Predicting the Success of Bank Telemarketing

Jin No

March 2019

Table of Contents

The initial idea for my Capstone 1 Project originated from my personal experience seeing "Machine Learning Methods reducing X% in costs" over and over again in annual investor reports and news headlines

The Original Question

1 What does it mean when Financial Services companies say that their implementing AI & Machine Learning to reduce costs?

The Data Source

2 Found dataset online to conduct a data science capstone project related to original question

UCI Machine Learning Repository

Exhibit: Snapshot of data source from UCI Machine Learning Repository

Data Set Characteristics:	Multivariate	**Number of Instances:**	45211	**Area:**	Business
Attribute Characteristics:	Real	**Number of Attributes:**	17	**Date Donated**	2012-02-14
Associated Tasks:	Classification	**Missing Values?**	N/A	**Number of Web Hits:**	842229

Source:

[Moro et al., 2014] S. Moro, P. Cortez and P. Rita. A Data-Driven Approach to Predict the Success of Bank Telemarketing. Decision Support Systems, Elsevier, 62:22-31, June 2014

The Approach

3 Applied Data Science Project Methodology

Dataset from UCI

- **Data importation into dataframe**
- **Data Cleaning/Wrangling**

B Setting up the Data

- **Exploratory Data Analysis**

C Exploratory Data Analysis

- **Statistical Testing of Key Trends**

D Statistical Testing

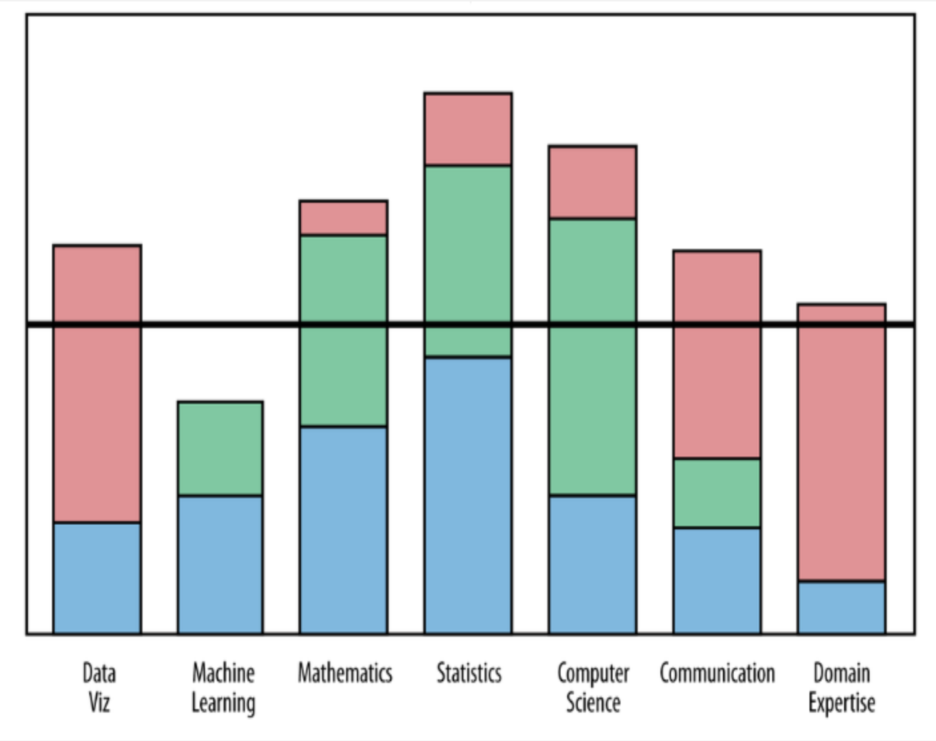
- **Pre-processing of Data**
- **Training and Testing Model(s)**
- **Evaluating Model(s)**
- **Tuning Model(s)**

E Predictive Modeling

In addition to my personal interest in machine learning applications in the financial services industry, I wanted to focus more on gaining experience with machine learning packages and less on data wrangling/cleaning

