposal development:

* **WEKA** - I used WEKA just for a little bit of preliminary analysis.

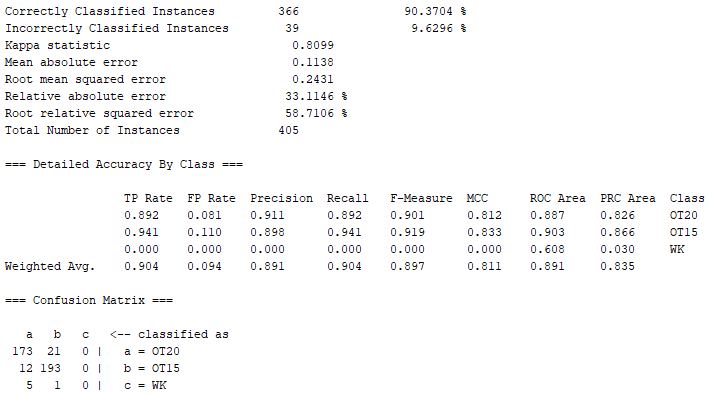
*IBk -* this algorithm is the equivalent of KNN in R and Python. I used cross-validation with 10 folds and here are the results I got:



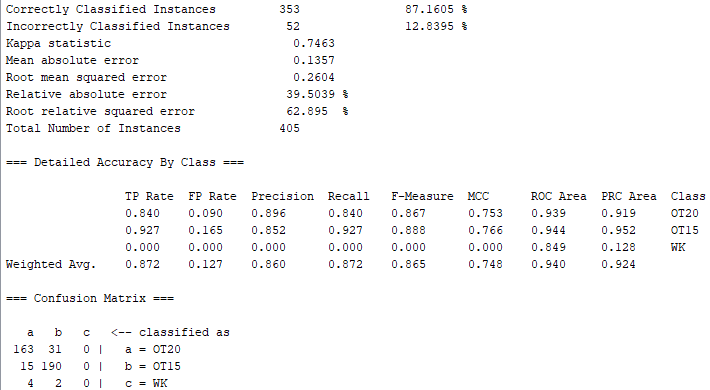
*REPTree (CART)* - this algorithm implements cost-complexity pruning for classification trees. I used cross-validation with 10 folds:

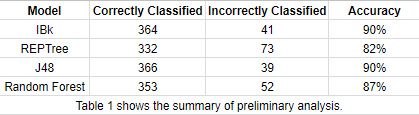


*J48 (C4.5) -* is an algorithm used to generate decision tree:



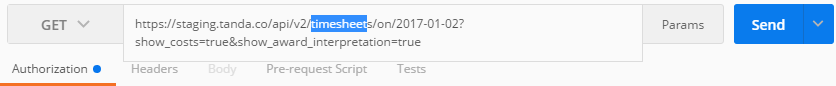
*Random Forest -* this algorithm is good for supervised classification tasks:



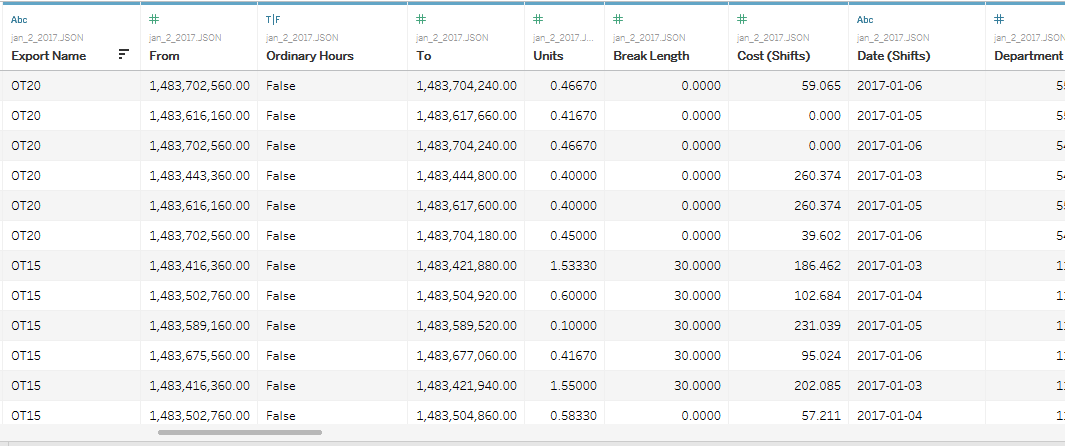


Based on my preliminary analysis with the help of WEKA, decision-tree like algorithms performs better. The problem is that, Tensorflow does not support decision trees at the moment. The reason for this is that Tensorflow is build for deep learning which is much more computationally intensive than other traditional training methods. If I would run a model outside Tensorflow it will take a whole day to complete a single training.

