Machine Learning Engineer Nanodegree

Capstone Proposal

- Yash Singhal
  31 October 2017

Definition

1. Project Overview

The main motive of the project is to accurately calculate whether a person is employed
In a particular field is likely to quit his job depending upon the circumstances or Situations related to it. If Yes than which of the characteristics surrounding his job Motivate the person the most in taking the Decision
For this problem a Featured dataset is Used which have Observation regarding The person's on some Features(Salary no of promotions etc) and wether he left the job or not . The algorithm will train on the Dataset and will Form Model based On the Rules learned from the set which can be thus used be used by Companies
The Dataset Link is
https://www.kaggle.com/ludobenistant/hr-analytics

Ensemble Classifiers are one of the Most Accurate Classifiers Out there for MultiClass Classification

.Here is A paper by Oregon State regarding it

- http://web.engr.oregonstate.edu/~tgd/publications/mcs-ensembles.pdf

`

The Academic Paper Describing the Random Forest Classification and Working on A Dataset

- http://www.bios.unc.edu/~dzeng/BIOS740/randomforest.pd

2. Problem Statement

Giving the Circumstances surrounding the job (Satisfaction level, no of hours)

Which Person would likely to leave the Job and out of all the Circumstances Surrounding the job which Circumstance motivate the people the most in leaving Their Job. Thus the Problem is classic case of Binary Classification where the Given set of observations our goal is to predict will the person likely to quit his job

The input Observations in the Dataset are **Satisfaction Level, Last Evaluation Number Projects, Average_Monthly_Hours, Time_Spent_Company,Work Accident , promotion Last 5 years, sales and Salary**

The output Observation or Labels would be 0 or 1 (1 Stands the Person likely to Quit his Job) . The Given Problem is the Classic Case of Binary Classification Upon which depending upon the input features and the rules the Model Would have formed on Training would output either 0 or 1 .

Lastly we also get to look which Feature or circumstances out of all circumstance The model is giving most importance in Classifying (High Level View of the Rules) in the Form of Ranking

3 Metrics

To calculate the Final results the precision would be my option which will tell how

Accurately i classified my Test as well as Holdout Dataset. I choose Precision As a metric because it is a problem of simple Binary Classification and we are Only concerned whether the Model has correctly identified the Class out of Two Classes on which the Sample could have Belonged . The precision is intuitively the ability of the classifier not to label as positive a sample that is Negative . The precision as a metric works very well for simple Classifications

Analysis

1. Data Exploration

The link of the dataset is http://www.kaggle.com/ludobenistant/hr-analytics
The Following Featured dataset from Kaggle has about 16000

‘

Observations and about 9 features and One Output Label
The result or output variable would simply be the binary Classification that the Person will Survive depending upon the Circumstances Surrounding the Job

The input Observations in the Dataset are **Satisfaction Level, Last Number Projects, Average_Monthly_Hours, Time_Spent_Company, Work Accident , promotion Last 5 years, sales and Salary**

Apart from Sales, promotion last 5 Years , Salary, Work Accident (Which are Categorical) all seems to Be Continuous

- Satisfaction Level
  The Satisfaction Level of a person is measured from 0 to 1 Whether he is satisfied with the position he is in the company

- Promotion Last 5 years
  The total no of Promotion that a person who was employed Had in the Company

- Time_Spent_Company
  The years in total which the Person was Employed in Particular Job

As per our Data Exploratory Findings there are no outliers in the Continuous Feature of the Dataset . Apart from the Working hours All the Features concerned have low mean and valid Std with satisfaction Level having a high standard devaiation in comparison of about 0.24. This Is expected as there is no definite Standard Measurment with regards to It and people have different ways to measure it

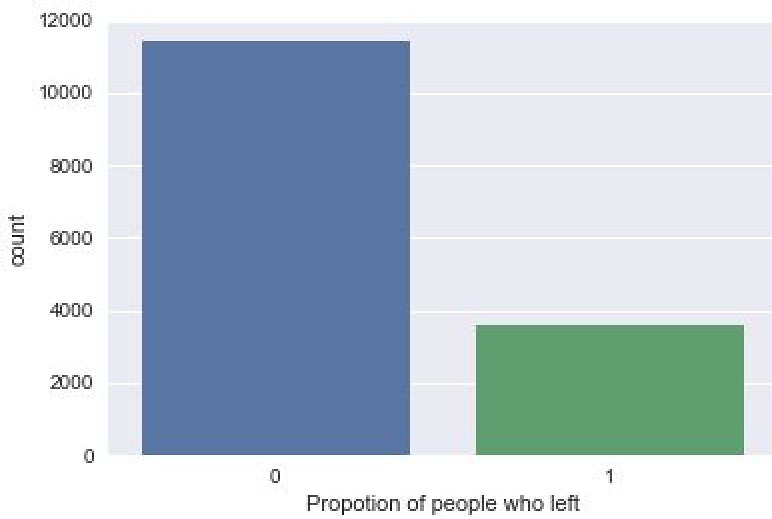
The Working hours has the standard deviation roughly of about 2 Days Its an interesting Finding and belief how much a person need to work harder than some other person