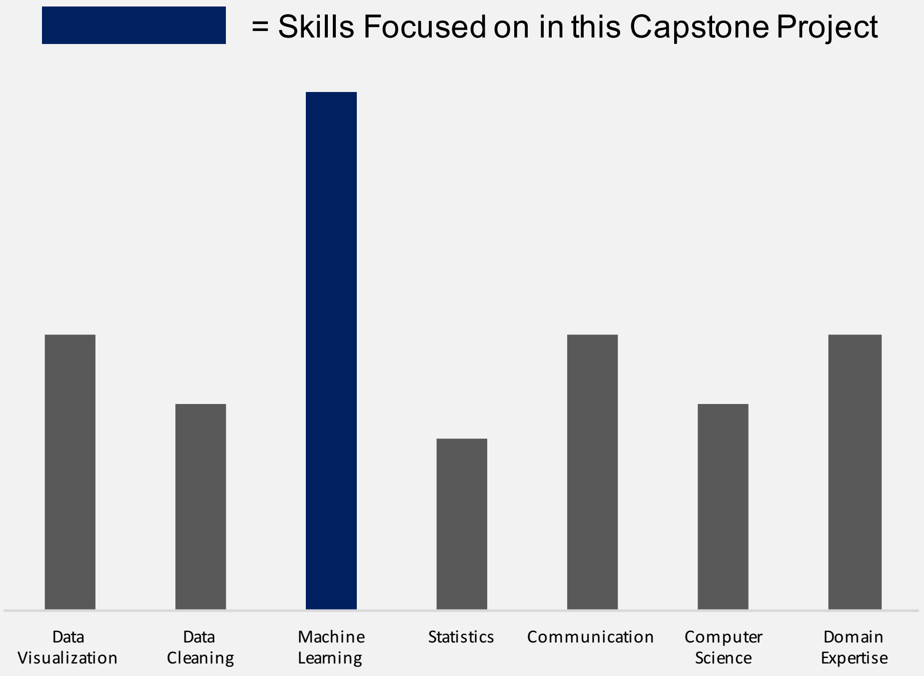
Skills Used in Data Science

Exhibit: Data Science Profile from "Doing Data Science" by Schutt and O'Neil



Skills learned from Capstone 1 Project

Exhibit: Relative Scale of Data Science Skills used in this Capstone Project



Scale is meant to be illustrative. Added Data Cleaning.

Table of Contents

The original dataset was turned into a Pandas DataFrame to allow for easy, pythonic data cleaning and data analysis to guide hypothesis formation…

Data Source and Importation

1 Imported Data into Data Frame

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 41188 entries, 0 to 41187
Data columns (total 21 columns):
age               41188 non-null int64
job               41188 non-null object
marital           41188 non-null object
education         41188 non-null object
default           41188 non-null object
housing           41188 non-null object
loan              41188 non-null object
contact           41188 non-null object
month             41188 non-null object
day_of_week       41188 non-null object
duration          41188 non-null int64
campaign          41188 non-null int64
pdays             41188 non-null int64
previous          41188 non-null int64
poutcome          41188 non-null object
emp.var.rate      41188 non-null float64
cons.price.idx    41188 non-null float64
cons.conf.idx     41188 non-null float64
euribor3m         41188 non-null float64
nr.employed       41188 non-null float64
y                 41188 non-null object
dtypes: float64(5), int64(5), object(11)
memory usage: 6.6+ MB
```

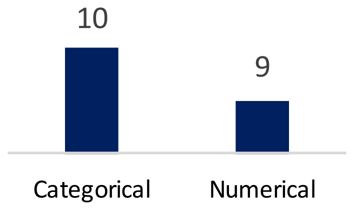
Data Cleaning

2 Removed Variables

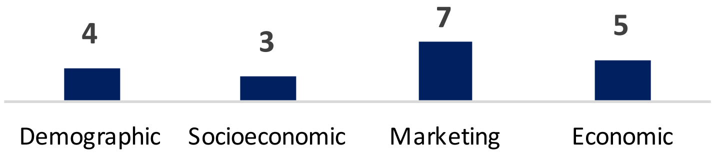
11 - duration: last contact duration, in seconds (numeric). Important note: this attribute highly affects the output target (e.g., if duration=0 then y='no'). Yet, the duration is not known before a call is performed. Also, after the end of the call y is obviously known. Thus, this input should only be included for benchmark purposes and should be discarded if the intention is to have a realistic predictive model.

3 Categorized Variables

Categorical vs. Numerical



Type of Variable



4 Encoded Target Variable

Sample Number	Old Output Variable	Encoded Output Variable
41159	yes	1
41160	yes	1
41161	no	0
41162	no	0
41163	yes	1

…with a majority of the data cleaning efforts spent on pre-processing the data frame to enable usage of popular machine learning libraries (e.g., sci-kit learn)

Data Cleaning (Cont.)

Exhibit: Jupyter Notebook Markdown in Capstone 1

5 Dealing with unknowns