* **Postman -**  I used postman software for data mining. I did it by sending an HTTP request, requesting to return employees timesheets. Afterward, I save the generated file as JSON file.



* **Tableau desktop -** I used Tableau desktop to load all the JSON files I generated via Postman. I also used this software for data cleansing, data analysis, features selection, sample or data gathering, and data splitting.



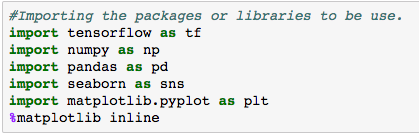
* **Python** **& Tensorflow** - Tensorflow is an open source library for fast numerical computing and also a good tool for machine learning. It contains a wide range of functionality and is mainly designed for deep neural network models. Tensorflow works pretty well with Python 2.7 and Python 3.3+.

*Code analysis*

In this section, I intend to explain how the codes work. For this project proposal, I used Python 2.7 and integrate Tensorflow library for fast numerical computation.

* *Import the packages I used during the development of the project.*

The Python packages I used in this project are tensorflow, numpy, pandas, seaborn, and matplotlib.pyplot. Tensorflow, numpy, and matplotlib libraries are good for dealing scientific and numerical computation with the use of data flow graphs. Pandas is used for reading our dataset, while seaborn is used for data visualization.



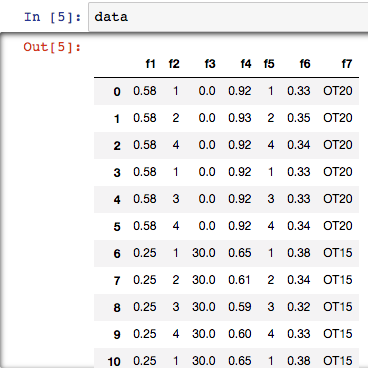
**Figure 1** shows how to import packages or libraries.

* *Importing the dataset.*

The dataset contains a one month of overtime shifts or four timesheets.



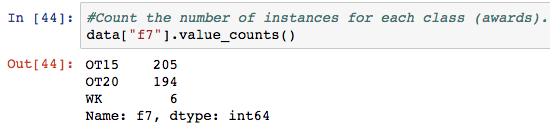
**Figure 2** shows how to import the dataset.



**Figure 3** shows the dataset is imported successfully.

* *Count the number of instances for each class (awards).*

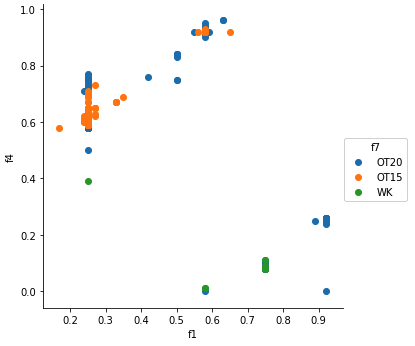
In the output we see that there are 205 shifts for OT15, 194 shifts for OT20, and only 6 shifts for WK or weekend multiplier.



**Figure 4** shows how count the number of instances for each class.

* *Data visualization with the help of seaborn package.*

Using seaborn package, it is easy to visualize each feature. For this example, I chose three features which are clock-in, clock-out, and the award or our class. The x-axis or f1 is the clock-in and the y-axis or f4 is the clock-out and the dots (f7) are the classes or the awards given to the employees.

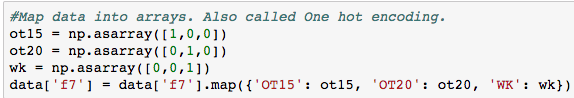


**Figure 5** shows the statistical data visualization of the dataset.

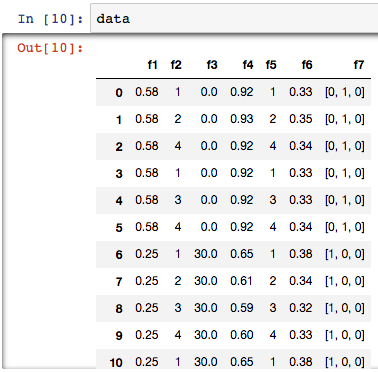
We can see that around 0.37 and 0.75 the employer usually grants OT15 multiplier while from 0.92 to 0.98 the employee grants OT20 multiplier. With this visualization, we can assume that if the employee works longer hours he/she can get an OT20 multiplier.

* *Map the classes (awards) into arrays.*

Mapping the class into arrays is also called One-hot encoding, it is a process by which categorical variables are converted into a form that could be provided to the machine and deep learning algorithms to do a better job in prediction.