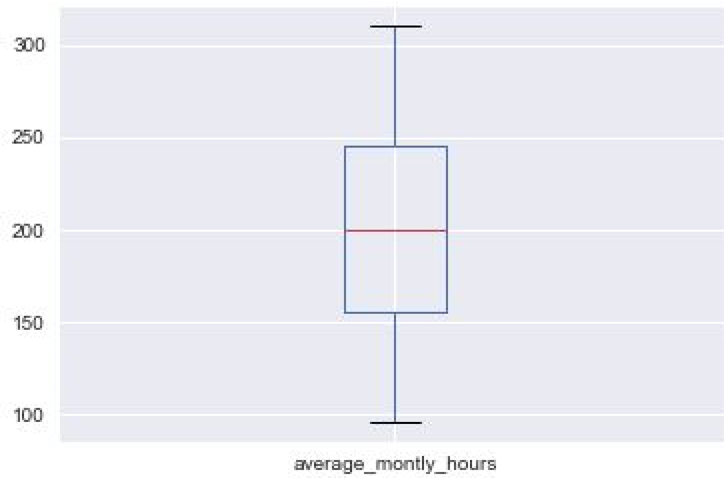
One of the Main issues with the Said Dataset was the Immbalancing of Target Variable which is about 0.24 . That corresponds to about 3600 Positive Responses and rest negative.

NONE OF THE CONTINUOUS FEATURE WAS OUTSIDE THE CONFIDENCE INTERVAL
OF 68 PERCENT WHICH IS INDEED A RIGHT POSITION TO WORK

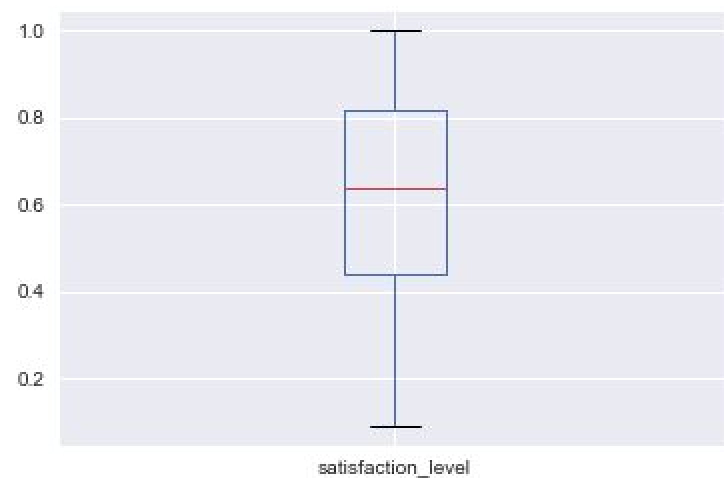
2. Exploratory Visualization

I used couple of Visualization for my dataset using Seaborn

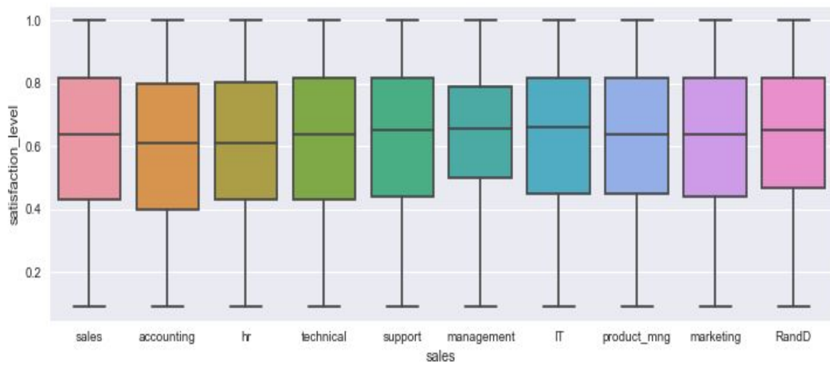




300
250
200
150
100
average_montly_hours



1.0
0.8
0.6
0.4
0.2
satisfaction_level



- A CountPlot is used to calculate the balancing of the target Variable which indeed found to be highly imbalanced. Fortunately Random Forest Classifier is pretty Robust To these and Have a virtually No Effect

- A Boxplot is used to Visualize the Relative Range within Each of the Data Lie and thus to detect Outliers. Fourtanely No Outliers was detected in most of the Continuous Features Of the Dataset

  As I said Boxplot works very well in Identifying The Outliers And detecting the Abnormality.