```
admin.           10422
blue-collar       9254
technician        6743
services          3969
management        2924
retired           1720
entrepreneur      1456
self-employed     1421
housemaid         1060
unemployed        1014
student            875
unknown            330
Name: job, dtype: int64
NAs for job : 330
Most frequently occuring value is: admin.
```

```
married     24928
single      11568
divorced     4612
unknown        80
Name: marital, dtype: int64
NAs for marital : 80
Most frequently occuring value is: married
```

= Replaced unknowns with mode for each categorical variable

Exhibit: Jupyter Notebook Cell in Capstone 1

6 Splitting Data Frame into Feature & Target Arrays and Encoding Feature Array using sci-kit learn

```python
#Processing Feature Variables

#Using LabelEncoder
le = preprocessing.LabelEncoder()
df2_le = df2
df2_le["job"] = le.fit_transform(df2_le["job"])
df2_le["marital"] = le.fit_transform(df2_le["marital"])
df2_le["education"] = le.fit_transform(df2_le["education"])
df2_le["default"] = le.fit_transform(df2_le["default"])
df2_le["housing"] = le.fit_transform(df2_le["housing"])
df2_le["loan"] = le.fit_transform(df2_le["loan"])
df2_le["contact"] = le.fit_transform(df2_le["contact"])
df2_le["month"] = le.fit_transform(df2_le["month"])
df2_le["day_of_week"]  = le.fit_transform(df2_le["day_of_week"])
df2_le["poutcome"] = le.fit_transform(df2_le["poutcome"])
```

Used LabelEncoder because most of our categorical features have some sort of "hierarchical order"