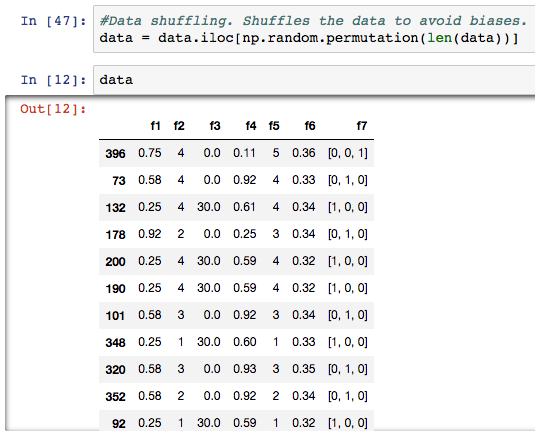
**Figure 6** shows the process of one-hot encoding.



**Figure 7** shows that we successfully converted our categorical value into arrays.

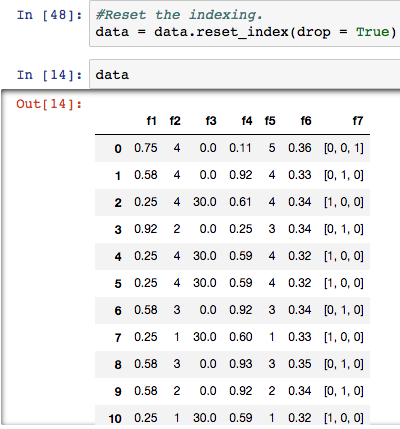
* *Data shuffling.*

Shuffling data serves the purpose of reducing variance and making sure that the model will remain general and overfit less.



**Figure 8** shows the shuffled data.

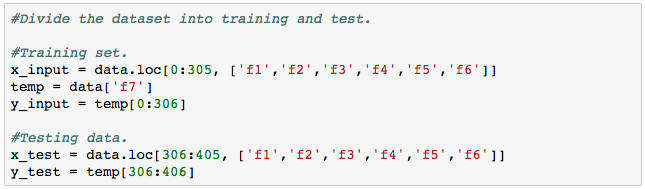
* *Resetting the index.*



**Figure 9** shows the process of index reset.

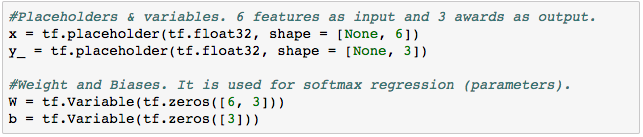
* *Divide the dataset into training and testing set.*

I used a training set for learning, to fit the parameters of the model, and find the optimal weights of the learner while test data is used to assess the performance of a fully-trained learner and also use to estimate the error rate of the model.



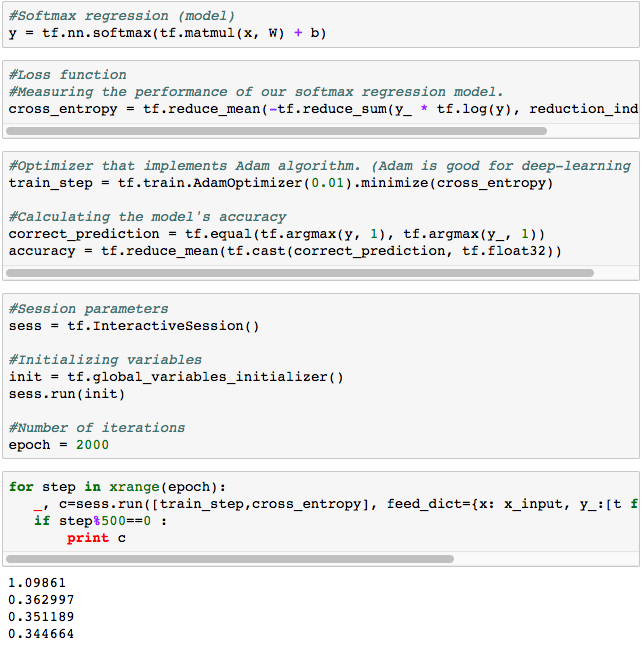
**Figure 10** shows the process of dividing the dataset into training and test set.

* *Assigning placeholders and variables.*

****

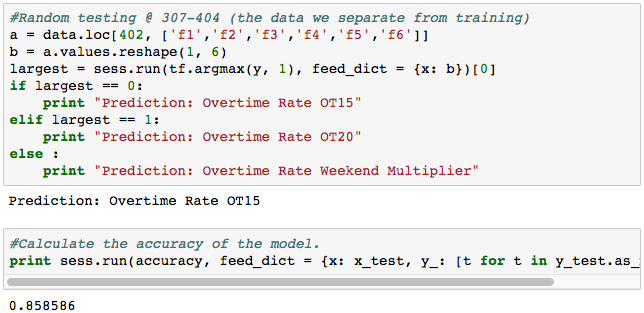
**Figure 11** shows the assigning of placeholders and variables.

