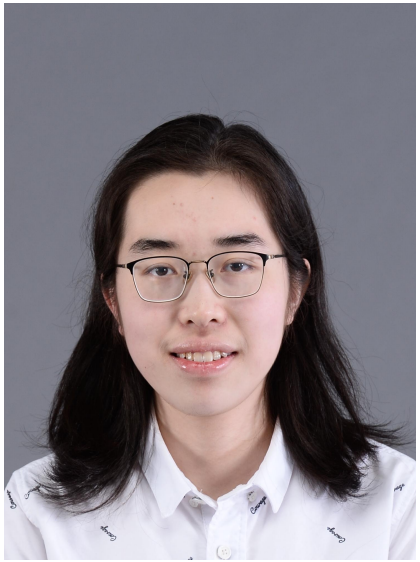
Team Double Ones

Will a Customer Subscribe to a Term Deposit?

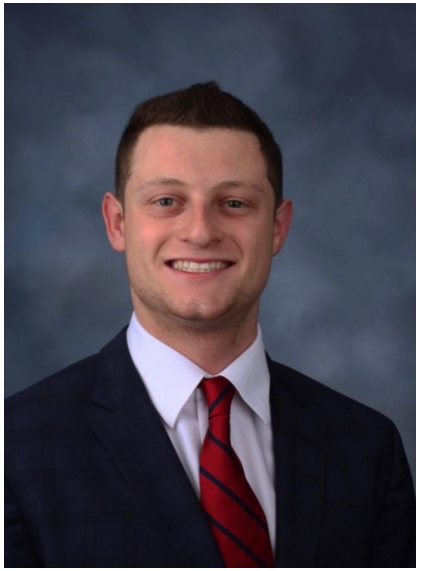
Team

Julie Wang



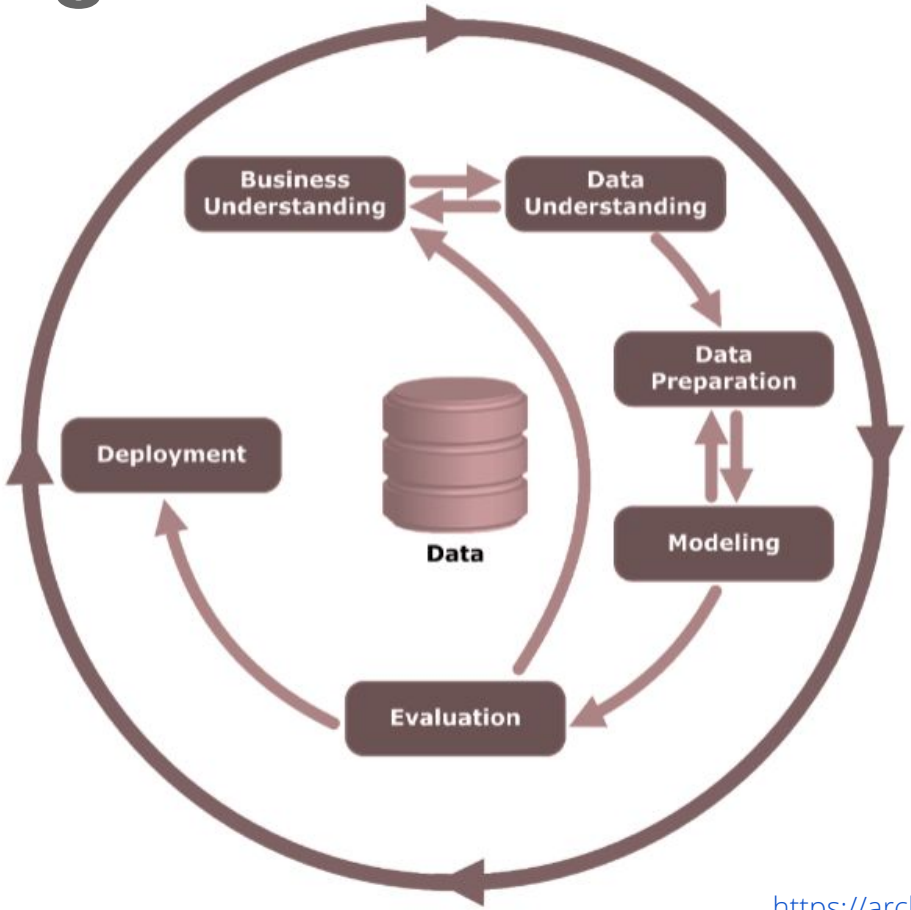
Susie Bai

Fei Hou



Daniel Weinberg

Agenda



Data

- Based on the dataset about records of telemarketing calls for a Portuguese Bank
- Key Task: Predict whether the customer will subscribe to a term deposit following the call
- 4 major aspects of information for predictive analysis and decision making
  - client characteristics
  - current campaign data
  - historical campaign data
  - socio-economic factors

https://archive.ics.uci.edu/ml/datasets/Bank+Marketing, bank-additional-full.csv

Business Understanding

What is a Term Deposit?

- A fixed-term investment that includes the deposit of money into an account for an agreed rate of interest
- Investors can only withdraw the money once the term ends

What type of problem is this?

- Binary classification problem of the target variable
- Will the customer subscribe or not? --"Yes"(encoded "1") or "No"(encoded "0")

Why is this prediction problem important?

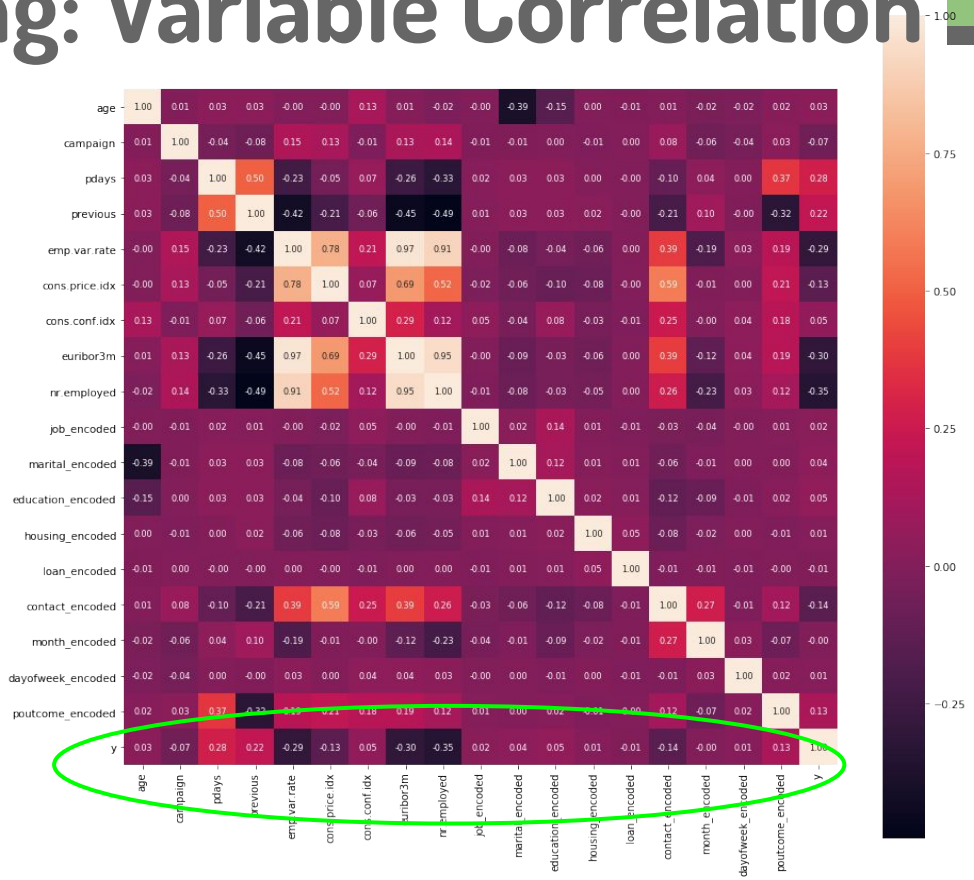
- Source of Profit
- Key Performance Goals for direct marketing campaigns in banks
- Predictive Analytics - Optimize the marketing strategies and improve success rates

Data Understanding: Overview

- Original Dataset:
  - 20 Features & 1 Target Variable (“y”: whether the customer has deposited or not)
    - Features including - ‘age’, ‘job’, ‘marital’, ‘education’, ‘default’, ‘housing’, ‘loan’, ‘contact’, ‘month’, ‘day_of_week’, ‘duration’, ‘campaign’, ‘pdays’*, ‘previous’, ‘poutcome’, ‘emp.var.rate’, ‘cons.price.idx’, ‘cons.conf.idx’, ‘euribor3m’ & ‘nr.employed'
  - 41,188 observations
- Problems detected:
  - Data Imbalance: 4640 ‘yes’ (**11.3%**) vs. 36548 ‘no’ (**88.7%**)
  - Data Leakage: “Duration”feature
    - Extremely correlated with target variable (y=No’ if duration=0)
    - Information not available at the time of decision making (call duration not known when making the call)

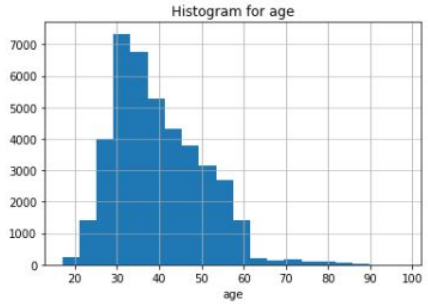
Data Understanding: Variable Correlation

- Top features correlated with the target variable (indicating more possible importance):
  - “nr.employed”
  - “euribor3m”
  - “emp.var.rate”
  - “pdays”
  - “previous”
- A little possible multicollinearity among some socio-economic factors
  - yet broader economic context matters
  - different aspects of macroeconomic conditions are reflected



Data Understanding: Numeric Features

- ‘age’, ‘campaign’, ‘previous’, ‘pdays’, ‘emp.var.rate’, ‘cons.price.idx’, ‘cons.conf.idx’, ‘euribor3m’, ‘nr.employed’
- Right-Skewness
  - e.g.: ‘age’, ‘campaign’, ‘previous’
- Left-Skewness
  - e.g.: ‘euribor3m’, ‘nr.employed’, ‘emp.var.rate’
- Special notation:
  - “‘pdays”: the value of “999” actually represents ‘customer not contacted in last campaign’ -> turned to be encoded as “-1” later

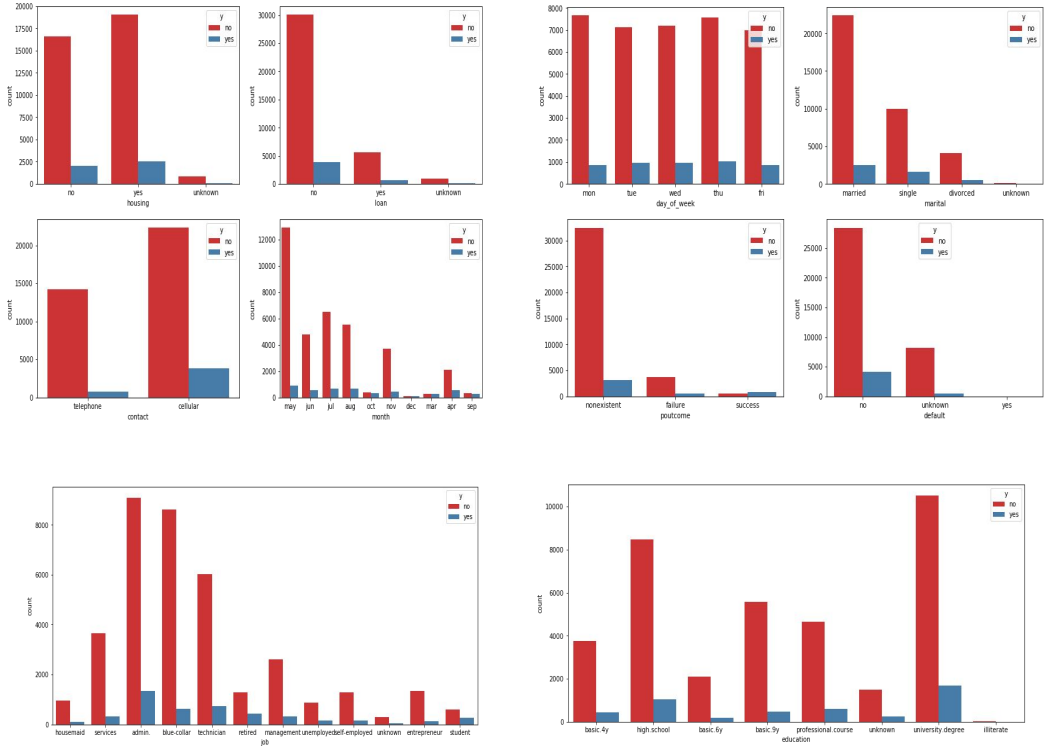


Description of Quantitative Values

	age	campaign	pdays	previous	emp.var.rate
count	41188.00000	41188.000000	41188.000000	41188.000000	41188.000000
mean	40.02406	2.567593	962.475454	0.172963	0.081886
std	10.42125	2.770014	186.910907	0.494901	1.570960
min	17.00000	1.000000	0.000000	0.000000	-3.400000
25%	32.00000	1.000000	999.000000	0.000000	-1.800000
50%	38.00000	2.000000	999.000000	0.000000	1.100000
75%	47.00000	3.000000	999.000000	0.000000	1.400000
max	98.00000	56.000000	999.000000	7.000000	1.400000

	cons.price.idx	cons.conf.idx	euribor3m	nr.employed
count	41188.000000	41188.000000	41188.000000	41188.000000
mean	93.575664	-40.502600	3.621291	5167.035911
std	0.578840	4.628198	1.734447	72.251528
min	92.201000	-50.800000	0.634000	4963.600000
25%	93.075000	-42.700000	1.344000	5099.100000
50%	93.749000	-41.800000	4.857000	5191.000000
75%	93.994000	-36.400000	4.961000	5228.100000
max	94.767000	-26.900000	5.045000	5228.100000