Exhibit: Diagram of Dataset Split in Capstone 1

7 Splitting into Training & Testing Datasets

Models to be explained in more detail in Section E – Predictive Modelling

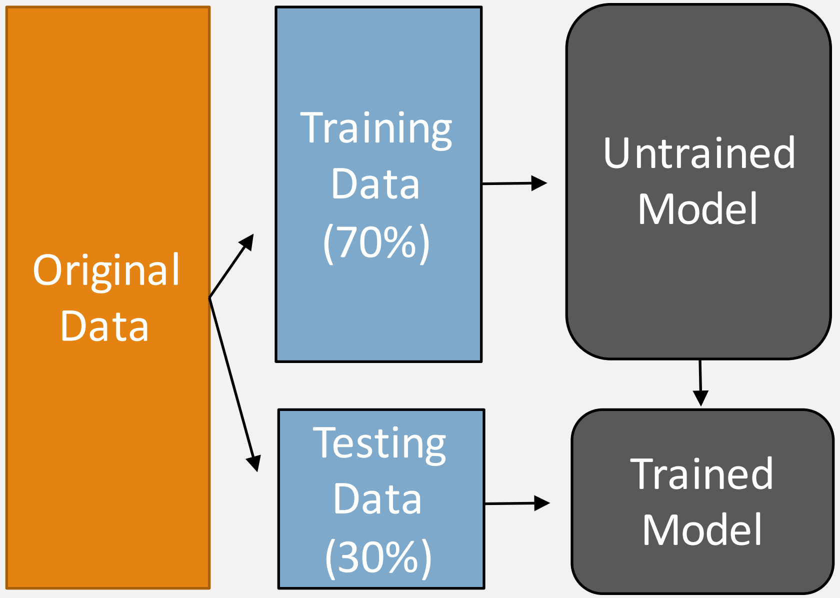
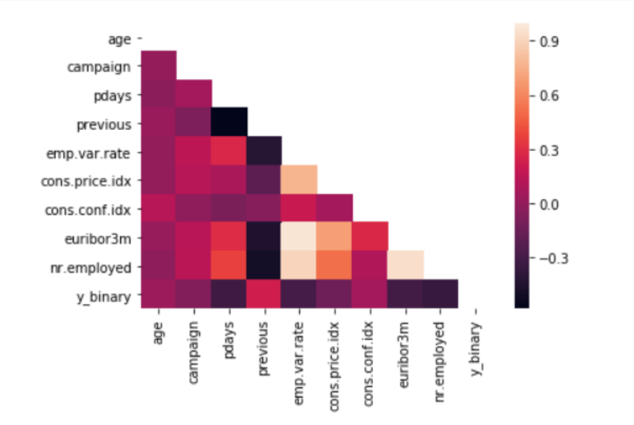


Table of Contents

To gain a better understanding of our data, a heatmap of the un-processed numerical features and bar plots of the un-processed categorical features reveal some insights regarding distributions by outcome variable and correlations amongst other variables

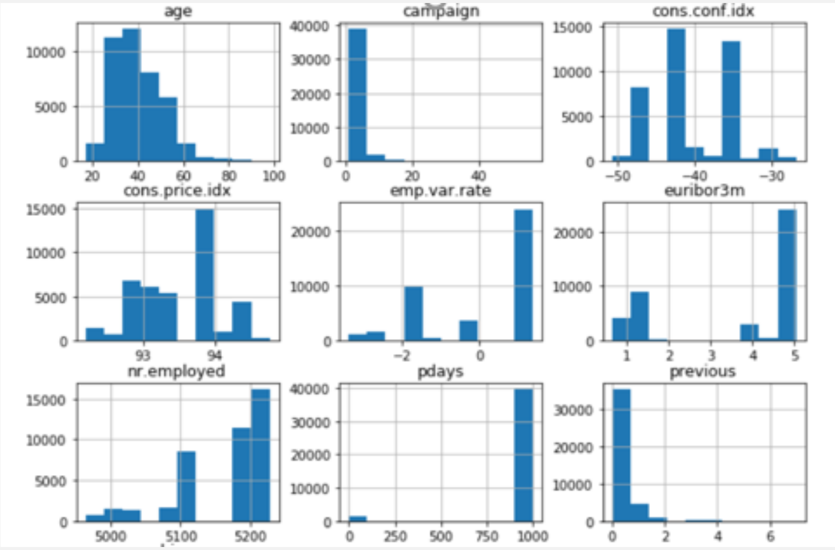
Numerical Variables

Exhibit: Jupyter Notebook Markdown in Capstone 1



Heatmap

- **Purpose:** To identify correlations amongst numerical predictor variables and target variables
- **Outcome:** No significant correlations besides economic variables which seem to be highly correlated amongst each other

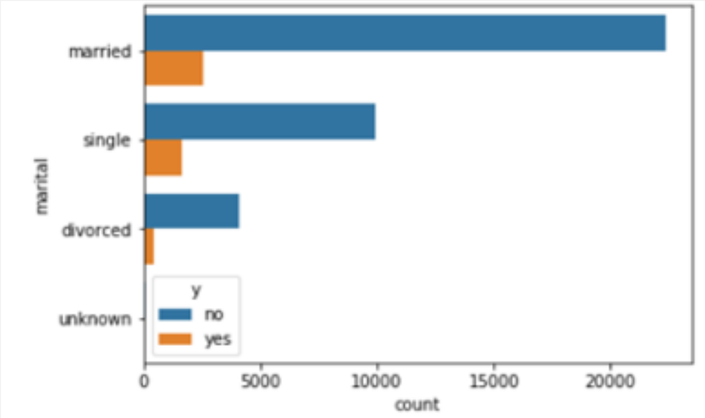


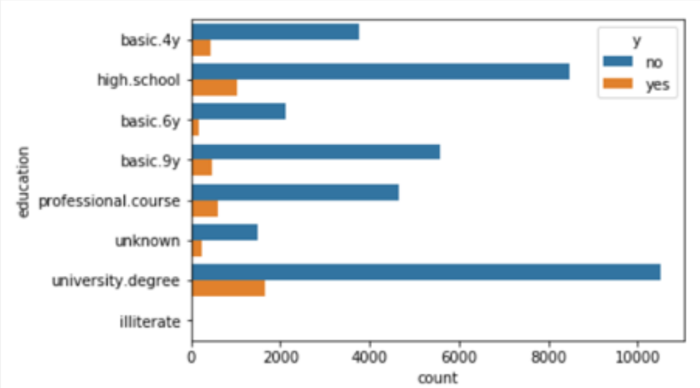
Histograms

- **Purpose:** To identify distribution of all values against a given variable
- **Outcome:** Guidance on how to account for unusual values (e.g. 999 for the "pdays" variable) and/or outliers in future analyses

Categorical Variables

Exhibit: Jupyter Notebook Markdown in Capstone 1





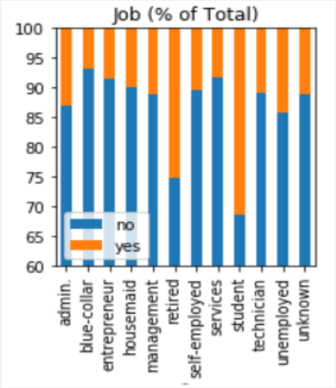
Barplots

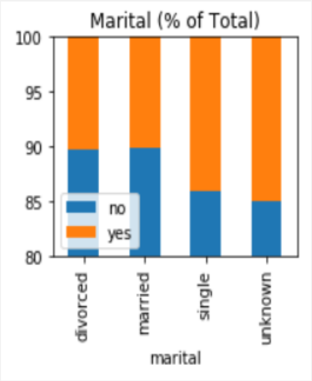
- **Purpose:** To identify distribution of all values by outcome against a given variable