* *Training the Softmax model and optimize it using Adam algorithm.*



**Figure 12** shows the implementation of Softmax model and optimization using Adam algorithm.

* *Test and evaluate our final model and calculate the accuracy of each prediction.*



**Figure 13** shows the process of testing or prediction as well as calculating the accuracy of the model.

**Challenges**

* Unfamiliarity with different models - I’m only familiar with supervised learning algorithms like a decision tree, and regression logistics.
* Lack of experience - I admit that this is the first time I created my own dataset. In college, we were given a clean dataset already or we just get it from UCI machine learning repository or Kaggle websites.
* Lack of knowledge about feature engineering and feature selection - for me to be able to increase the performance of my models, I need to understand more about feature engineering and feature selection.
* Too many resources (tutorials about Tensorflow) - the biggest challenge for me to face while learning Tensorflow is not dearth of learning material – but too much of it. I’m not sure where to start learning, what to practice, how much time to spend on a concept, where to get the useful resources etc. For most of the beginners like me, this becomes overwhelming and some of them simply drop out before even learning a single skill.
* Time management - one of the biggest challenge for me as an intern while doing this project is that I’m bad at time management. There’s a lot of things I wanted to experiment like applying models or algorithms I never used before but there are also a lot of things to do in my college that I need to prioritize.

**Future work**

If I were to work again to improve this project proposal that would be great since I can dedicate my time to this proposal without worrying on other stuffs. There are also a lot of things a wanted to try and I know there is still room for improvements. The ideas I have in mind are:

* Add more data - having more data is always a good idea. It allows the data to tell for itself, instead of relying on assumptions and weak correlations. Presence of more data results in better and accurate models.
* Feature engineering - I will focus on feature engineering to extract more information from existing data. New information is extracted in terms of new features. These features may have a bigger ability to explain the variance in the training data. Feature engineering also influenced by hypotheses generation. Good hypothesis results in good features but I need to invest a quality amount of time.
* Feature selection - I will select useful features based on some metrics like domain knowledge - selecting a feature/s which may have a higher impact on target variable, visualization - visualize the relationship between variables which makes feature selection process easier, and principal component analysis - it helps to represent the training data into lower dimensional space.
* Algorithm tuning (parameter tuning) - machine learning algorithms is driven by parameters. It majorly influences the outcome of learning the process. For me, parameter tuning is more than an art compare to science.
* Apply ensemble methods - applying Bagging, and boosting to combine the result of multiple weak models and produce better results. Ensembling methods help a lot to improve the accuracy of the model. They are generally more complex than traditional methods.
* Multiple algorithms - some algorithm are better suited to a particular type of data set than others. Applying all relevant models like classification, regression, clustering models and check the performance.
* Validation set - for this project proposal I only used training and testing data. Using validation set is one of the most important concepts in data modeling. The validation set is typically a leave sample on which you do not train the model and test the model on this sample before finalizing the model. Cross-validation method is used to achieve more generalized relationships.

Take note that feature engineering and selection are two different process. In feature engineering, it enables you to build more complex models using raw data only while feature selection will help you to limit these features to manageable number. It also reduces the risk of overwhelming the algorithms and also trims down computation time since you won’t perform as many data transformations.

**Summary**

In this section, I summarized all the findings that might help to manage overtime pay rates or awards effectively:

Award Rules for Food Manufacturing

*Award (OT15)*

1.1 15% of overtime pay on Mondays to Thursday between 6:00 - 18:00.

1.2 15% of overtime pay on Fridays between 14:00 - 23:00.

*Award (OT20)*

1.1 20% of overtime pay on weekdays (morning) from 6:00 and after 18:01.

1.2 20% of overtime pay on weekdays (afternoon) between 12:00 and 7:00 (the next day).

1.3 20% of overtime pay on weekends with a total of at least 4 hours and 36 minutes of working hours.

*Award (OT25)*

1.1 25% of overtime pay during public holidays.

*Award (C150)*

1.1 150% of overtime pay on weekends with a total of at least 3 hours of working hours.

*Award (C200)*

1.1 200% of overtime pay on weekends with a total of at least 4 hours and 35 minutes.

------------------------------

*Link to project proposal:* https://github.com/CEdanoJr/TandaAwardLearner

*Model (algorithm) used:* Softmax Regression with Adam optimizer algorithm.

*Prediction accuracy*: 86%