**HyperparameterDB**

***Sharwari Makarand Gothe, Sharvari Karnik, Cassian Gonsalves***

**{gothe.s, karnik.sh, gonsalves.c}@husky.neu.edu**

**INFO 6510, Spring 2019, Northeastern University**

***Abstract—In statistics, hyperparameter is a parameter from a prior distribution; it captures the prior belief before data is observed. In any machine learning algorithms, these parameters need to be initialized before training a model. Hyperparameters are important because they directly control the behaviour of the training algorithm and have a significant impact on the performance of the model is being trained. Our aim is to find proper hyperparameter with proper tuning for our dataset which would help the database team in modelling the database schema in an efficient way. We would create H2O models for this dataset for getting proper hyperparameters. In this paper the dataset is on Bank Marketing in which the goal is to predict if the client will subscribe a term deposit.***

***Keywords—Hyperparameter, H2O, Bank Marketing, Prediction, Classification.***

##### Introduction

# **1.1 Background**

Hyperparameters are variables that we need to set before applying a learning algorithm to a dataset. In machine learning scenarios, a significant part of model performance depends on the hyperparameter values selected. The goal of hyperparameter exploration is to search across various hyperparameter configurations to find the one that results in the optimal performance. The challenge with hyperparameters is that there are no magic number that works everywhere. The best numbers depend on each task and each dataset. The hyperparameter database to be developed as a part of this project is an open resource with algorithms, tools, and data that allows users to visualize and understand how to choose hyperparameters that maximize the predictive power of their models. Phase I of the project involves selecting a unique dataset containing predicted target variables, hyperparameters, meta-data etc. by running different models (with varying hyperparameters) on it using H2O.

Hyperparameters can be divided into 2 categories:

1. Optimizer hyperparameters

* They are related more to the optimization and training process - If our learning rate is too small than optimal value then it would take a much longer time (hundreds or thousands) of epochs to reach the ideal state
* If our learning rate is too large than optimal value then it would overshoot the ideal state and our algorithm might not converge.

2. Model specific Hyperparameters

* They are more involved in the structure of the model

Currently, the hyperparameter database analyzes the effect of hyperparameters on the following algorithms: Distributed Random Forest (DRF), Generalized Linear Model (GLM), Gradient Boosting Machine (GBM). Naïve Bayes Classifier, Stacked Ensembles, XGBoost and Deep Learning Models (Neural Networks).

**1.2 Dataset**

The data is related with direct marketing campaigns of a Portuguese banking institution. The marketing campaigns were based on phone calls. Often, more than one contact to the same client was required, in order to access if the product (bank term deposit) would be ('yes') or not ('no') subscribed.

* Age (numeric)
* Job : type of job (categorical: 'admin.', 'blue-collar', 'entrepreneur', 'housemaid', 'management', 'retired', 'self-employed', 'services', 'student', 'technician', 'unemployed', 'unknown')
* Marital : marital status (categorical: 'divorced', 'married', 'single', 'unknown' ; note: 'divorced' means divorced or widowed)
* Education (categorical: 'basic.4y', 'basic.6y', 'basic.9y', 'high.school', 'illiterate', 'professional.course', 'university.degree', 'unknown')
* Default: has credit in default? (categorical: 'no', 'yes', 'unknown')
* Housing: has housing loan? (categorical: 'no', 'yes', 'unknown')
* Loan: has personal loan? (categorical: 'no', 'yes', 'unknown')

This is a classification dataset. The project is to determine whether the client will subscribe or not. The target variable is y - has the client subscribed a term deposit? (binary: 'yes', 'no')

##### **2. Data Cleaning and Exploration**

In this task, we are inspecting and auditing the data to identify the data problems, and then fix the problems. Different generic and major data problems could be found in the data might include:

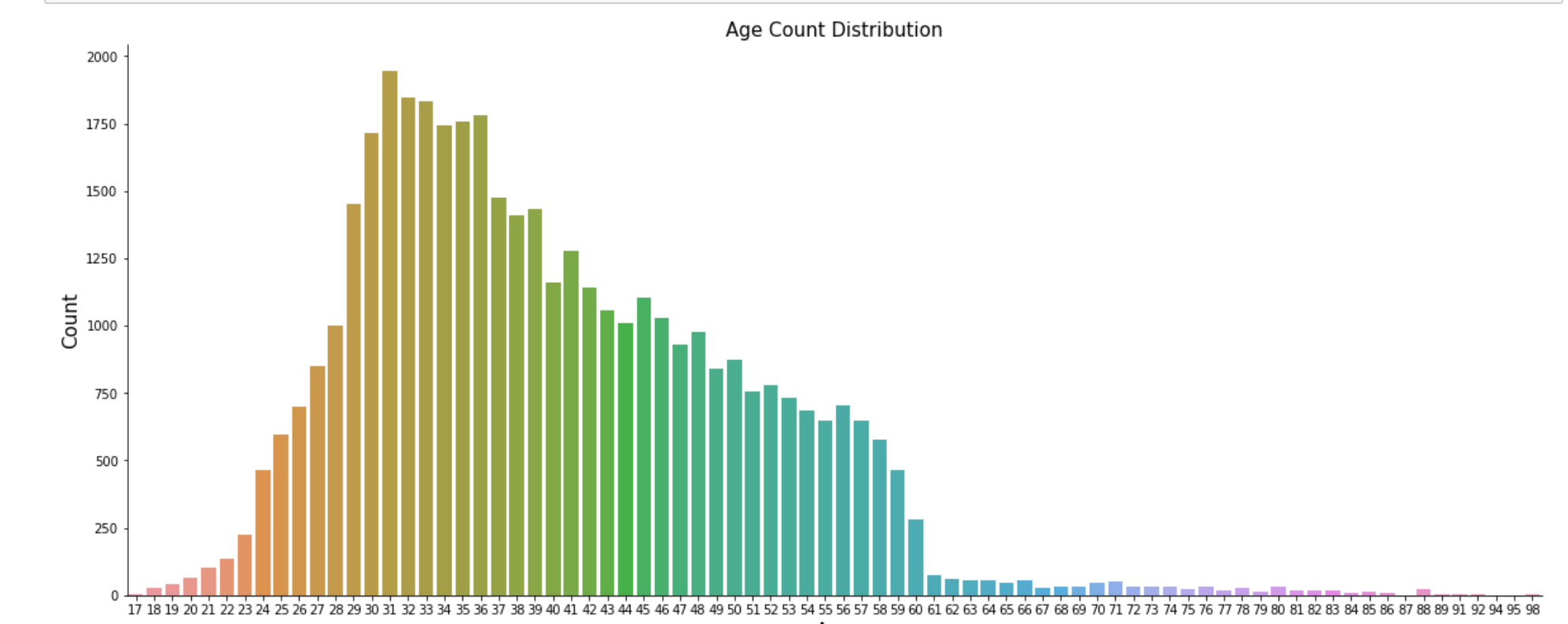
1. Lexical errors, e.g., typos and spelling mistakes
2. Irregularities, e.g., abnormal data values and data formats
3. Violations of the Integrity constraint.
4. Outliers
5. Duplications
6. Missing values
7. Inconsistency, e.g., inhomogeneity in values and types in representing the same data

The following libraries are used throughout the notebook for purpose of data cleaning and visualization

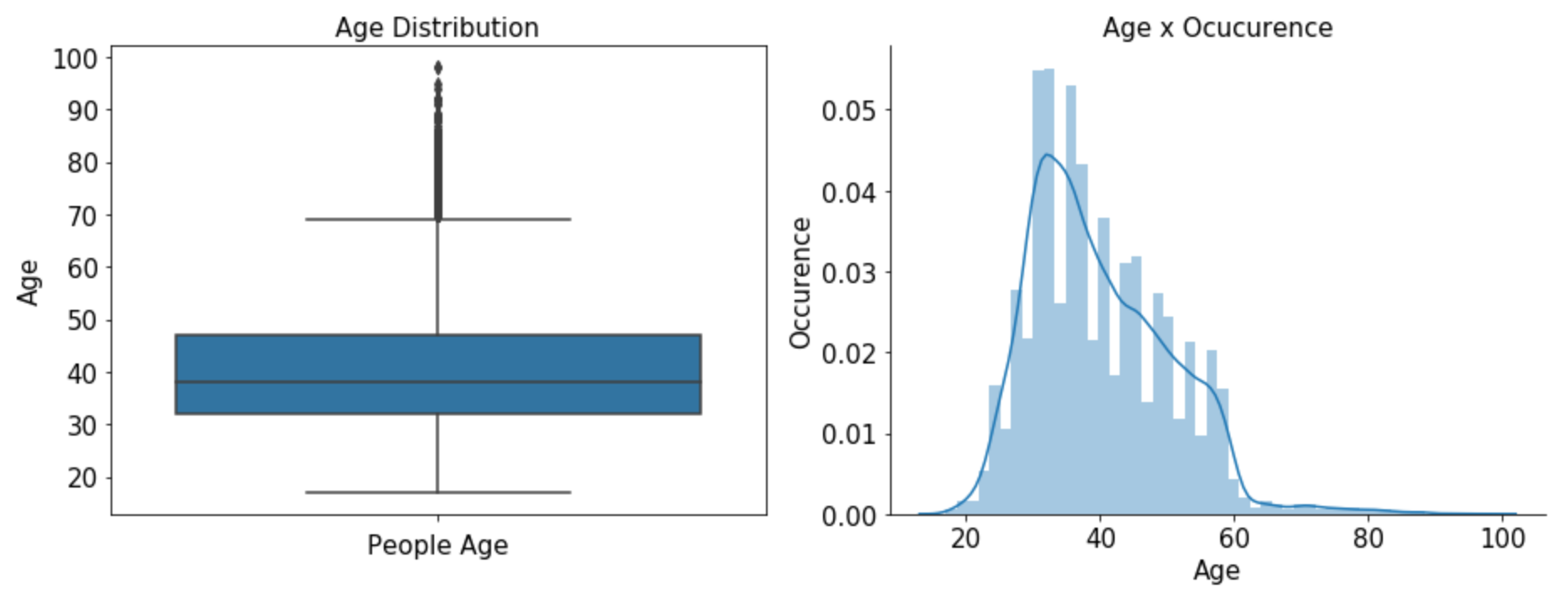


*Libraries used in data cleaning and visualization*

Figuring out some inference based on the age of the clients. Following are few graphs which will help in figuring out the distribution and effect of age on our independent variable.