- **Outcome:** Guidance on how to account for unusual values and unknowns (See pg. 7 to see how unknowns were dealt with)

With further exploratory data analysis being done on each type of variable to uncover any unusual trends that will serve as the fact-base for hypothesis formation and hypothesis testing

Exploratory Data Analysis of Feature Array - Demographic Variables

Exhibit: Jupyter Notebook Markdown in Capstone 1



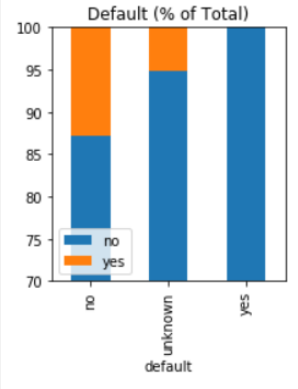


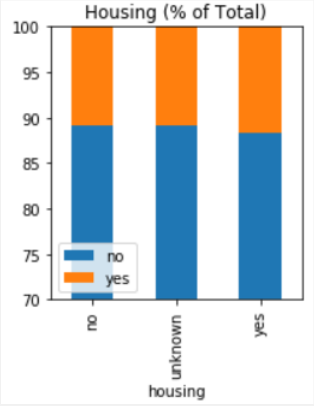
Key Takeaways

- **Certain Jobs and Single People:** Some jobs had a high success rate. Additionally, seems like single people were most likely to purchase a term deposit.
- **Further Statistical Testing:** These two trends will be explored via statistical testing in the next section

Exploratory Data Analysis of Feature Array – Socioeconomic Variables

Exhibit: Jupyter Notebook Markdown in Capstone 1



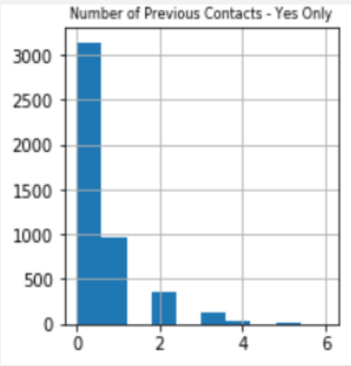


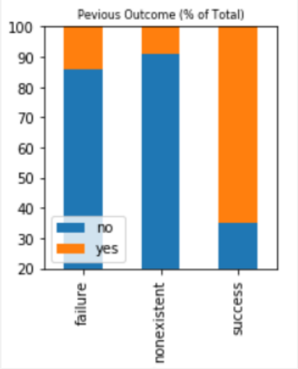
Key Takeaways

- **No more Defaults:** Everyone who defaulted did not purchase a term deposit. We should not waste marketing campaigns on those we've identified as having defaulted on their house
- **Housing Indifference:** Additionally, there is no drastic difference between in success between No and Yes for Housing

Exploratory Data Analysis of Feature Array – Marketing Variables

Exhibit: Jupyter Notebook Markdown in Capstone 1



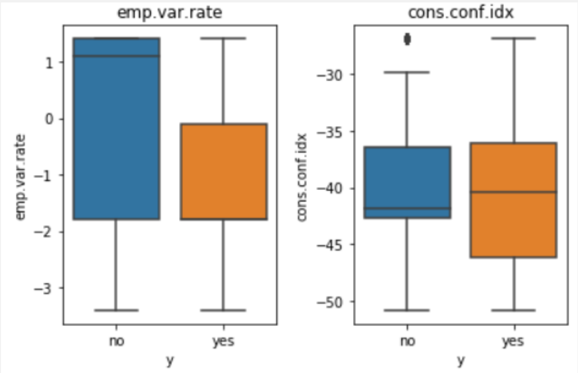


Key Takeaways