Data Understanding: Categorical Features



- Categorical variable in text format
  - e.g.: “contact”: {“cellular, “telephone”}
- “Unknown’s” in categorical features: “Job”(0.8%), ‘marital’(0.2%), ‘education’(4.2%), ‘default’(**20.9%**), ‘housing’(2.4%), ‘loan’(2.4%)

- **“month”**
  The month with the highest level of marketing activities is May.
- **“contact”**
  Customers reached out via 'telephone' type of contact seem less likely to subscribe.
- **“poutcome”**
  Customers who subscribed in previous marketing campaign more likely do it again.
- **“job”**
  Students or retired people seem more likely to be customers for term deposit subscription.

Data Preparation

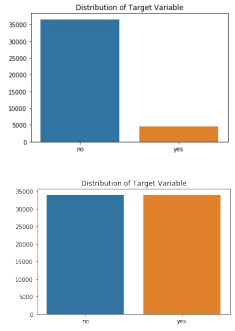
3 types of models: Basic, Basic Optimized, Transformed Optimized

Remove Observations with 'unknown's

Removed data points with any 'unknown' in feature vector' (92.9% data points left)

Data Rebalancing with Resampling Technique

Oversampling of Minority class. The two labeled classes achieve exactly **50/50 ratio**; **67,974 observations after resampling**



Remove Features

Removed:
'duration' - data leakage;
'default' - Not informative

Categorical Data Encoder

Encoded the 9 categorical variables using LabelEncoder

Transformations

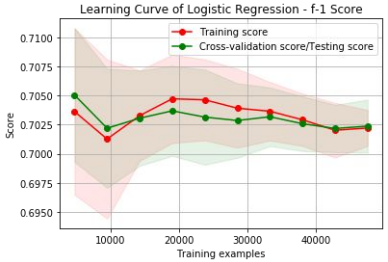
Transformed the data to make features approximate normal distribution as much as possible

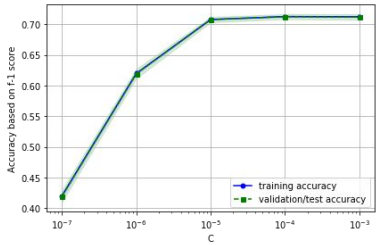
Model: Basic Default

Logistic Regression

Default Parameters
C = 1.0
Penalty = 'l2'

F1 = 0.7274

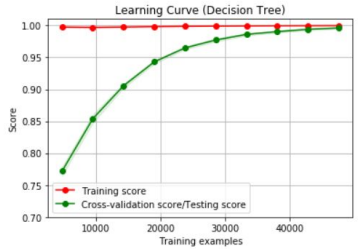


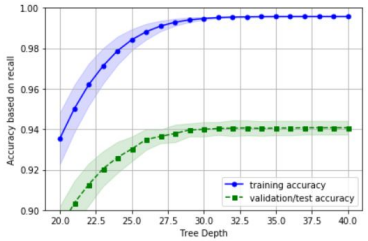


Decision Tree

Default Parameters
Criterion = 'gini'
Min_Samples_Split
=2

F1 = 0.9407

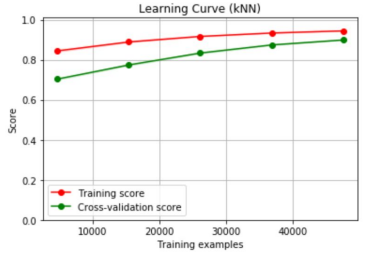


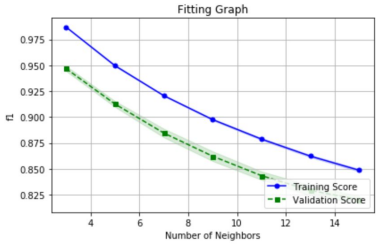


k-Nearest Neighbor

Default Parameters
3-NN
Euclidean Distance
Minkowski

F1 = 0.8935



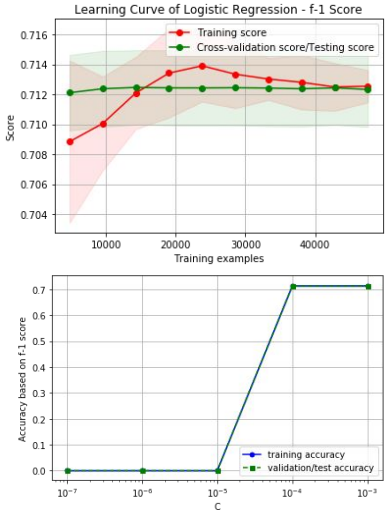


Model: Basic Optimized

Logistic Regression

C = .001
Penalty = 'l1'

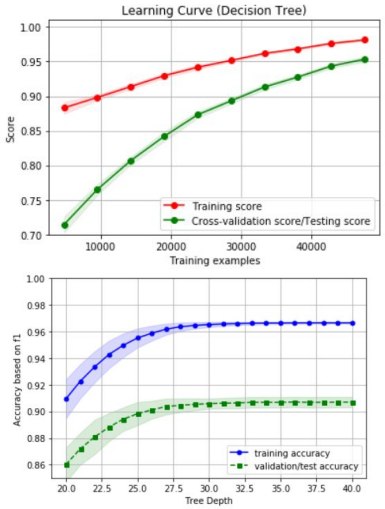
F1 = 0.7122



Decision Tree

Criterion = 'entropy'
Max_Depth = 38
Min_Samples_split = 10

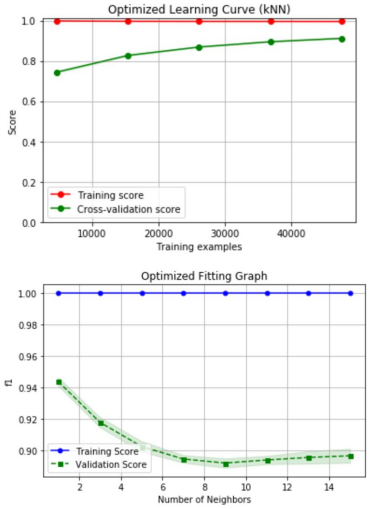
F1 = 0.9124



k-Nearest Neighbor

3-NN
Manhattan Distance
Weights - Distance
Minkowski

F1 = 0.9091

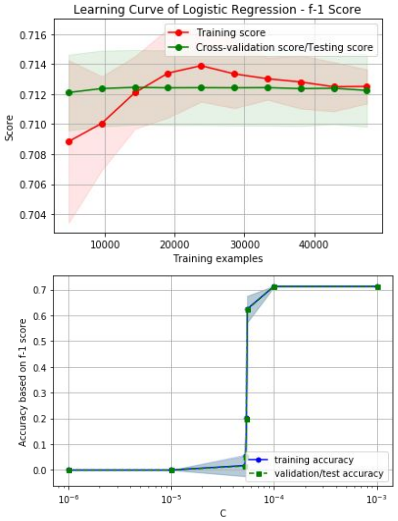


Model: Transformed Optimized

Logistic Regression

C = .001
Penalty = 'l1'

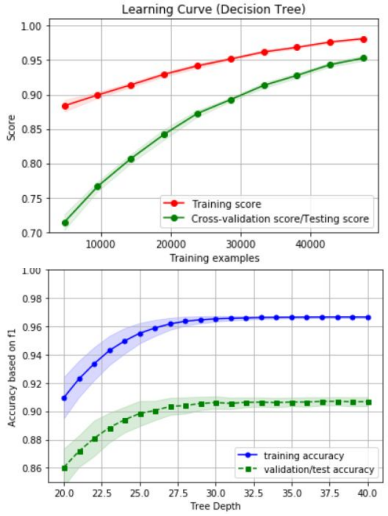
F1 = 0.7122



Decision Tree

Criterion = 'entropy'
Max_Depth = 36
Min_Samples_split = 10

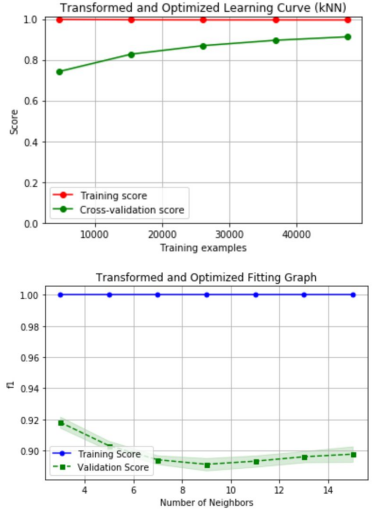
F1 = 0.9124



k-Nearest Neighbor

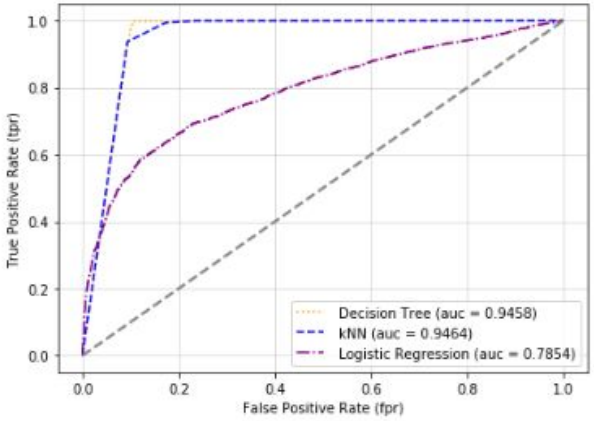
3-NN
Manhattan Distance
Weights - Distance
Minkowski

F1 = 0.9096



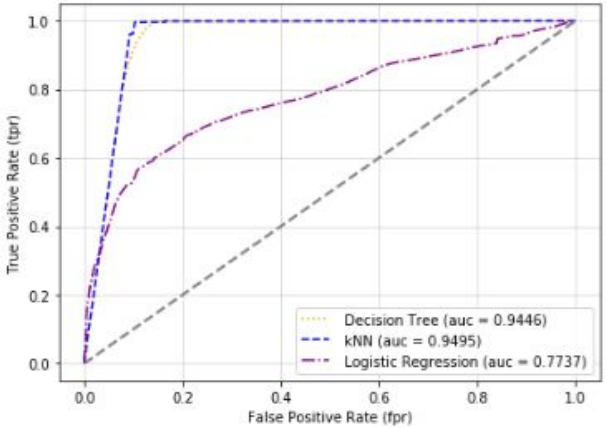
Model: ROC – AUC

Basic - Default



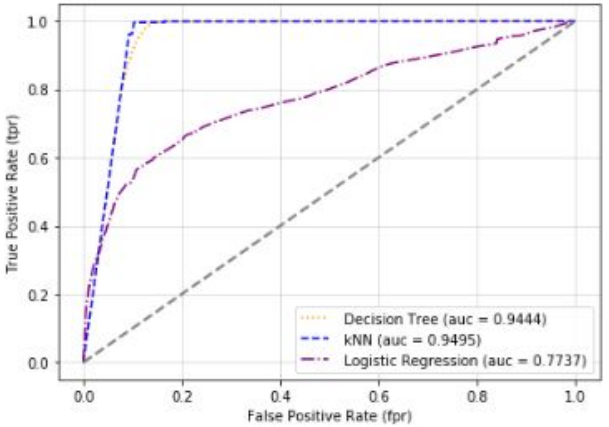
Decision Tree = .9458
k-NN = .9464
Logistic Regression = .7854

Basic - Optimized



Decision Tree = .9446
k-NN = .9495
Logistic Regression = .7737

Transformed - Optimized



Decision Tree = .9444
k-NN = .9495
Logistic Regression = .7737

Model Choice

Which Model should we go with?