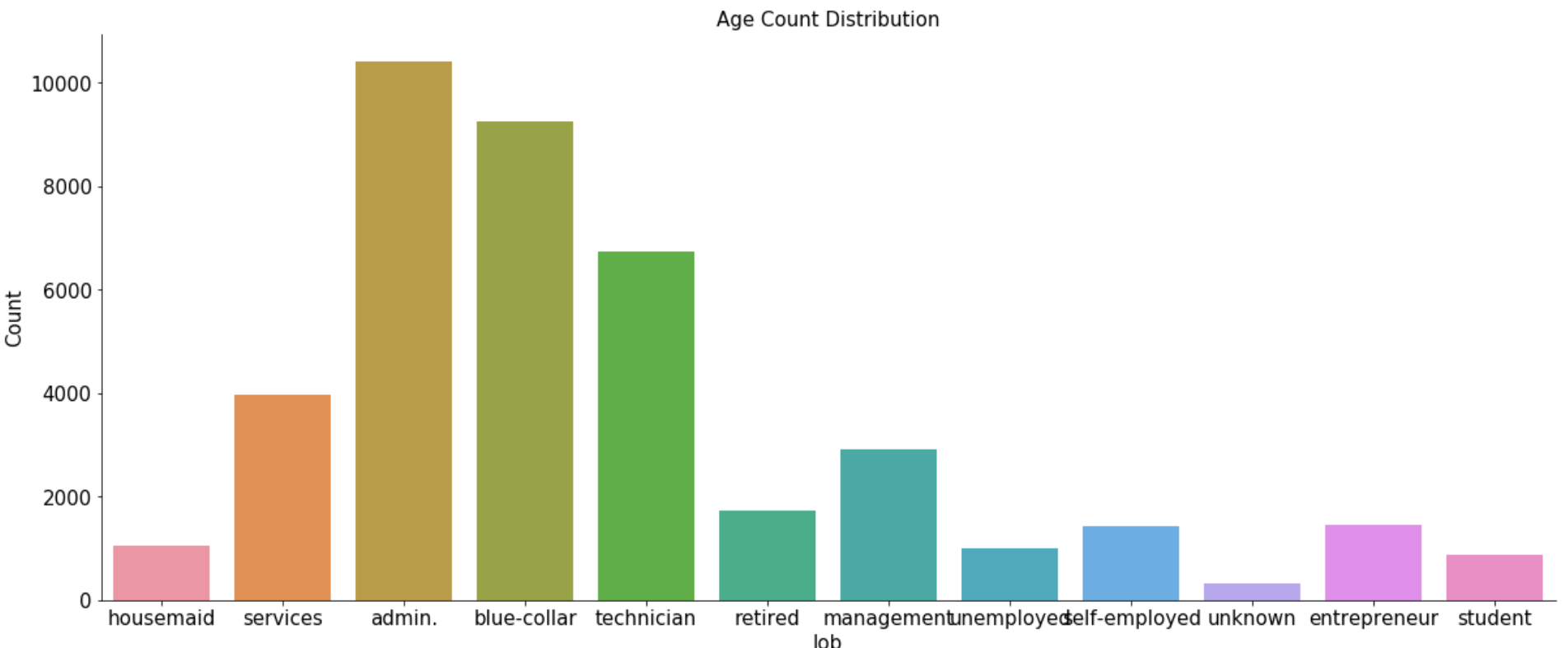
*Seaborn Plot to see the age count distribution*