- **Previous Contact:** Marketing campaigns were most successful when customer was previously contacted <2 times
- **Previous Outcome:** To no surprise, the marketing campaigns are more successful when targeted to people who had previously purchased a term deposit.

Exploratory Data Analysis of Feature Array – Economic Variables

Exhibit: Jupyter Notebook Markdown in Capstone 1



Economic Indicators will be more relevant once we get to Section E – Predictive Modelling

Table of Contents

Of the key trends observed from our exploratory data analysis, 3 trends of particular interest were picked to explore further via statistical testing in order to gain a better *depth* of understanding into some key variables

Effect of Occupation on Outcome

Exhibit: Jupyter Notebook Markdown in Capstone 1

Students

Null Hypothesis: Are students less likely to purchase a term deposit?

Test Used: Proportion Z-test with alpha of 0.01

Test Statistic: Z-test Statistic

Results

Z-test Statistic of 18.4 with

p-value of <0.0001

Reject the Null – Students are more likely to purchase a term deposit

Retired Customers

Null Hypothesis: Are retired people less likely to purchase a term deposit?

Test Used: Proportion Z-test with alpha of 0.01

Test Statistic: Z-test Statistic

Results

Z-test Statistic of 18.4 with

p-value of <0.0001

Reject the Null – Retired Customers are more likely to purchase a term deposit

Effect of Marital Status on Outcome

Exhibit: Jupyter Notebook Markdown in Capstone 1

Single Customers