	Logistic Regression	Decision Tree	k-Nearest Neighbor
Basic	Default Model Choices C = 1.0 Penalty = 'l2' **F1 = 0.7274**	Default Model Choices Criterion = 'gini' Min_Samples_Split =2 **F1 = 0.9407**	Default 3-NN Euclidean Distance Minkowski **F1 = 0.8935**
Basic Optimized	C = .001 Penalty = 'l1' **F1 = 0.7122**	Criterion = 'entropy' Max_Depth = 38 Min_Samples_split = 10 **F1 = 0.9124**	3-NN Manhattan Distance Weights - Distance Minkowski **F1 = 0.9091**
Transformed Optimized	C = .001 Penalty = 'l1' **F1 = 0.7122**	 Criterion = 'entropy' Max_Depth = 36 Min_Samples_split = 10 **F1 = 0.9124**	3-NN Manhattan Distance Weights - Distance Minkowski **F1 = 0.9096**

Model Choice & Evaluation

Which Model should we go with?

- AUC
  - Transformed Optimized k-NN is better by .0051
- F1 Score
  - Transformed Optimized Decision Tree has better F1 score by .0028
- Interpretability
  - Decision Tree is best by revealing a set of rules, easier for managers to understand
- Overfitting Concerns
  - Not going with Basic Decision Tree due to possible overfitting

Transformed Optimized Decision Tree

Criterion = 'entropy'
Max_Depth = 36
Min_Samples_split = 10
F1 = 0.9124
Accuracy = .91

Naïve Profit

How should the results of the data mining be evaluated?

- Cost per call: **$.015**
- Proportional 5% Return on min term deposit → (.05)($1000) = **$50**

➢ Priority: maximizing profit for every correct model prediction

	Deposit	No Deposit
Contact	$49.985	-$0.015
No Contact	$0	$0

Expected Profit

Expected Value Method

	Deposit	No Deposit
Contact	8189	1167
No Contact	317	7321

	Deposit	No Deposit
Contact	0.4819	0.0687
No Contact	0.0187	0.4308

	Deposit	No Deposit
Contact	$49.985	-$0.015
No Contact	$0	$0

$$Expected\ Value = \$12.04338$$

Deployment

How will our data mining be deployed?

- Preliminary Screening
- Pilot Program
- Customer Targeting Tweaks

What issues could we have with deployment?

- Full Observations with ‘unknowns’
- Purchase more consumer data to have complete observations
- Prediction Lag

Deployment

Are there important ethical considerations?

- Personal Data
- Do not prey on vulnerable people

What are the risks associated with our proposed plan and how can we mitigate them?

- Personal data laws
- Different Model Options
- Risk of Underfitting and/or Overfitting
-

Thank you!
Questions?

Appendix

Data Understanding

Qualitative Data

```
Description of 'job'
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
```

```
Description of 'marital'
married     24928
single      11568
divorced     4612
unknown        80
Name: marital, dtype: int64
```

```
Description of 'education'
university.degree      12168
high.school             9515
basic.9y                6045
professional.course     5243
basic.4y                4176
basic.6y                2292
unknown                 1731
illiterate                18
Name: education, dtype: int64
```

```
Description of 'default'
no         32588
unknown     8597
yes            3
Name: default, dtype: int64
```

```
Description of 'housing'
yes        21576
no         18622
unknown      990
Name: housing, dtype: int64
```

```
Description of 'loan'
no         33950
yes         6248
unknown      990
Name: loan, dtype: int64
```

```
Description of 'contact'
cellular     26144
telephone    15044
Name: contact, dtype: int64
```

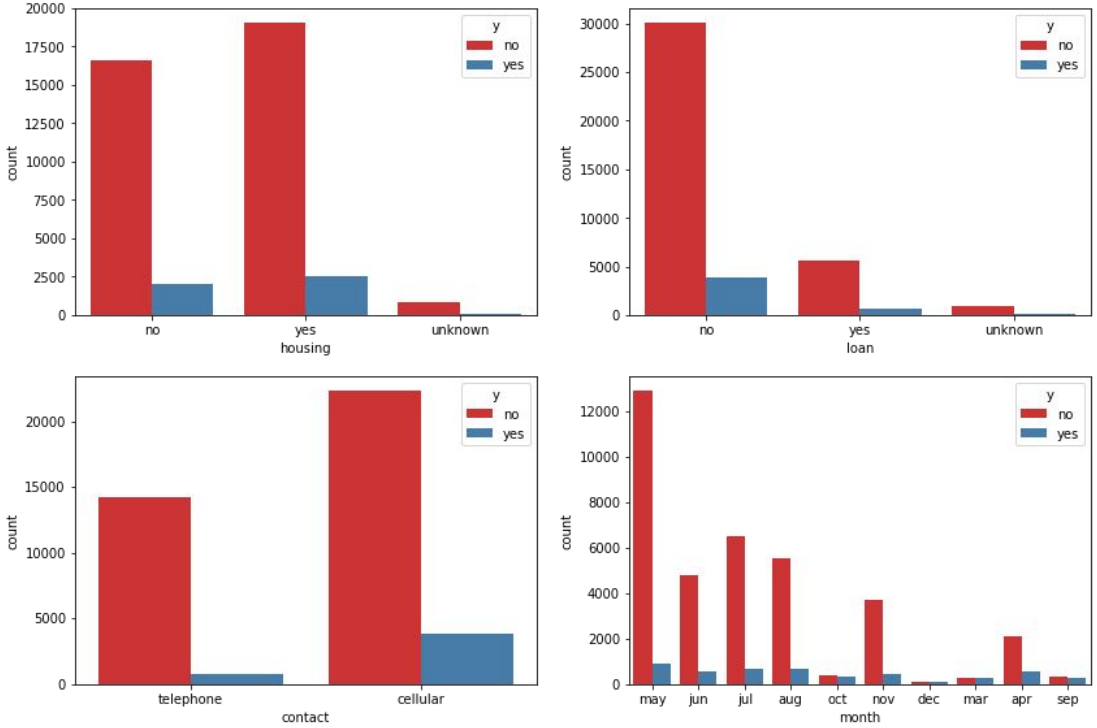
```
Description of 'month'
may    13769
jul     7174
aug     6178
jun     5318
nov     4101
apr     2632
oct      718
sep      570
mar      546
dec      182
Name: month, dtype: int64
```

```
Description of 'day_of_week'
thu    8623
mon    8514
wed    8134
tue    8090
fri    7827
Name: day_of_week, dtype: int64
```

```
Description of 'poutcome'
nonexistent    35563
failure         4252
success         1373
Name: poutcome, dtype: int64
```