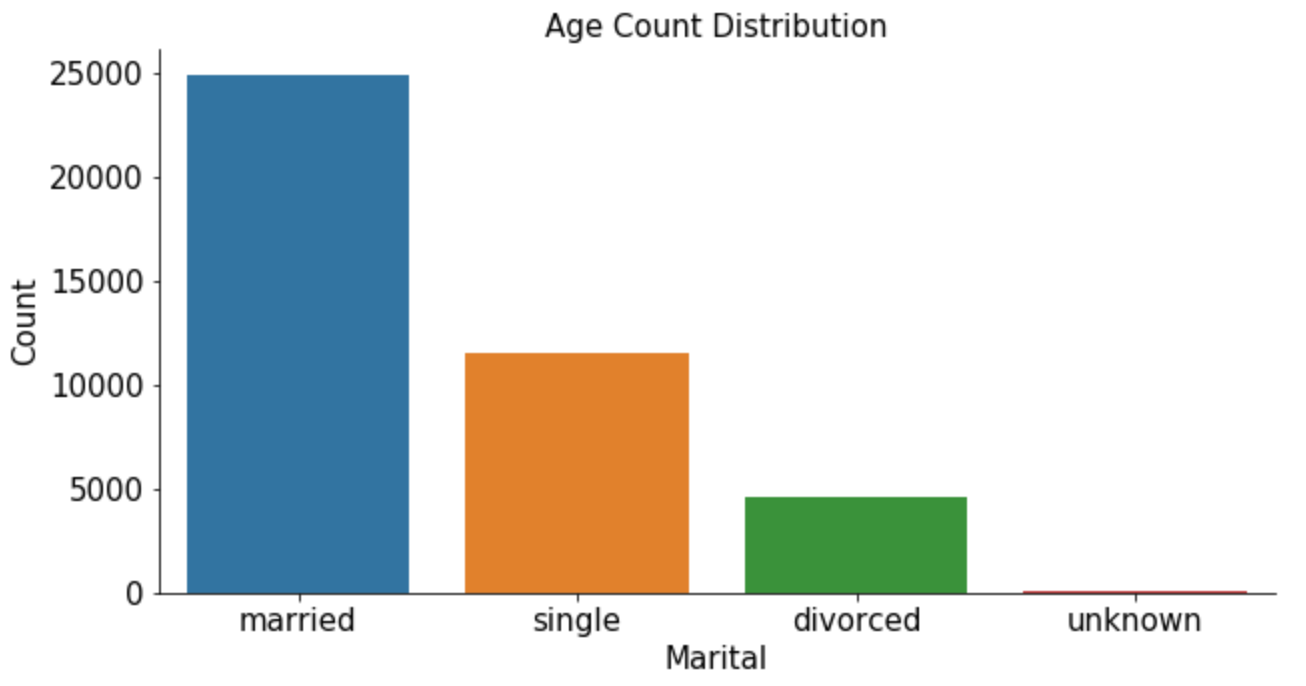
*Age distribution Vs the occurrence count*

Based on the above graphs our conclusion says that in our opinion due to almost high dispersion and just looking at this this graph we cannot conclude if age have a high effect to our variable y, need to keep searching for some pattern. high middle dispersion means we have people with all ages and maybe all of them can subscript a term deposit, or not. The outliers was calculated, so my thinking is fit the model with and without them

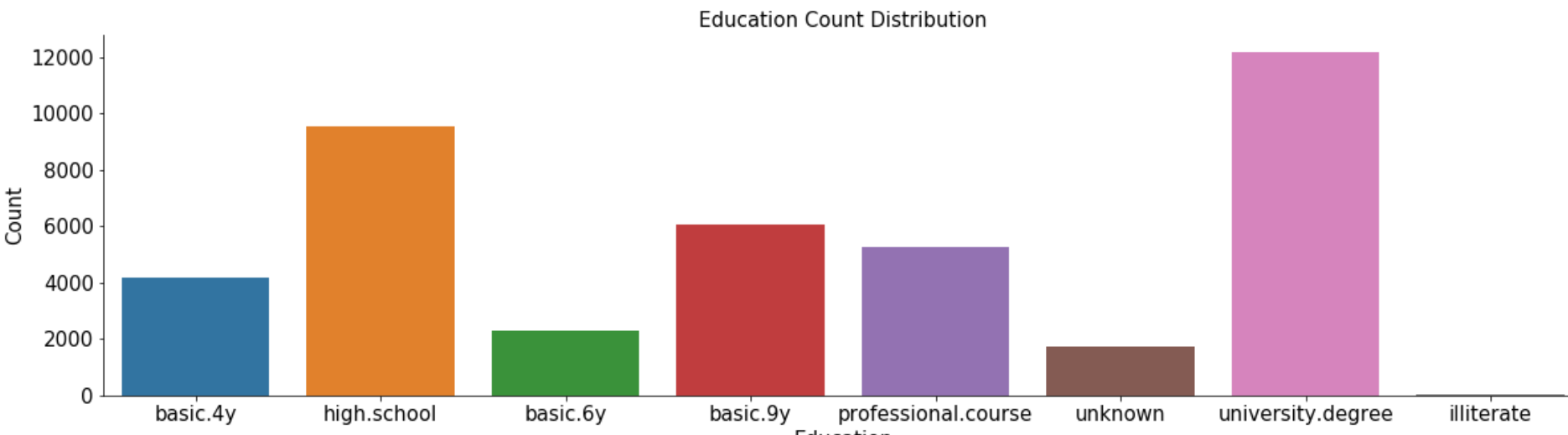
Exploring Jobs:



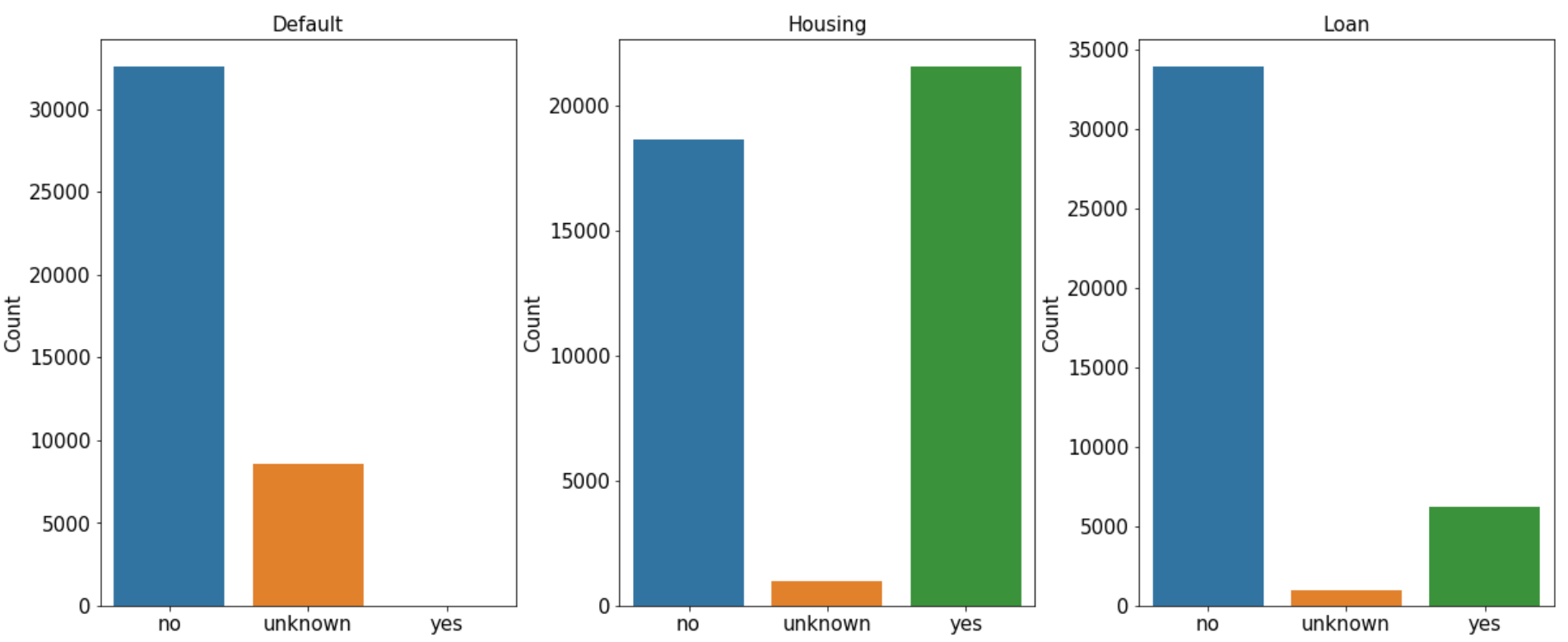
Exploring Marital Status:



Exploring Education data:



Exploring Default, Housing and Loan Data:



The above graphs and analysis shows that The age don’t mean too much, has a medium dispersion and don’t make sense relate with other variables will not tell any insight

Jobs, Marital and Education is probably the best analysis is just the count of each variable, if we relate it with the other ones its is not conclusive, all this kind of variables has yes, unknown and no for loan, default and housing.

Default, loan and housing, its just to see the distribution of people.

##### **3.Algorithm and code source**

H2O is an open source predictive analytics platform for data scientists and business analysts who need scalable and fast machine learning. Unlike traditional analytics tools, H2O provides a combination of extraordinary math and high performance parallel processing with unrivaled ease of use. H2O intelligently combines the following features to solve today’s most challenging business problems:

* Best of breed Open Source Technology
* Easy-to-use WebUI and Familiar Interfaces
* Data Agnostic Support for all Common Database and File Types
* Massively scalable Big Data Munging and Analysis
* Real-time Data Scoring

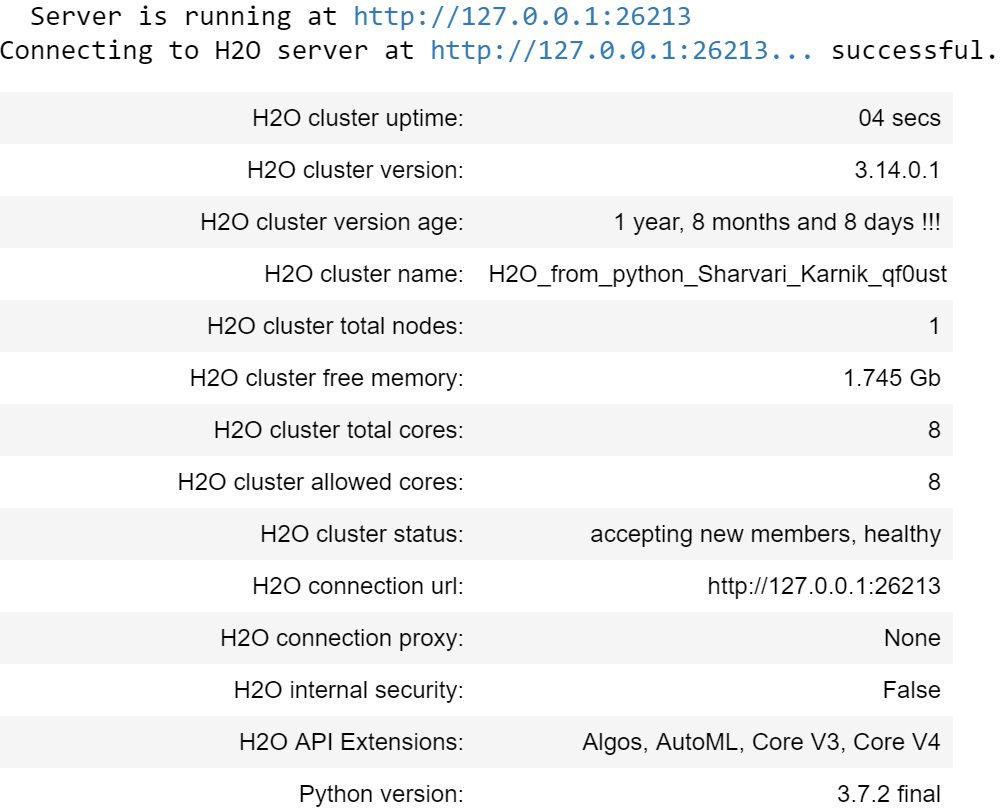
The H2O python module is not intended as a replacement for other popular machine learning frameworks such as scikit-learn, pylearn2, and their ilk, but is intended to bring H2O to a wider audience of data and machine learning devotees who work exclusively with Python.