Null Hypothesis: Are single people less likely to purchase a term deposit?

Test Used: Proportion Z-test with alpha of 0.01

Test Statistic: Z-test Statistic

Results

Z-test Statistic of 18.4 with

p-value of <0.0001

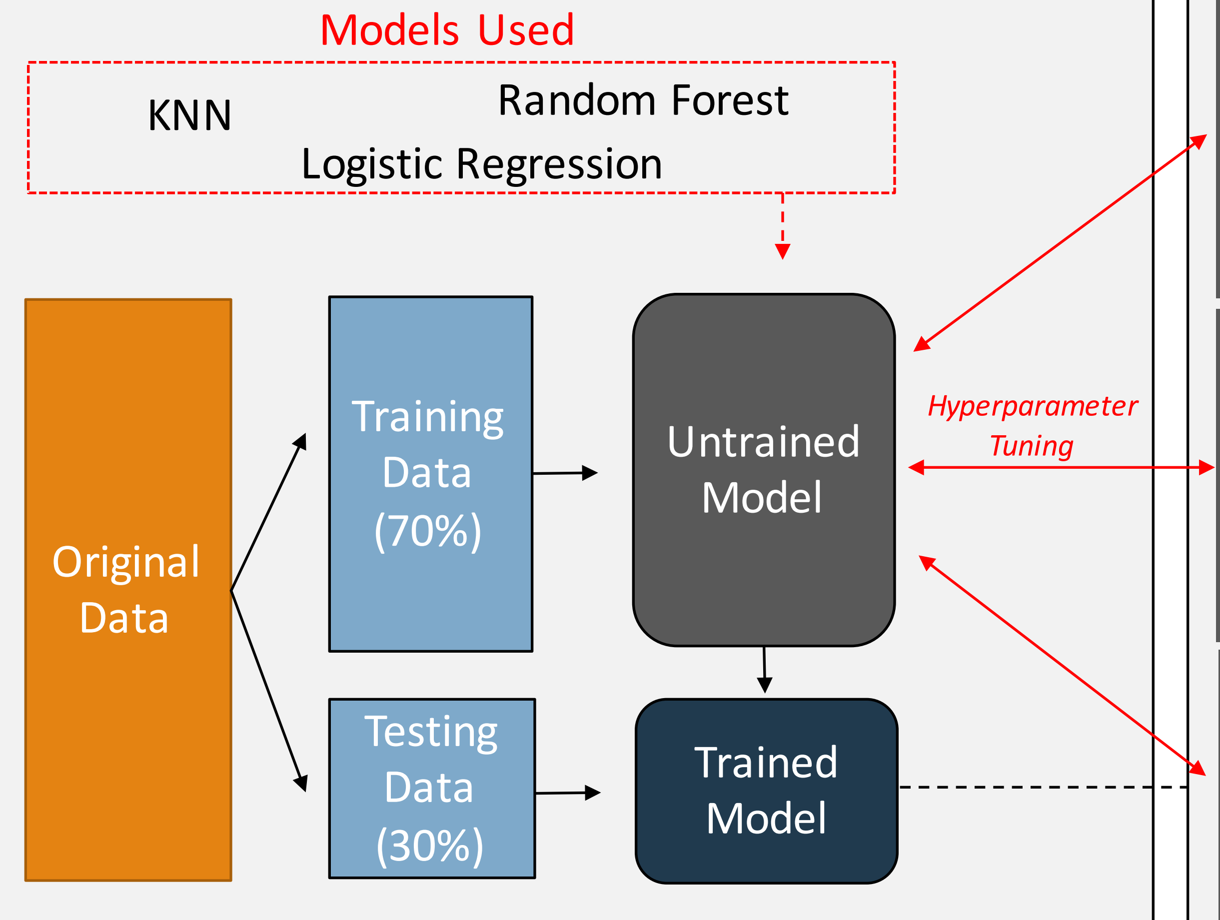
Reject the Null – Single Customers are more likely to purchase a term deposit

Table of Contents

In order to account for class imbalance issues with our dataset, AUROC scores were used to evaluate 3 different classifiers after each classifier underwent minor (i.e. 1 hyperparameter for each) hyperparameter tuning

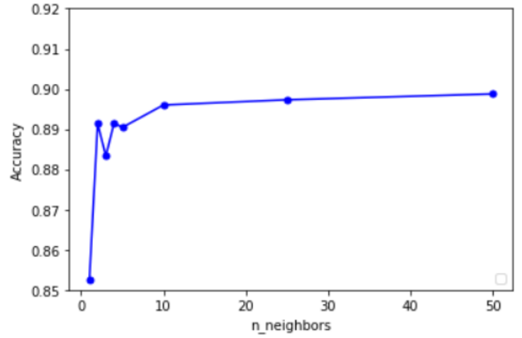
Training and Testing Model

Exhibit: Diagram of Workflow for Training Models used in Capstone 1

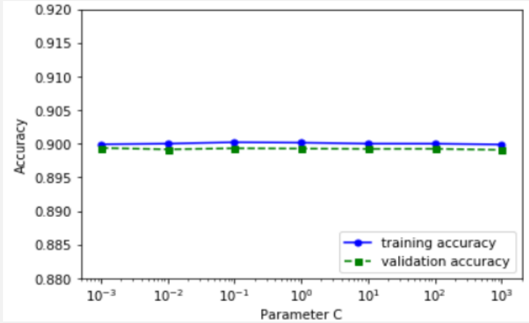


Tuning Models

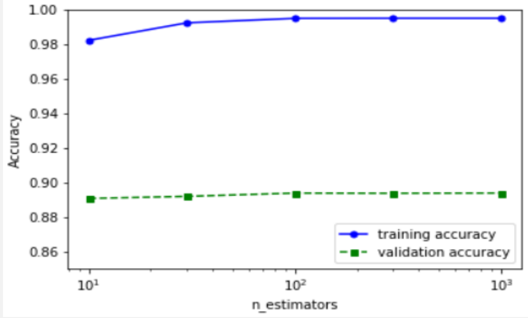
KNN



Logistic Regression



Random Forest



Evaluating Models

- **Evaluation Methodology: AUROC** Scores were used instead of accuracy scores to account for class imbalance issues with our dataset
- **RFC as best predictor:** We should pick the random forest classifier model with n_estimators of 100 as it performed the best

AUROC Scores

KNN: 0.7696
Log Reg: 0.7579
RFC: 0.7783

Table of Contents

Since my Capstone 1 Project was conducted to illustrate my basic understanding of machine learning methods, I focused less on all of the ways I could improve my models' accuracies but listed suggestions to come back to later

Results

- **Random Forest Classifies Bank Customers pretty well:** With our RFC classifier, we are able to predict whether or not a customer will purchase a term deposit ~78% of the time. This is fairly good as it beats a random predictor guessing yes or no, as the AUROC score for that would be 0.5. The bank can implement this predictor and focus their marketing efforts on the 78% of the customers that they can accurately predict the outcome for.
- **Translating to Actual Business Results:** According to EMI, a marketing agency specialized in banking, the top 40 banks spent nearly 14b in 2017, with 17 of those banks reporting double-digit growth since the prior year. For sake of simplicity, if we assume each customer for each bank is allocated the same marketing spend and that banks do not currently use any machine learning techniques or strategic prioritization, the top 40 banks could've saved nearly ~$3b. Now I can see why machine learning applications in business operations are getting all the buzz.

Final Thoughts

- **Spending more time on Feature Engineering:**
  - Getting rid of outliers in each variable in my feature array
  - Trying different combinations of one-hot encoder and label encoder of categorical variables to improve the "order" (e.g., perhaps a more defined ranking system of different education or occupation values)
  - More advanced methods of imputing unknown values, instead of just using the mode
- **Translating to Actual Business Results:**
  - Trying more hyperparameters for each classifier
  - Trying more classifiers