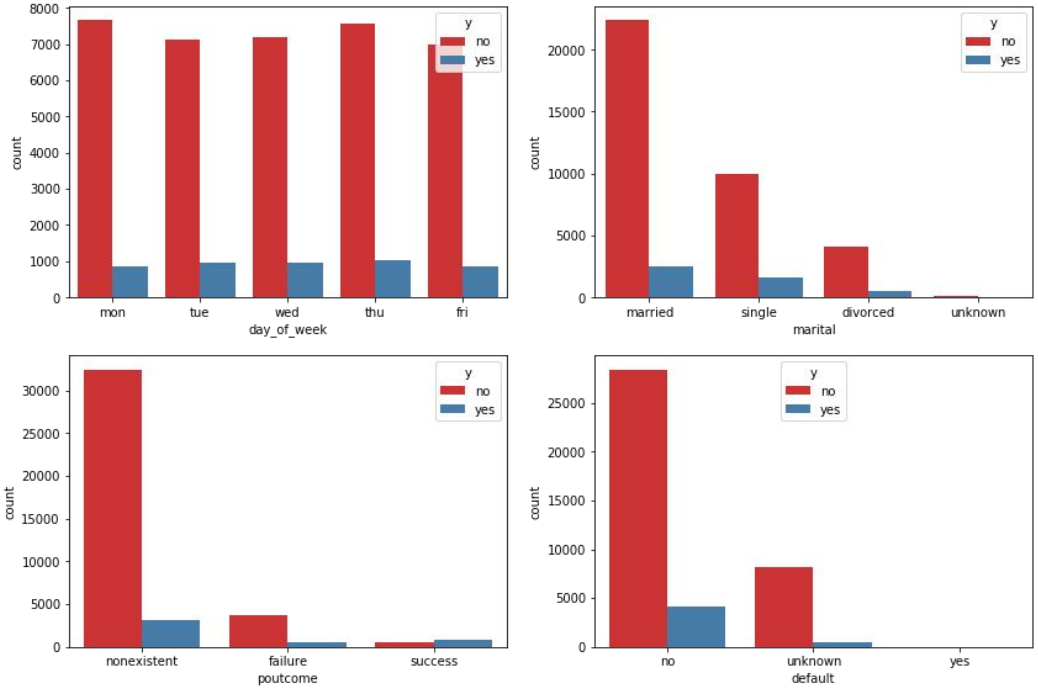
Data Understanding

Qualitative Data



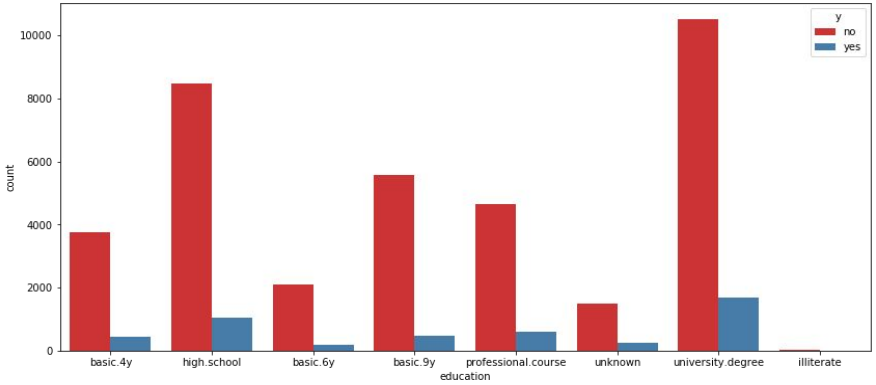
Data Understanding

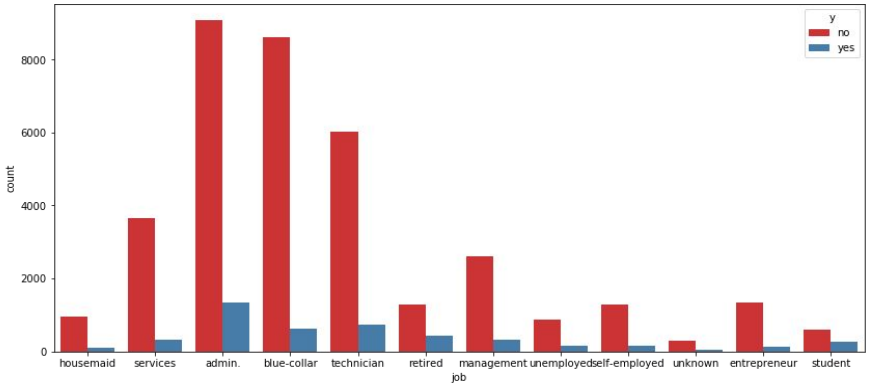
Qualitative Data



Data Understanding

Qualitative Data





Data Preparation

Data Encoding

```
Categories for 'job' now: ['housemaid' 'services' 'admin.' 'technician' 'blue-collar' 'retired'
 'management' 'unemployed' 'self-employed' 'entrepreneur' 'student']
Number of Categories for 'job' now: 11

Corresponding Encodings Display:
[ 3  7  7  0  7  7  0  9  7  7  1  3  1  5  1  1  1  4 10  1  5  9  0  9
  6  9  4  1  7  9  0  9  2  0  0  0  0  9  0  1  0  1  0  3  0  0  1  9  4
  7 10]
['housemaid' 'services' 'services' 'admin.' 'services' 'services' 'admin.'
 'technician' 'services' 'services' 'blue-collar' 'housemaid'
 'blue-collar' 'retired' 'blue-collar' 'blue-collar' 'blue-collar'
 'management' 'unemployed' 'blue-collar' 'retired' 'technician' 'admin.'
 'technician' 'self-employed' 'technician' 'management' 'blue-collar'
 'services' 'technician' 'admin.' 'technician' 'entrepreneur' 'admin.'
 'admin.' 'admin.' 'technician' 'admin.' 'blue-collar' 'admin.'
 'blue-collar' 'admin.' 'housemaid' 'admin.' 'admin.' 'blue-collar'
 'technician' 'management' 'services' 'unemployed']
```

```
Categories for 'education' now: ['basic.4y' 'high.school' 'basic.6y' 'basic.9y' 'professional.course'
 'university.degree' 'illiterate']
Number of Categories for 'education' now: 7

Corresponding Encodings Display:
[0 3 3 1 3 2 5 5 3 3 3 0 1 2 1 1 2 2 3 0 3 3 3 6 2 6 0 0 3 2]
['basic.4y' 'high.school' 'high.school' 'basic.6y' 'high.school'
 'basic.9y' 'professional.course' 'professional.course' 'high.school'
 'high.school' 'high.school' 'basic.4y' 'basic.6y' 'basic.9y' 'basic.6y'
 'basic.6y' 'basic.9y' 'basic.9y' 'high.school' 'basic.4y' 'high.school'
 'high.school' 'high.school' 'university.degree' 'basic.9y'
 'university.degree' 'basic.4y' 'basic.4y' 'high.school' 'basic.9y']
```

```
Categories for 'month' now: ['may' 'jun' 'jul' 'aug' 'oct' 'nov' 'dec' 'mar' 'apr' 'sep']
Number of Categories for 'month' now: 10

Corresponding Encodings Display:
[6 6 6 6 6 6 6 6 6 6]
['may' 'may' 'may' 'may' 'may' 'may' 'may' 'may' 'may' 'may']
```

```
Categories for 'day_of_week' now: ['mon' 'tue' 'wed' 'thu' 'fri']
Number of Categories for 'day_of_week' now: 5

Corresponding Encodings Display:
[1 1 1 1 1 1 1 1 1 1]
['mon' 'mon' 'mon' 'mon' 'mon' 'mon' 'mon' 'mon' 'mon' 'mon']
```

```
Categories for 'marital' now: ['married' 'single' 'divorced']
Number of Categories for 'marital' now: 3

Corresponding Encodings Display:
[1 1 1 1 1 1 1 2 2 2 2 0 1 1 1 1 1 2 1 1]
['married' 'married' 'married' 'married' 'married' 'married' 'married'
 'single' 'single' 'single' 'single' 'divorced' 'married' 'married'
 'married' 'married' 'married' 'single' 'married' 'married']
```

```
Categories for 'housing' now: ['no' 'yes']
Number of Categories for 'education' now: 2

Corresponding Encodings Display:
[0 0 1 0 0 0 0 1 1 1 0 1 1 1 1 1 1 0 0 1 1 1 1 1 0 0 0 1 0 0 1]
['no' 'no' 'yes' 'no' 'no' 'no' 'no' 'yes' 'yes' 'yes' 'yes' 'no' 'yes' 'yes'
 'yes' 'yes' 'yes' 'yes' 'no' 'no' 'yes' 'yes' 'yes' 'yes' 'no' 'no' 'no'
 'yes' 'no' 'no' 'yes']
```

```
Categories for 'loan' now: ['no' 'yes']
Number of Categories for 'loan' now: 2

Corresponding Encodings Display:
[0 0 0 0 1 0 0 0 0 0 1 0 0 1 0 1 1 0 0 0 0 0 0 1 0 0 0 0 0 0]
['no' 'no' 'no' 'no' 'yes' 'no' 'no' 'no' 'no' 'no' 'yes' 'no' 'no' 'yes'
 'no' 'yes' 'yes' 'no' 'no' 'no' 'no' 'no' 'no' 'yes' 'no' 'no' 'no' 'no'
 'no' 'no']
```

```
Categories for 'contact' now: ['telephone' 'cellular']
Number of Categories for 'contact' now: 2

Corresponding Encodings Display:
[1 1 1 1 1 1 1 1 1 1]
['telephone' 'telephone' 'telephone' 'telephone' 'telephone' 'telephone'
 'telephone' 'telephone' 'telephone' 'telephone']
```

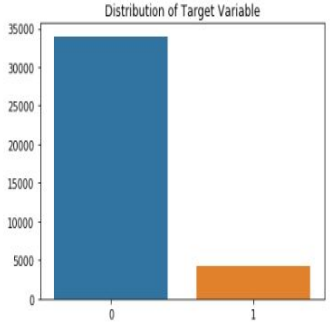
```
Categories for 'poutcome' now: ['nonexistent' 'failure' 'success']
Number of Categories for 'poutcome' now: 3

Corresponding Encodings Display:
[1 1 1 1 1 1 1 1 1 1]
['nonexistent' 'nonexistent' 'nonexistent' 'nonexistent' 'nonexistent'
 'nonexistent' 'nonexistent' 'nonexistent' 'nonexistent' 'nonexistent']
```

Data Preparation

Data Rebalancing with Resampling Technique

```
# Exploring Target Variable after dropping "unknowns"
unique, counts = np.unique(y, return_counts=True) # 2 distinct classes relatively balanced
print("The frequency of instances per class is: " , dict(zip(unique, counts)), " where YES means the customer deposited, and NO
print("The Percentages per class:", "No: %.2f;" % (counts[0]/(counts[0]+counts[1])), "Yes: %.2f" % (counts[1]/(counts[0]+counts[
sns.barplot(unique, counts)
plt.title("Distribution of Target Variable")
```

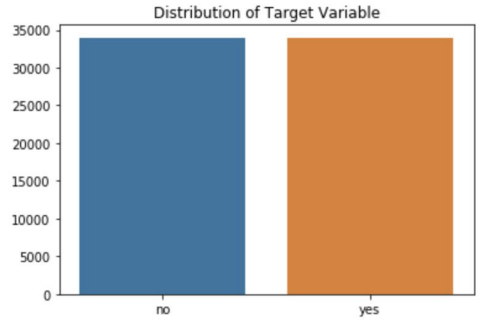
```
The frequency of instances per class is:  {0: 33987, 1: 4258}  where YES means the customer deposited, and NO means the custome
r did not deposit.
The Percentages per class: No: 0.89; Yes: 0.11

Text(0.5, 1.0, 'Distribution of Target Variable')
```



Before Resampling

```
In [36]: #check the distribution of the binary target variable again after resampli
unique, counts = np.unique(y_balance, return_counts=True) # 2 distinct cla
print("The frequency of instances per class is: " , dict(zip(unique, count
print("The Percentages per class:", "No: %.2f;" % (counts[0]/(counts[0]+co
sns.barplot(unique, counts)
plt.title("Distribution of Target Variable")
```

```
The frequency of instances per class is:  {'no': 33987, 'yes': 33987}  wh
ere YES means the customer deposited, and NO means the customer did not d
eposit.
The Percentages per class: No: 0.50; Yes: 0.50
Out[36]: Text(0.5, 1.0, 'Distribution of Target Variable')
```

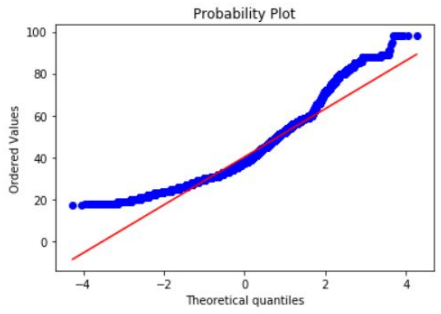


After Resampling

Data Preparation

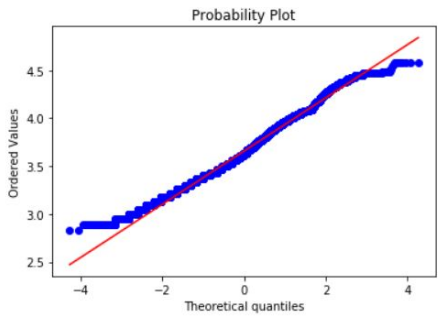
Transformations

QQplot of age



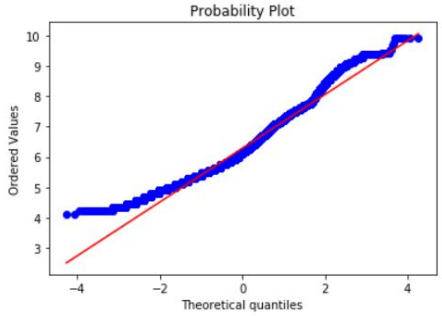
0.9988905745380053

QQplot of age (log)



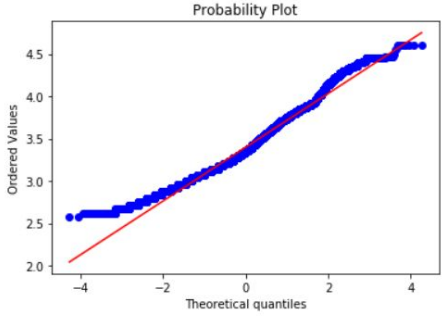
0.2803588348096721

QQplot of age (sqrt)



0.6258668265924763

QQplot of age (cubert)



0.5084584142698018

Procedure:

- We wanted feature data to be as normal as possible
- Plotted the Q-Q plots and skewness of each feature with different transformations

Data Preparation

Transformations

Log Transformations

- age: log()
- campaign: log()

Log Transformations for Features with '0'

- pdays: log(x+2)
- previous: log(x+2)
- job_encoded: log(x+2)
- housing_encoded: log(x+2)
- loan_encoded: sqrt(x+2)
- contact_encoded: sqrt(x+2)

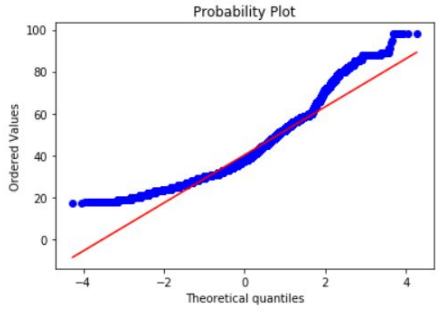
Sqrt Transformations for Features with Negative Values

- cons.conf.idx: sqrt(x+1- min(X[]))

Data Preparation

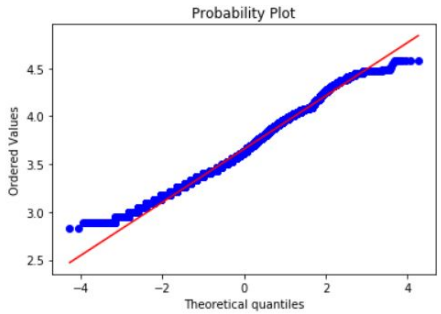
Transformation

QQplot of age



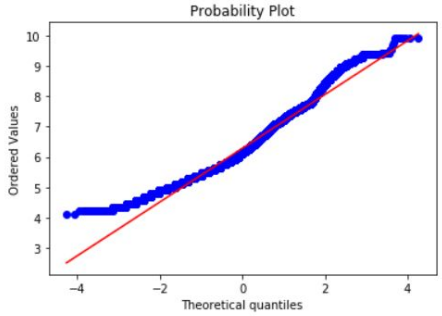
0.9988905745380053

QQplot of age (log)



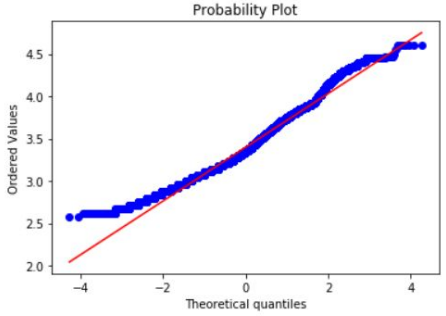
0.2803588348096721

QQplot of age (sqrt)



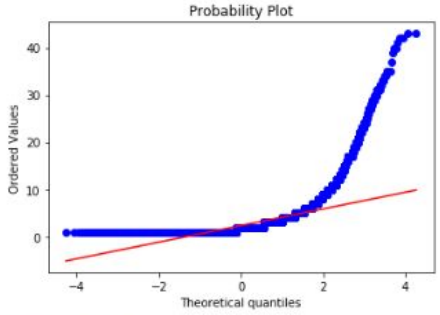
0.6258668265924763

QQplot of age (cubert)



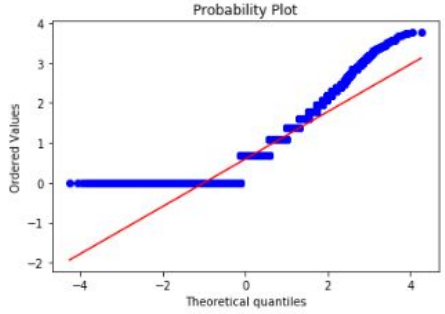
0.5084584142698018

QQplot of campaign



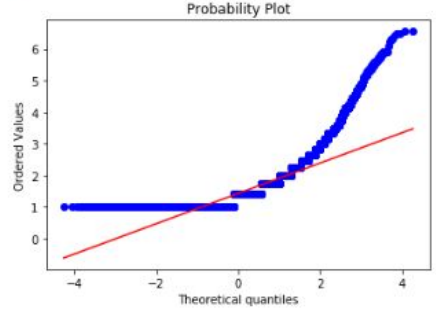
4.93464066852659

QQplot of campaign (log)