  For this Purpose we sketch a Boxplot for each of Continuous Variable and we got no Outliers in any of them indicating The Values lies within the Confidence interval of 68
  A box Plot was also made for each of the Continuous Feature specific to sales and we got Almost Uniform range of values

- A KDE plot shows the normal distribution of Continuous Feature data which is optimal for Machine Learning

3 Algorithms and Techniques

- **Decision Tree will be used As a Benchmark Model. The Inhert Working of Decision Tree is similar to the Working Of Random Forest except of few Differences and thus can Be served as a Benchmark Model**

- The Random Forest Classifier is used to Model the Learning It is because so a Random Forest belonging to the Ensemble Classifier is helpful when there are lot of features are need to Be considered as the inherent features of being these they will Automatically filter out the irrelevant features

  The Random Forest Classifier also helps when there is Imbalancing of the Target Variables and is found to Be pretty Robust against these

- Grid Search Cv automatically fine tune the parameters of the Model and helps in returning the tuned model which has a maximum Score against the input Dataset

- One hot Encoding is used to manage the Categorical Dataset by making a separate Column for each of the Values found in the Categorical Column. This may increase the dimensionality but it is way helpful for sklearn

DECISION TREE

The Decision Tree in Layman Terms could be Best attributed As the Computer or Model asking a series of Yes Or No questions and based on the Answers classifiying the Label into one of the

Possible Labels

Of course these type of Questioning won't get you much far if you Kept asking question which might be Unique to specific Category. For Asking question you need to Frame the questions which Would give the Best Split for Example in Classifying an Object Asking a question that whether object is living or Non Living would Be a good start as opposed to a question whether the object name Is Anthony or something.

Decision Tree does this automatically It determines the best Observations that would give the best split asking that question First and Framing the next Question accordingly

WHY RANDOM FOREST ??

It Can happen in Decision Tree Training on Subset of Whole of Dataset would return a Feature which would be too dominating And for Machine Learning that is a bad thing

Random Forest can be simply thought of as a Optimised Version of Decision Tree in which a Multitude of Decision Tree is Produced Where each Dtree is trained on Random Subset of samples and Features

Afterwards all the Results of Decision Tree is Combined to produce A final Tree which serves as a Model

Random Forest helps in averaging down the Dominating Feature And has the beifit as it is an amalgamation of several Decision Tree We can argue all Features and Samples are well considered in Making the Final Book of Rules for Classifying

4 Benchmark

Decision Tree will be used As a Benchmark Model. The Inhernt Working of Decision Tree is similar to the Working Of Random Forest . The random Forest could thus be Regarded As the Optimized Version of Decision Tree and Thus Dtree serves As a Valid Benchmark Model

The Model would be tested on the Holdout set which would be separate from test dataset with tuned parameters

Methodology

1. Data Preprocessing

I only used One hot as a form of Feature Transformation to help Sklearn to handle categorical Values. There was no missing value And no outliers as evident by the plot . While the Target Variables are Indeed highly imbalanced Ensemble Classifier are pretty robust in' Handling these

2 Implementation

There were Several Techniques that i used for solving the Problem

- Implementing One hot encoding on categorical Features of The dataset to make the sklearn job easy for handling these Type of data.

  The One Hot Encoding is implemented by using the get dummies

Function of pandas. The function takes a Dataset as a input
And returns a One hot encoded Version of Dataset

- Using Train Test Split the data into 60 20 and 20 which corresponds To training testing and holdout dataset respectively.

  Train Test Split was done using the sklearn train test Method Which takes a Features Dataset and a Labels or Output Array Dataset corresponding to each of the Observations In the Feature Dataset

- Classifying using the Decision Tree as a Benchmark Model. An Object of Decision tree is first Instantiated and a train subset Of Dataset is then passed as an argument to the Object Fit Function

- Classifying using Random Forest. Similar to the Above an Object of Random Forest is created and training subset of Dataset is passed To the Object Fit Function

- Grid Search Function is used to Fine tune the Parameters. The Model is tested on test Dataset using Precision as a metric . The Function takes classifier , a score Object and the parameters Dictionary as the argument. The parameter is in the form of Dictionary which have the Arguments that need to be tuned

  - **n_estimators** : integer, optional (default=10)
    The number of trees in the forest.
  - **min_samples_split** : int, float, optional (default=2)
    The minimum number of samples required to split an internal Node

- **min_samples_leaf** : int, float, optional (default=1)
  The minimum number of samples required to be at a leaf node

- Best feature attribute is then selected and value is predicted
  Of the Holdout Dataset
- Sklearn metrices contains the Classification report which takes 2
  Arguments the predicted Value and true Value which returns the
  Precision of the Model