H2O from Python is a tool for rapidly turning over models, doing data munging, and building applications in a fast, scalable environment without any of the mental anguish about parallelism and distribution of work.

In this project the dataset is cleaned and then passed to H2o to get all the possible models. The models are generated for various runtime. In this project the dataset has been run for 200, 700 and 1500 seconds.

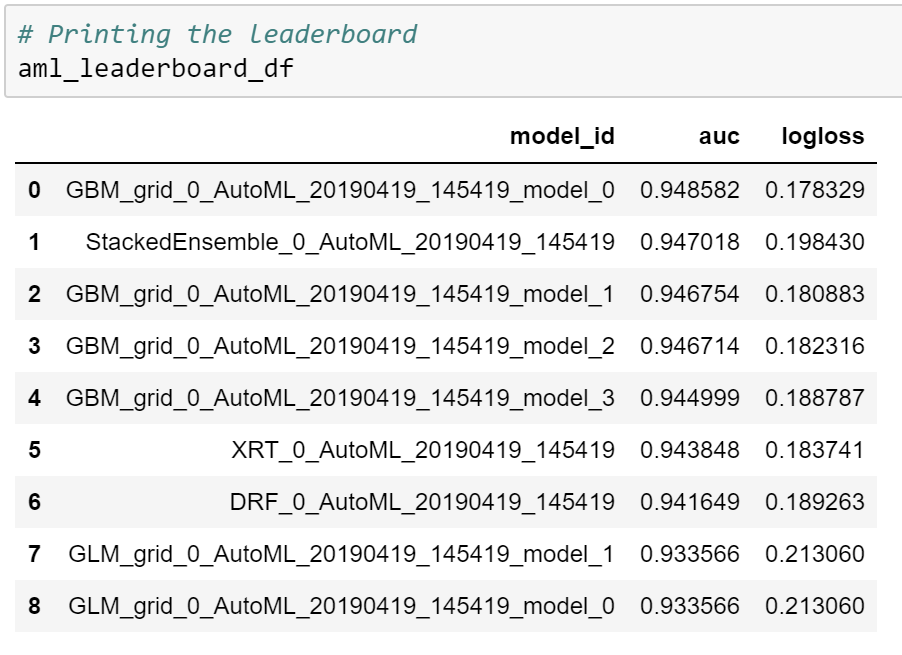
For 200 seconds:

The dataset is passed to automl and H2o connects with its server to generate various models. The log file is generated to save all the logs created in the process. Log files will help to track the actions and the results in the future. Following image shows that the server is running at the given port number and the address at which the H2O server is connected successfully.



As mentioned earlier, each model does not generate separate meta-data. For every run-time a single metadata is generated which is stored in a separate file.

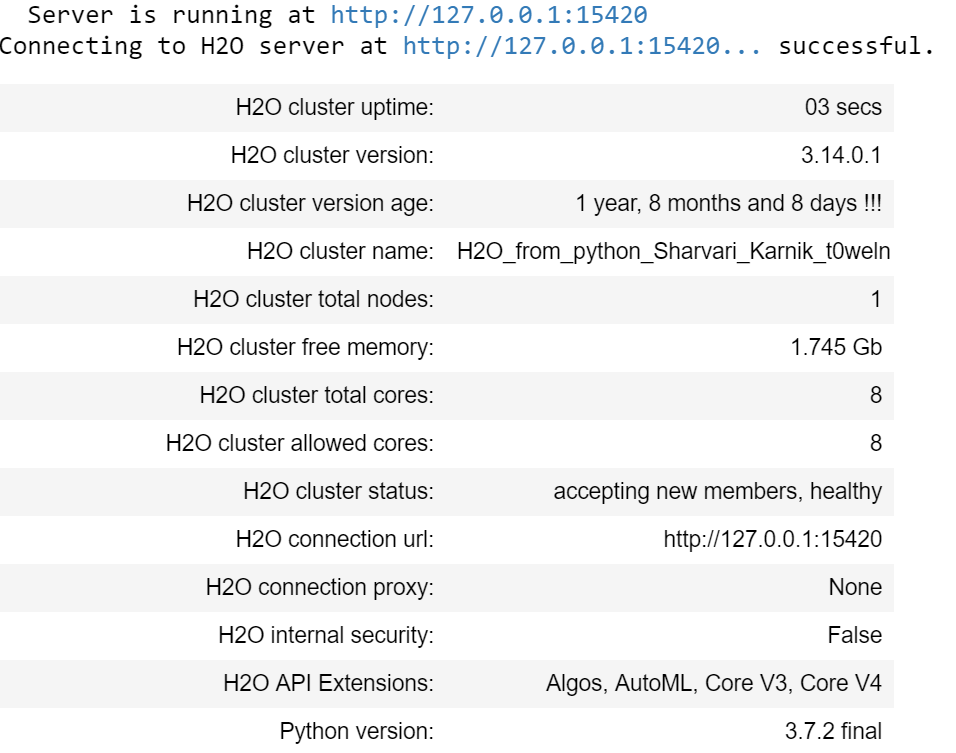
The leaderboard displays all the algorithms used to generate the models. The leaderboard for this dataset for the runtime of 200 seconds is as follows:

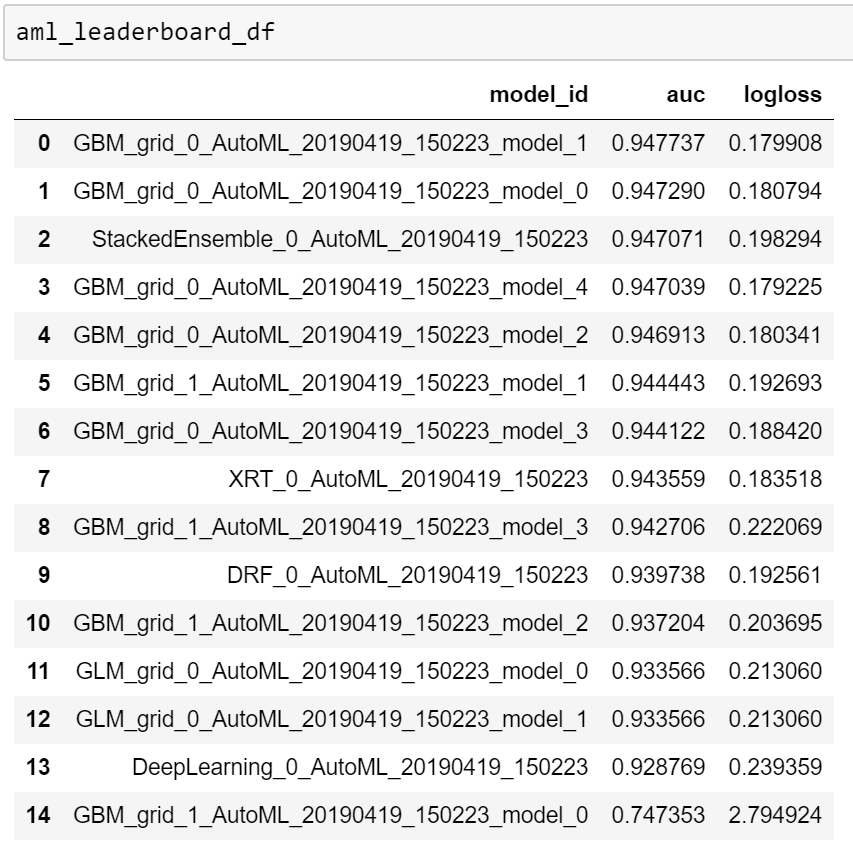


Every model is stored in the database for future reference. The hyperparameters of the best model are found using further analysis techniques.

For 700 seconds

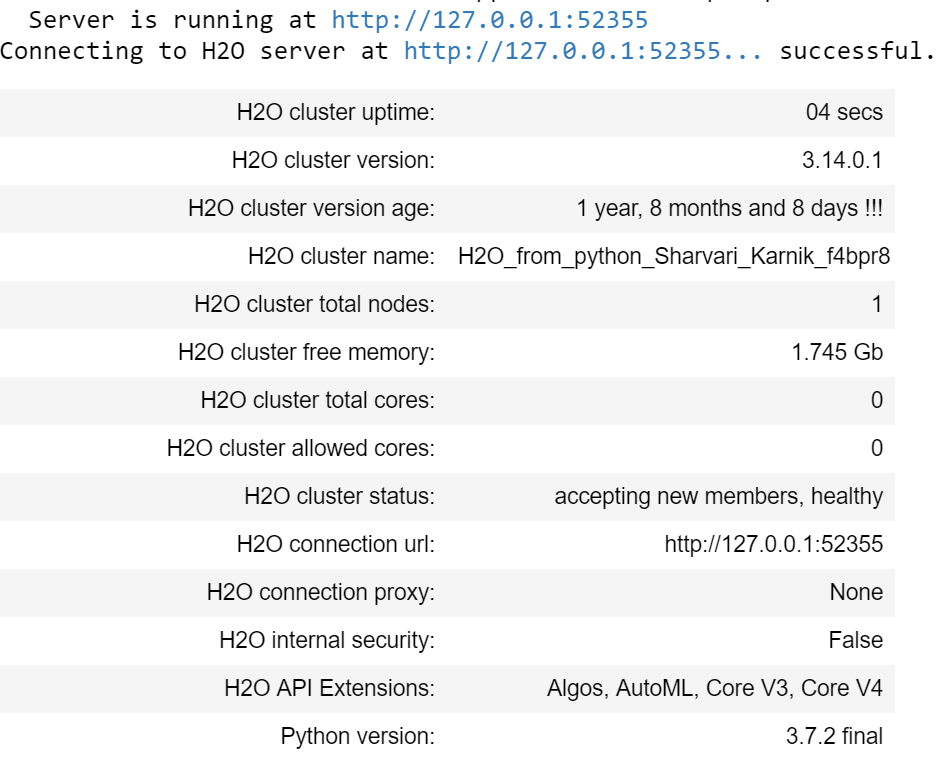
Similar to process followed for 200 seconds, the procedure is again followed for 700 seconds. There is a minute difference in the readings. Following images show the connection with H2O server and the leaderboard generated by it in 700 seconds.