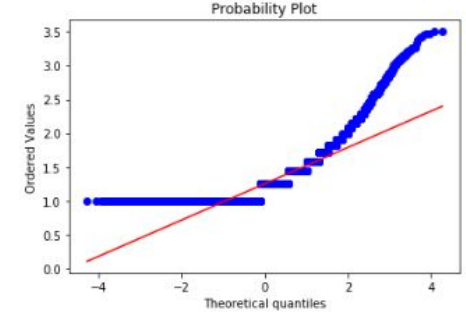
0.9358595105489252

QQplot of campaign (sqrt)



2.155538027525771

QQplot of campaign (cubert)

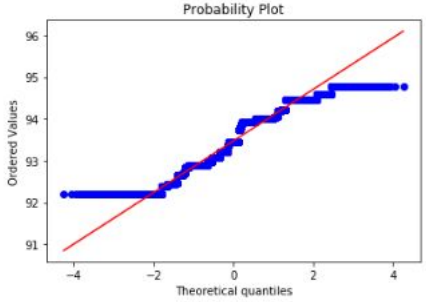


1.631865772268322

Data Preparation

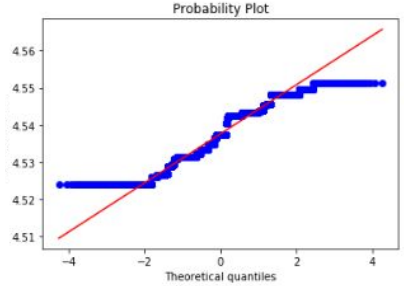
Transformation

QQplot of cons.price.idx



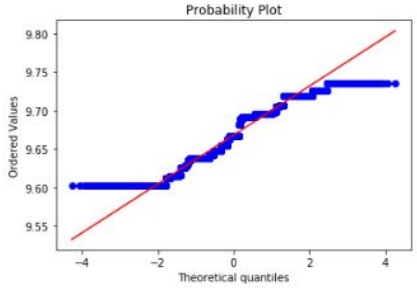
-0.09726198364667638

plot of cons.price.idx (log)



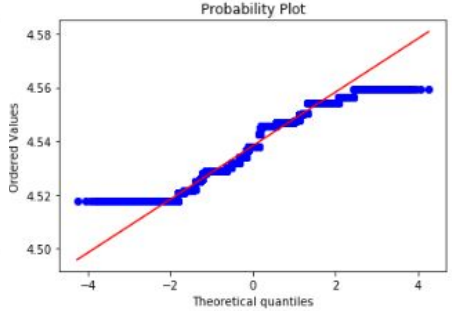
-0.10770917953764846

QQplot of cons.price.idx (sqrt)



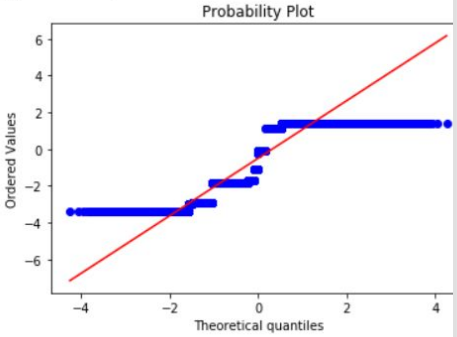
-0.10248309845266634

Qplot of cons.price.idx (cubert)



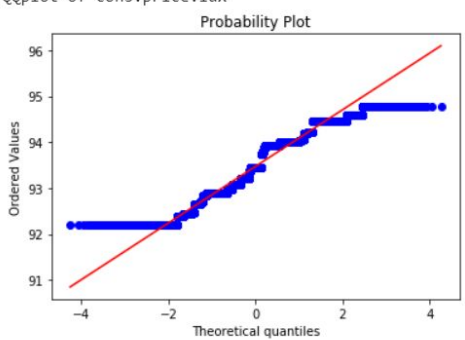
-0.10422457011491884

QQplot of emp.var.rate



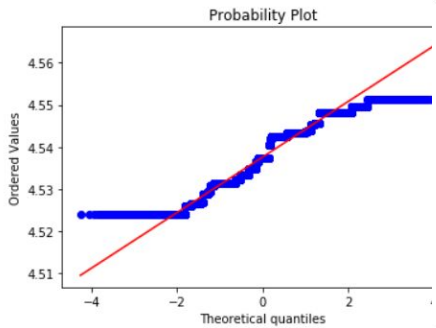
-0.1528190433781906

QQplot of cons.price.idx



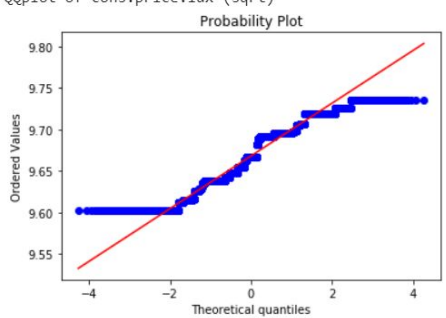
-0.09726198364667638

QQplot of cons.price.idx (log)



-0.10770917953764846

QQplot of cons.price.idx (sqrt)

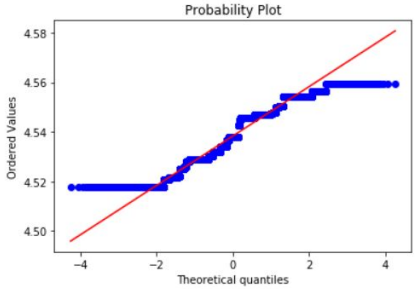


-0.10248309845266634

Data Preparation

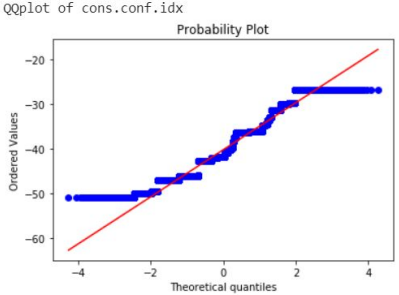
Transformation

QQplot of cons.price.idx (cubert)



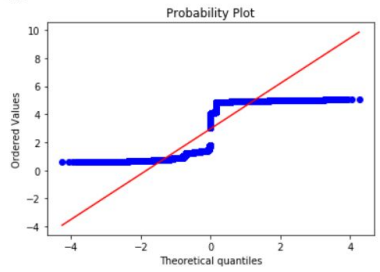
-0.10422457011491884

QQplot of cons.conf.idx



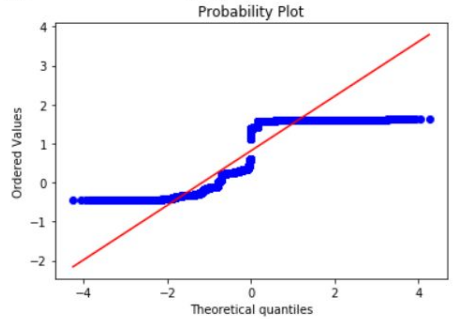
0.3488951118146305

QQplot of euribor3m



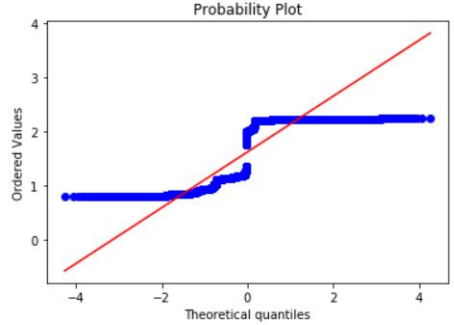
-0.028566476282109912

QQplot of euribor3m (log)



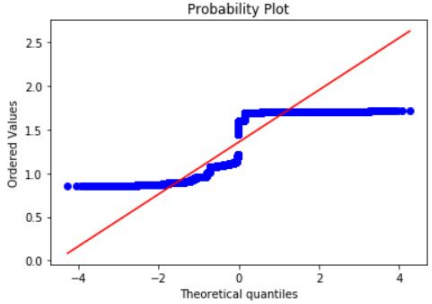
-0.20506880689083443

QQplot of euribor3m (sqrt)



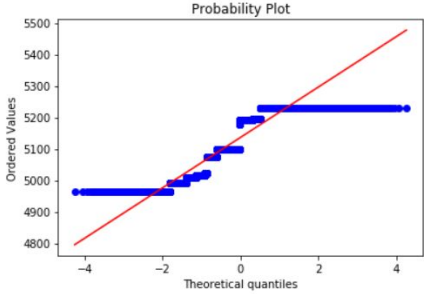
-0.09225496179306715

QQplot of euribor3m (cubert)



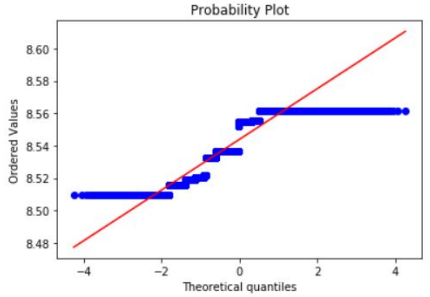
-0.12305852955953021

QQplot of nr.employed



-0.436213809425052

QQplot of nr.employed (log)

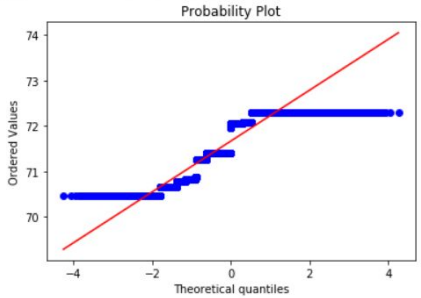


-0.45240715893871686

Data Preparation

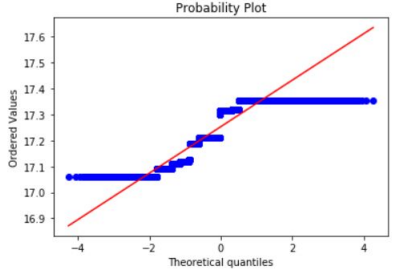
Transformation

QQplot of nr.employed (sqrt)



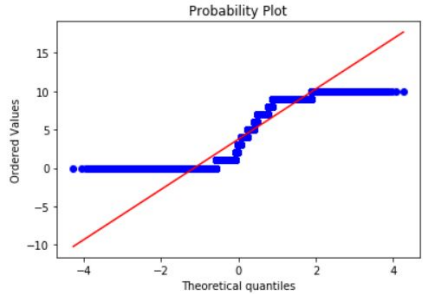
-0.44428238511650975

QQplot of nr.employed (cubert)



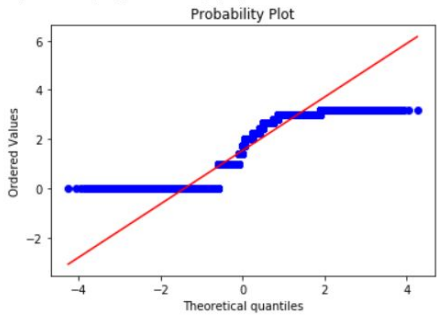
-0.44698440744377776

QQplot of job_encoded



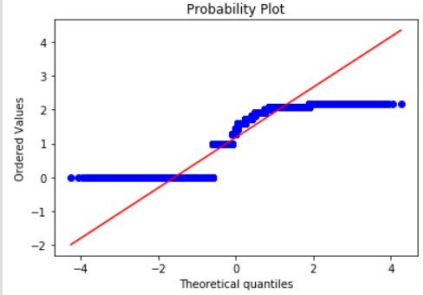
0.39763809027860214

QQplot of job_encoded (sqrt)



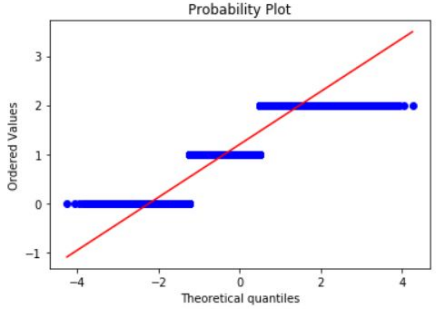
-0.10861458386329938

QQplot of job_encoded (cubert)