- **One of the issue that is recurrent in these type of Classification was Checking the Consistency of Your Dataset. Fortunately the Dataset That was Chosen was pretty much Balanced thus Apart from Hot encoding the Categorical Features I cant say that it took Me longer in any of the Sections. Part of the Credit also Goes to my Random Forest Classification which handled My imbalanced Target variables robustly which would have Been a problem if it wasn't for my Ensemble**

3 Refinement

Grid Search Function is used to Fine tune the Parameters. The Model is tested on test Dataset using Precision as a metric . The Function takes classifier , a score Object and the parameters Dictionary as the argument. The parameter is in the form of Dictionary which have the Arguments that need to be tuned

- **n_estimators** : integer, optional (default=10)
  The number of trees in the forest.. The model is tested
  Within the range [10,15,5,20]. It is basically the no of Trees

the

Random Forest will make Before Combining

- **min_samples_split** : int, float, optional (default=2)
  The minimum number of samples required to split an internal
  The value is tested in the range [2,3,4]. In simple language the
  Minimum no of observation that should be before making the
  Decision

- **min_samples_leaf** : int, float, optional (default=1)
  The minimum number of samples required to be at a leaf node
  The Value is tested in the range [1,2,3]. In simple
  Language keep on splitting until there is specified no of
  Samples

As said The Function will output the best tuned Parametric Model For the classifier

Results

1. **Model Evaluation and Validation**

The Final Model Being Choose has the 97 accuracy on the holdout Dataset which is awesome. A separate 3 splitting is done on the Observations so that after fine tuning the parameters on test Dataset it can be independently verified on the holdout set so as to Prove our test data generalize well to the real world

min_samples_split=4
min_samples_leaf=1
n_estimators=20

These were the Final Parametric Value of the Model . A 20 Estimator for the model seems like a valid Choice given a large No of Feature we have to consider

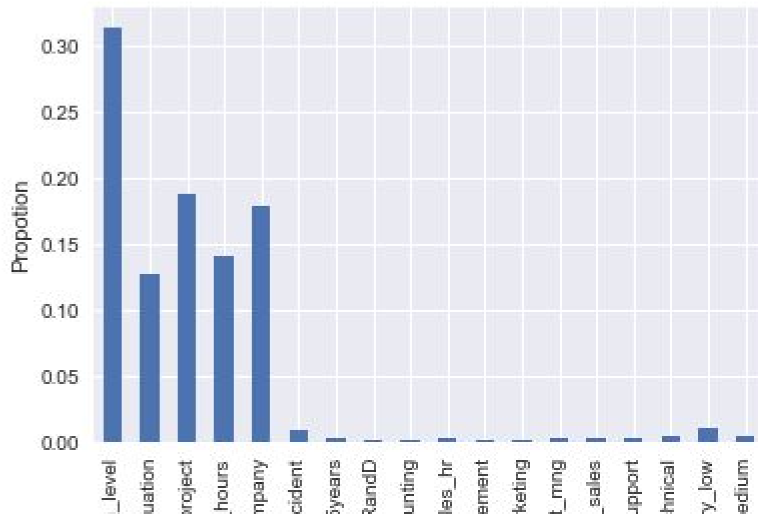
The min samples at the leaf indicate all data has been Well considered

2 Justification

The Benchmark model result is on par with our Finely tuned model While we didn't beat the benchmark model in terms of the precision , the precision score is high enough and High Precision Similar Score in both the benchmark and Our Finely tuned model could be attributed To Well represented and Clean Dataset and absence of over Dominating Characteristics in the Dataset

Conclusion

1. Visualization



A bar Plot is drawn which shows the Importance of various Feature in our final Dataset. The Bar Plot is made possible By the Random Forest feature importance attribute. According To the Graph satisfaction level was of the most importance Immediately following is surprisingly Last evaluation Done

2 Reflection

To conclude Given the Dataset we First Cleaned the Dataset and Checked for any abnormality . After that we encoded the Categorical Features of the Dataset . After That we Trained the Model against Against Random Forest and Tuned its Parameters To get the Best Fit. Once we were satisfied with our model we Checked the Features which were most important to our final Trained Model.

Surprisingly as per Findings expectedly While Job satisfaction Expectedly remained the top Priority. It was surprising to see the Feature last importance showing its appearance in top 5 indicating

The Employee always wants some sense of Reinforcement from the Company or from the place where he is currently Doing his job

3. Improvement

ADA Boosting Classifiers(Boosting Classifiers) can improve Upon the Random Forest since Random Forest Mainly Reduces the error by reducing the Variance while Boosting Classifier reduces error by reducing the fixed bias and to some Extent the Variance

Since our Dataset dosent have a large Variance to begin with The Boosting Algorithm could have perfomed well than RF by Most importantly reducing Bias

More In depth Look
http://statweb.stanford.edu/~jhf/ftp/trebst.pdf