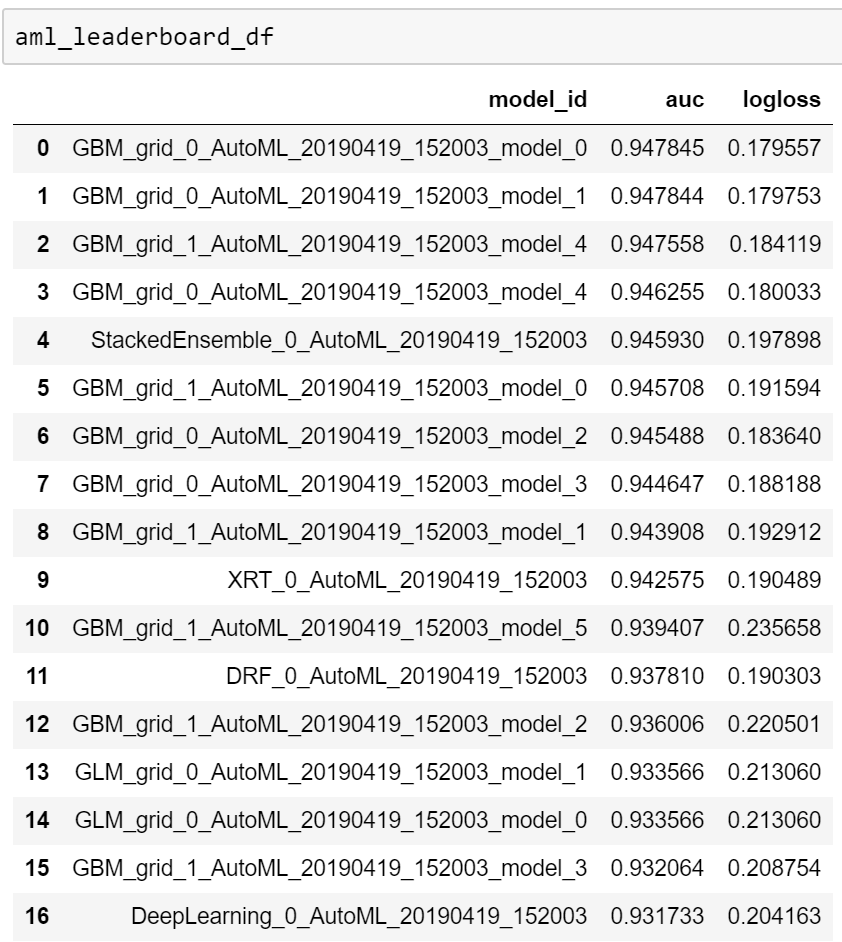




For 1500 seconds

Similar to process followed for 200 seconds, the procedure is again followed for 1500 seconds. There is a minute difference in the readings. Following images show the connection with H2O server and the leaderboard generated by it in 1500 seconds.





##### **4. Result**

##### **5. Discussion**

##### **6. Conclusion**

##### **7. Acknowledgment**

We want to thank Prof. Nick Brown for his constant support and guidance during this project. We would also like to thank Prabhu Subramanium, manager of this project who has helped us on every step and guided us wherever required.

##### **8. References**

[1]<http://localhost:8890/notebooks/Desktop/DSProject/hyperparameter-db-project-ds18/BankDB_200.ipynb>

[2]<https://www.kaggle.com/henriqueyamahata/bank-marketing>

**[3] https://medium.com/@alexandraj777/top-5-mistakes-data-scientists-make-with-hyperparameter-opti mization-and-how-to-prevent-them-767638b245f8 [4]**[**https://towardsdatascience.com/hyperparameters-in-deep-learning-927f7b2084dd**](https://towardsdatascience.com/hyperparameters-in-deep-learning-927f7b2084dd)

[5][**https://www.quora.com/What-are-hyperparameters-in-machine-learning**](https://www.quora.com/What-are-hyperparameters-in-machine-learning)

**[6]https://medium.com/@jrodthoughts/knowledge-tuning-hyperparameters-in-machine-learning-algorit hms-part-i-67a31b1f7c88**