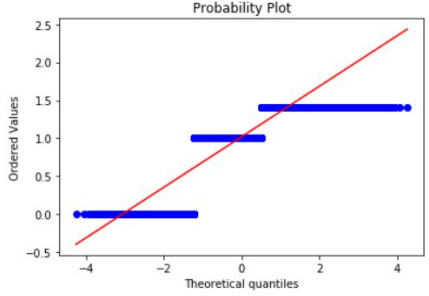
-0.42046874253809724

QQplot of marital_encoded



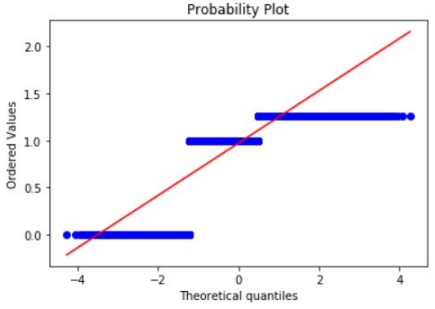
-0.1488878044510951

QQplot of marital_encoded (sqrt)



-1.482147310317928

QQplot of marital_encoded (cubert)

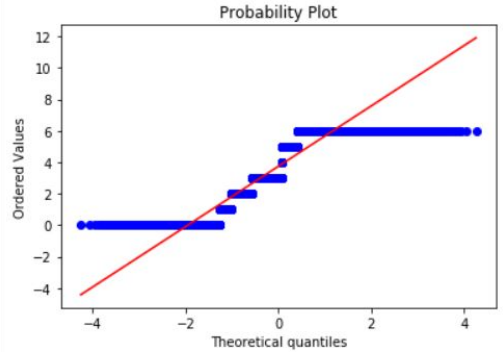


-1.9963055857370917

Data Preparation

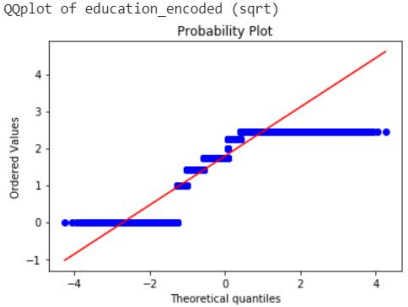
Transformation

QQplot of education_encoded



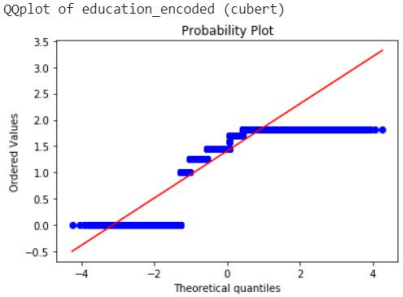
-0.34507456807422504

QQplot of education_encoded (sqrt)



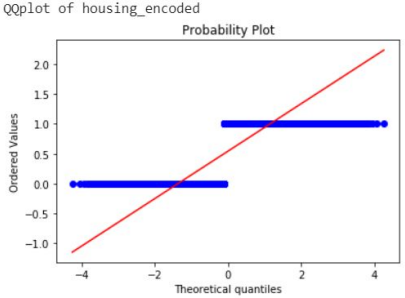
-1.224739424655726

QQplot of education_encoded (cubert)

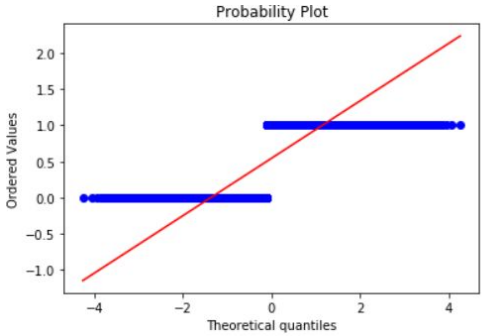


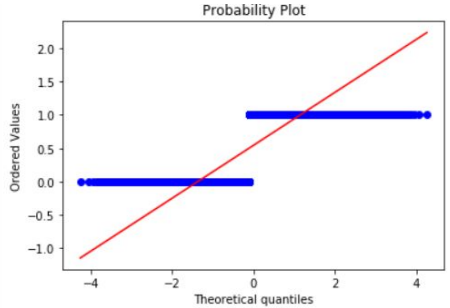
-1.759541211364812

QQplot of housing_encoded



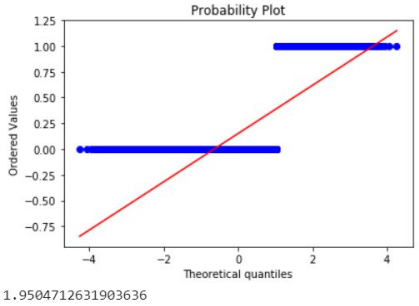
-0.18056886703997457





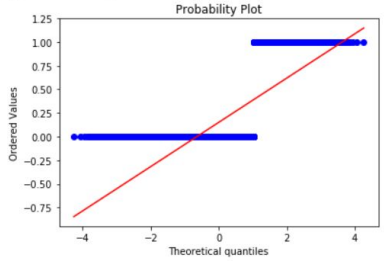
-0.18056886703997457

QQplot of loan_encoded



1.9504712631903636

QQplot of loan_encoded (sqrt)

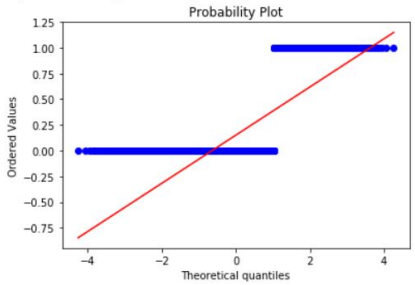


1.9504712631903636

Data Preparation

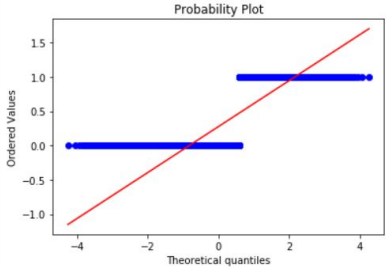
Transformation

QQplot of loan_encoded (cubert)



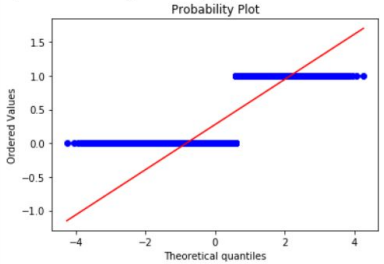
1.9504712631903636

QQplot of contact_encoded



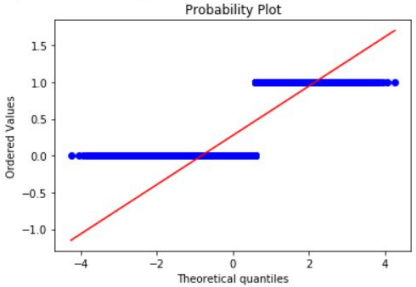
1.000972355553644

QQplot of contact_encoded (sqrt)



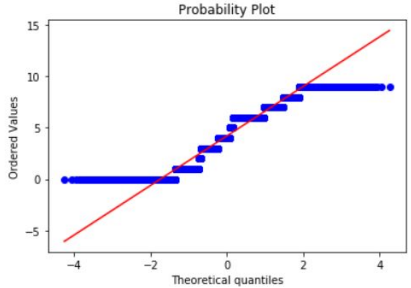
1.000972355553644

QQplot of contact_encoded (cubert)



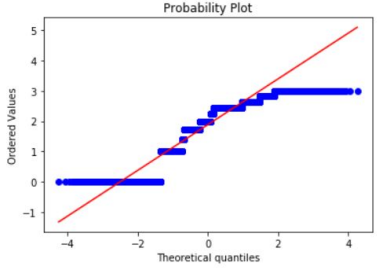
1.000972355553644

QQplot of month_encoded



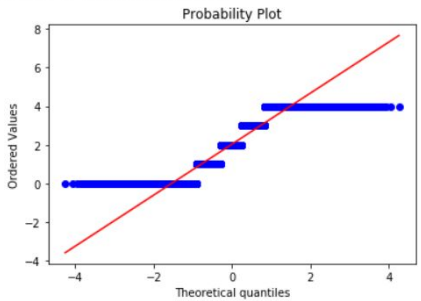
-0.16674756347642256

QQplot of month_encoded (sqrt)



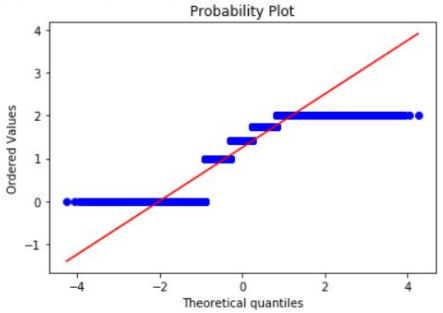
-0.9875177087741251

QQplot of dayofweek_encoded



-0.028864332248686085

QQplot of dayofweek_encoded (sqrt)

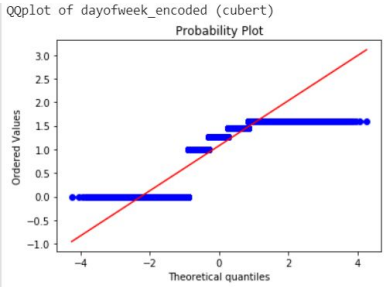


-0.8153949406132363

Data Preparation

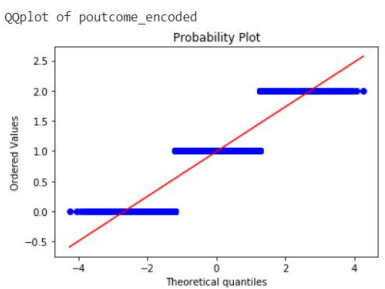
Transformation

QQplot of dayofweek_encoded (cubert)



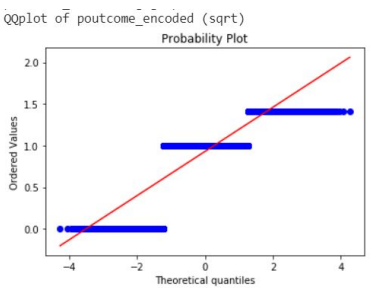
-1.1726082577822114

QQplot of poutcome_encoded



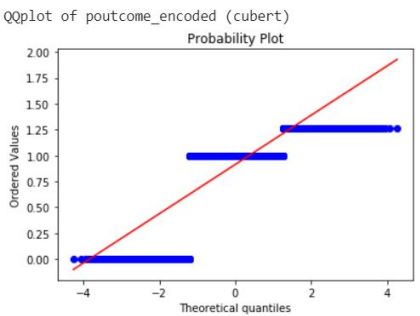
-0.03535001622374884

QQplot of poutcome_encoded (sqrt)



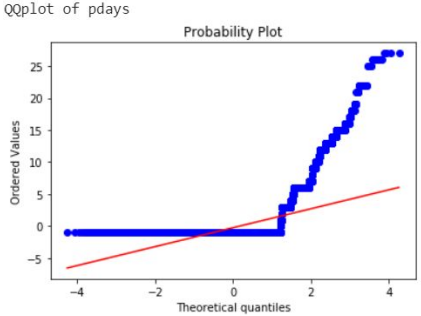
-1.7703430872341304

QQplot of poutcome_encoded (cubert)



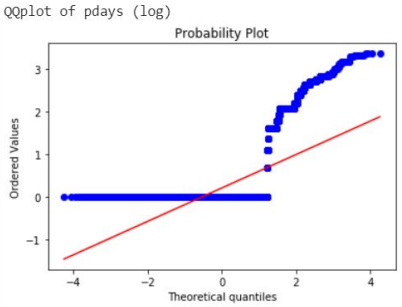
-2.160733369465757

QQplot of pdays



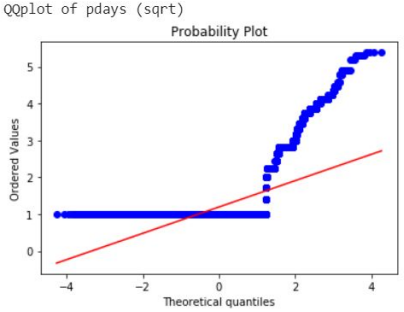
4.033659070849569

QQplot of pdays (log)



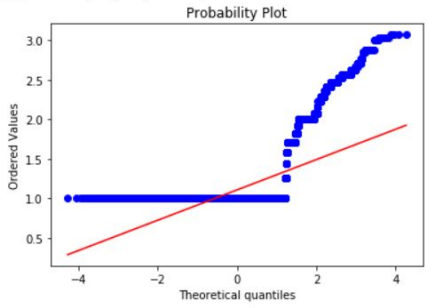
2.722804347430462

QQplot of pdays (sqrt)



3.1271512672503468

QQplot of pdays (cubert)

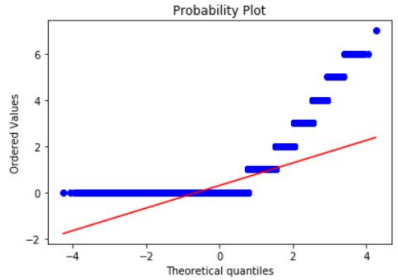


2.950797677351143

Data Preparation

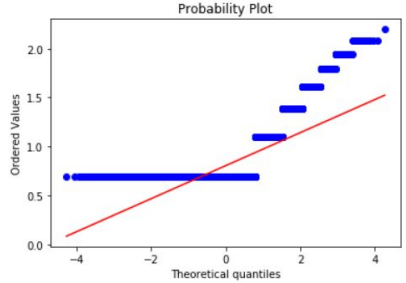
Transformation

QQplot of previous



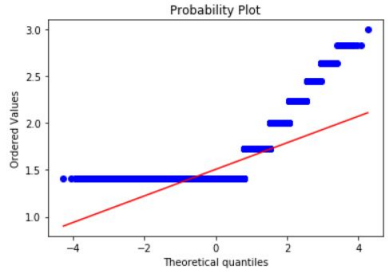
2.862043167736751

QQplot of previous (log)



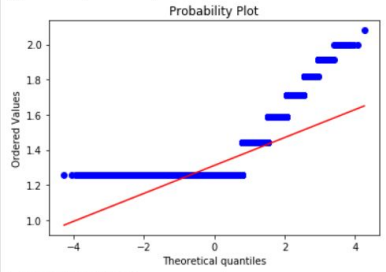
2.076648704299767

QQplot of previous (sqrt)



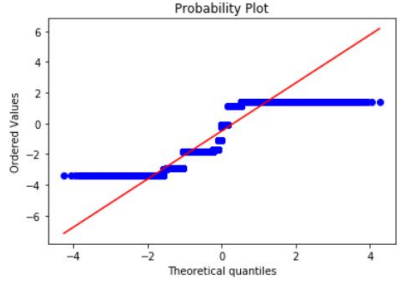
2.3975740834016563

QQplot of previous (cubert)



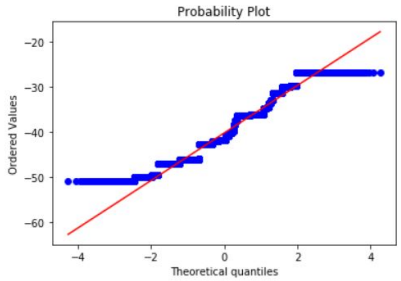
2.277052830776488

QQplot of emp.var.rate



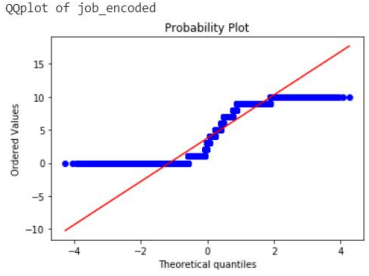
-0.1528190433781906

QQplot of cons.conf.idx



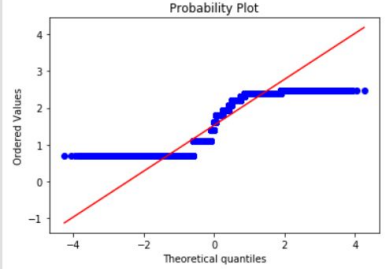
0.3488951118146305

QQplot of job_encoded



0.39763809027860214

QQplot of job_encoded (log)

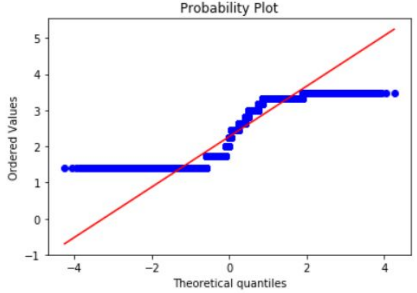


0.025025970129248596

Data Preparation

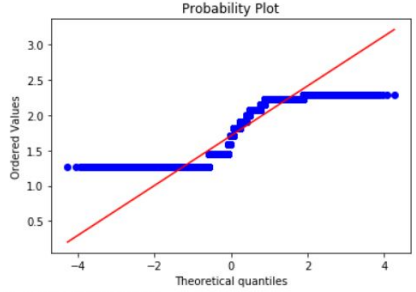
Transformation

QQplot of job_encoded (sqrt)



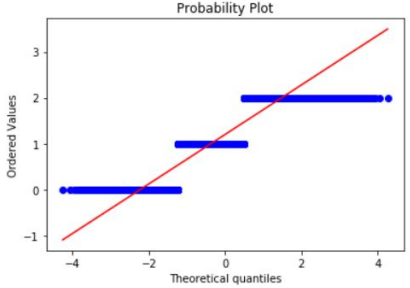
0.2134463853787997

QQplot of job_encoded (cubert)



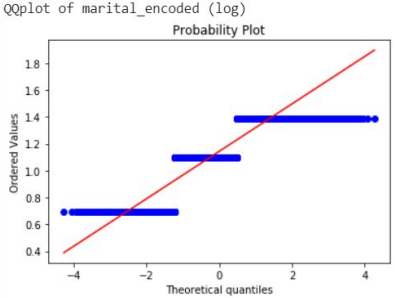
0.15079546000802696

QQplot of marital_encoded



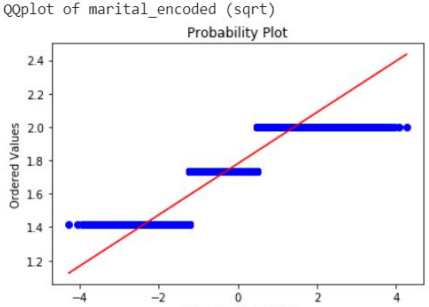
-0.1488878044510951

QQplot of marital_encoded (log)

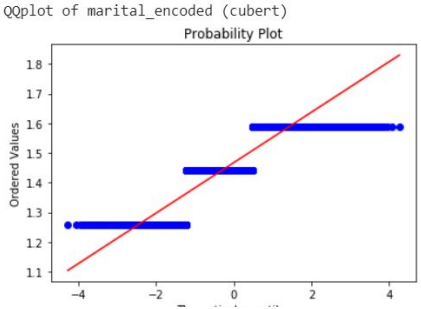


-0.6578195460276522

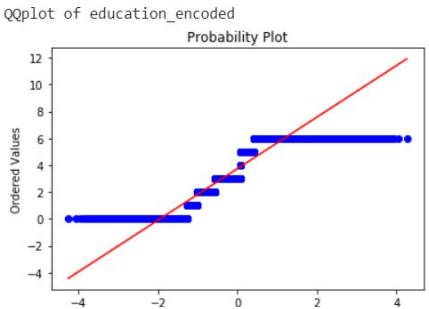
QQplot of marital_encoded (sqrt)



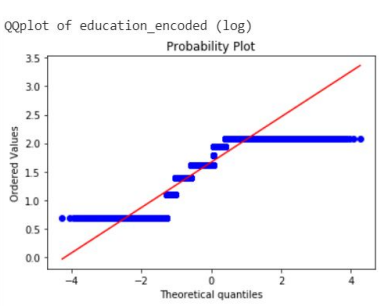
QQplot of marital_encoded (cubert)



QQplot of education_encoded



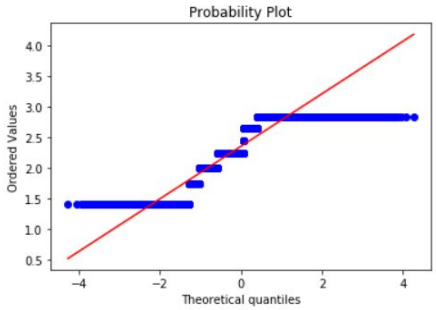
QQplot of education_encoded (log)



Data Preparation

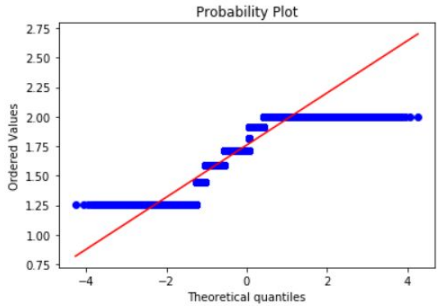
Transformation

QQplot of education_encoded (sqrt)



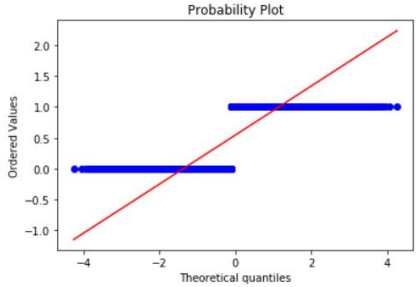
-0.6085065984930068

QQplot of education_encoded (cubert)



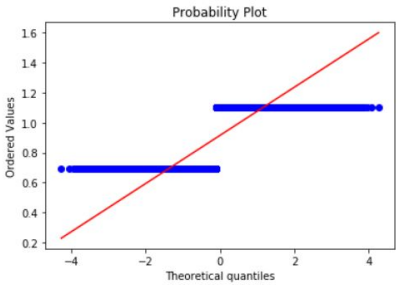
-0.710073925668459

QQplot of housing_encoded



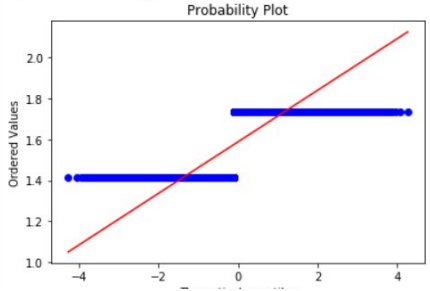
-0.18056886703997457

QQplot of housing_encoded (log)

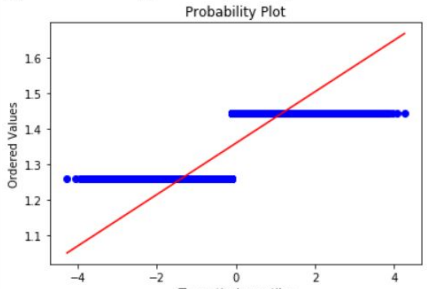


-0.18056886703997385

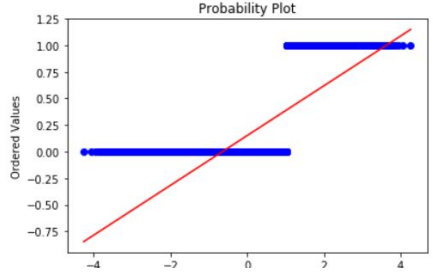
QQplot of housing_encoded (sqrt)



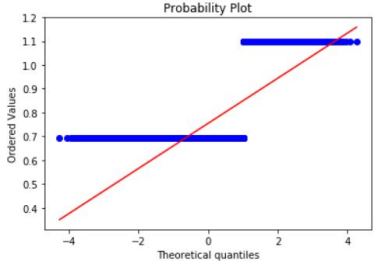
QQplot of housing_encoded (cubert)



QQplot of loan_encoded



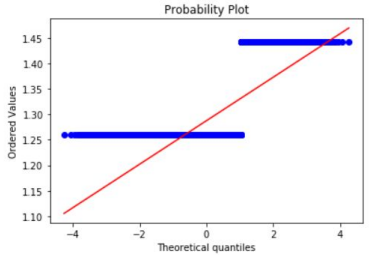
QQplot of loan_encoded (log)



Data Preparation

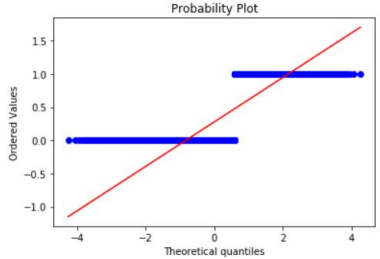
Transformation

QQplot of loan_encoded (cubert)



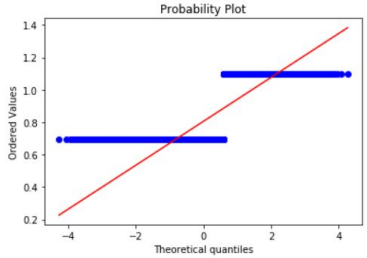
1.9504712631903582

QQplot of contact_encoded



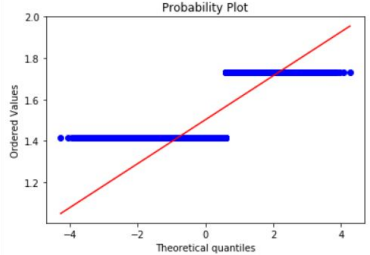
1.000972355553644

QQplot of contact_encoded (log)



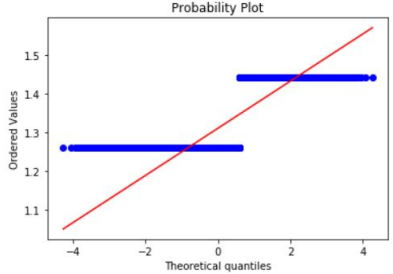
1.000972355553646

QQplot of contact_encoded (sqrt)



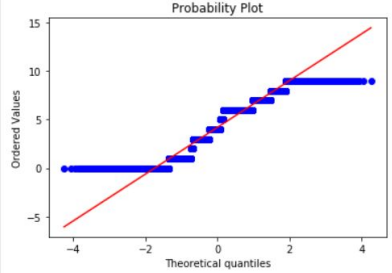
1.0009723555536354

QQplot of contact_encoded (cubert)



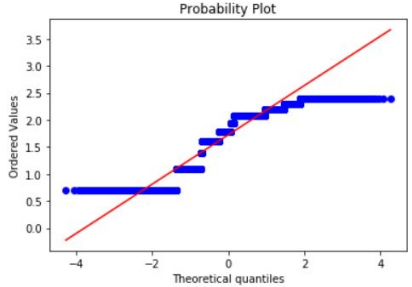
1.0009723555536452

QQplot of month_encoded



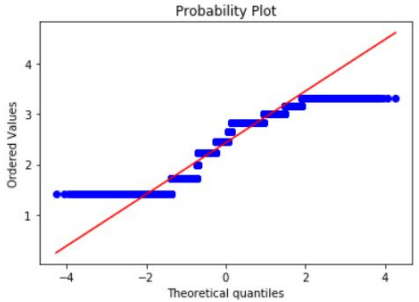
-0.16674756347642256

QQplot of month_encoded (log)



-0.7793713791874137

QQplot of month_encoded (sqrt)

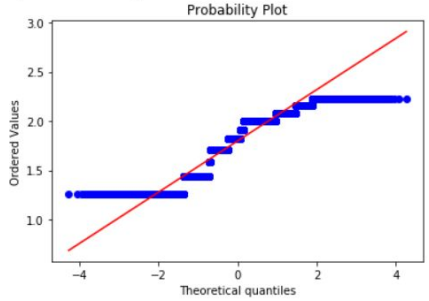


-0.46955849316362713

Data Preparation

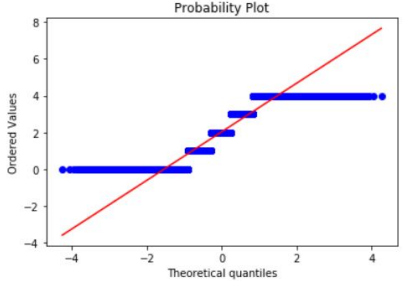
Transformation

QQplot of month_encoded (cubert)



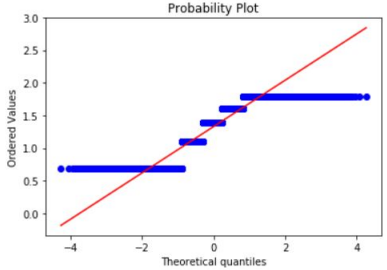
-0.5721393840632223

QQplot of dayofweek_encoded



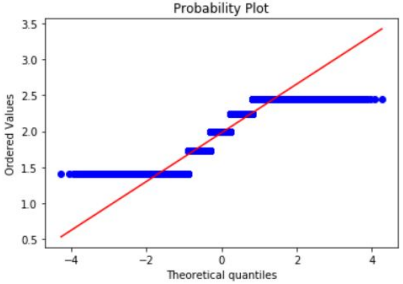
-0.028864332248686085

QQplot of dayofweek_encoded (log)



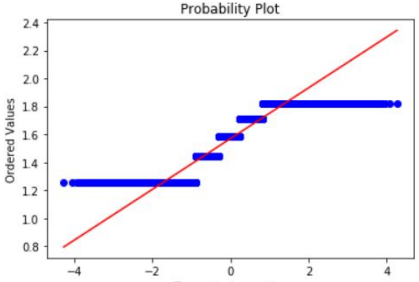
-0.45475135976186587

QQplot of dayofweek_encoded (sqrt)

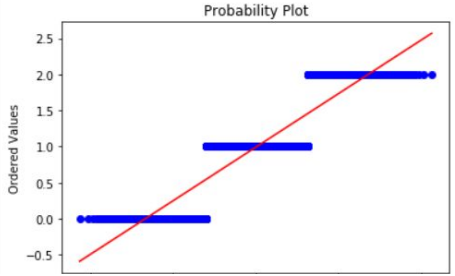


-0.23867190029865268

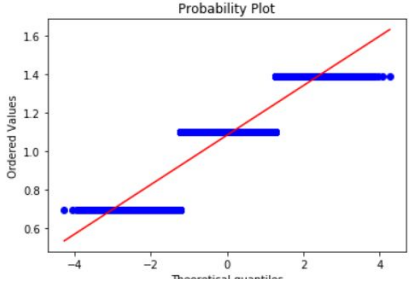
QQplot of dayofweek_encoded (cubert)

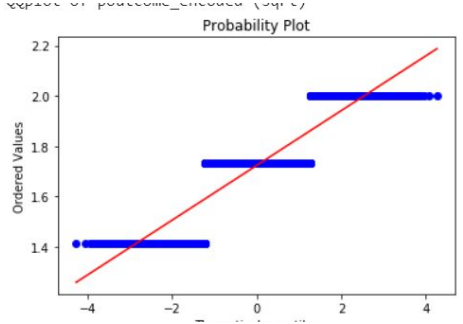


QQplot of poutcome_encoded



QQplot of poutcome_encoded (log)



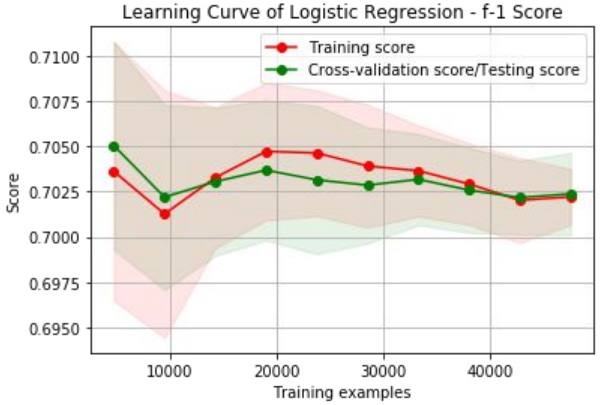


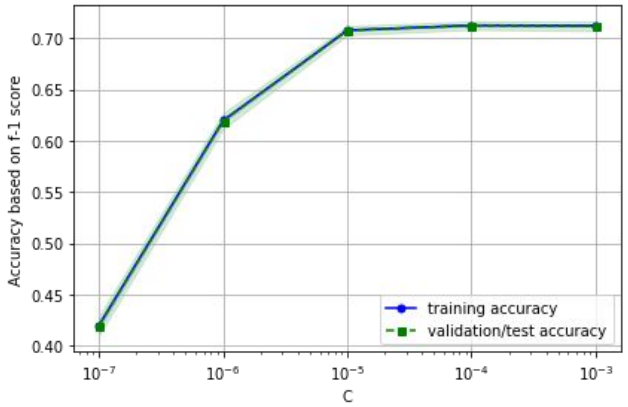
Model – Basic

Logistic Regression

```
(C=1.0, class_weight=None, dual=False, fit_intercept=True,
 intercept_scaling=1, l1_ratio=None, max_iter=100,
 multi_class='warn', n_jobs=None, penalty='l2',
 random_state=42, solver='warn', tol=0.0001, verbose=0,
 warm_start=False)
```

```
Accuracy (out-of-sample): 0.73
Accuracy (in-sample): 0.73
F1 score (out-of-sample):  0.7273684840164378
F1 score (in-sample)    :  0.7292898611050469
Kappa score (out-of-sample):  0.4595633943926605
Kappa score (in-sample)    :  0.4632053834041264
              precision    recall  f1-score   support

           1       0.69      0.82      0.75      8488
           0       0.78      0.64      0.70      8506

    accuracy                           0.73     16994
   macro avg       0.74      0.73      0.73     16994
weighted avg       0.74      0.73      0.73     16994
```

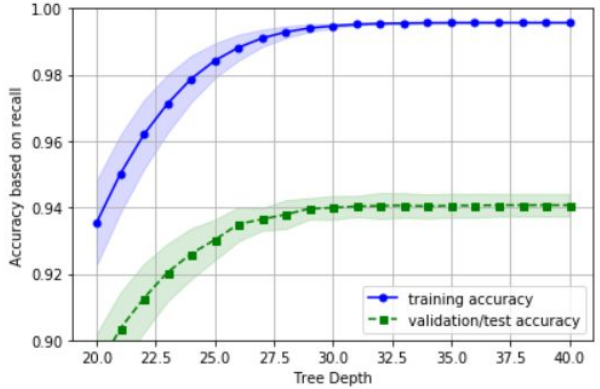




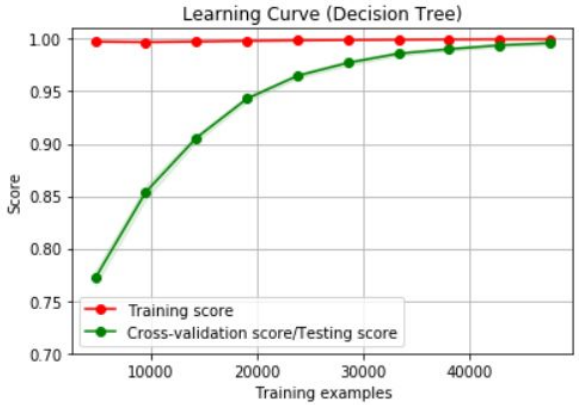
Model – Basic

Decision Tree

```
DecisionTreeClassifier(class_weight=None, criterion='gini', max_depth=None,
            max_features=None, max_leaf_nodes=None,
            min_impurity_decrease=0.0, min_impurity_split=None,
            min_samples_leaf=1, min_samples_split=2,
            min_weight_fraction_leaf=0.0, presort=False, random_state=42,
            splitter='best')
```



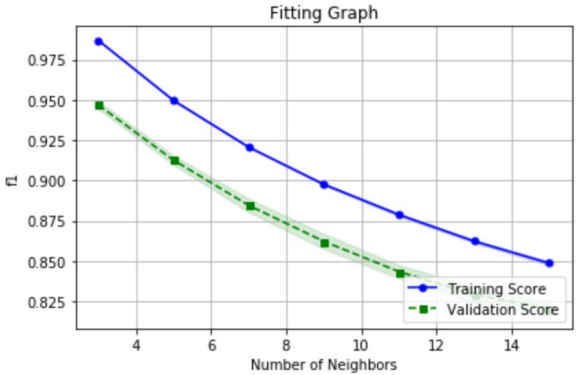

```
Accuracy (out-of-sample): 0.94
Accuracy (in-sample): 1.00
F1 score (out-of-sample):  0.9406749143304347
F1 score (in-sample)    :  0.9956845209777654
Kappa score (out-of-sample):  0.8817090411866169
Kappa score (in-sample)    :  0.9913691884080084
              precision    recall  f1-score   support

           1       1.00      0.89      0.94      8488
           0       0.90      1.00      0.94      8506

   micro avg       0.94      0.94      0.94     16994
   macro avg       0.95      0.94      0.94     16994
weighted avg       0.95      0.94      0.94     16994
```



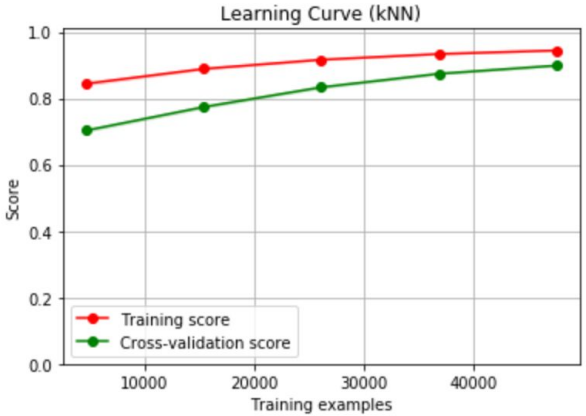
Model – Basic

k-NN

```
from sklearn.preprocessing import StandardScaler
sc = StandardScaler()
sc.fit(X_balance)
X_std = sc.transform(X_balance)

sc.fit(X_train)
sc.fit(X_test)
X_train_std = sc.transform(X_train)
X_test_std = sc.transform(X_test)

knn = neighbors.KNeighborsClassifier(n_neighbors=3,  p=2,  metric='minkowski')
knn = knn.fit(X_train_std, y_train)
```




```
Accuracy (out-of-sample): 0.89
Accuracy (in-sample): 0.94
F1 score (out-of-sample):  0.893595076569042
F1 score (in-sample)    :  0.9432591846601532
              precision    recall  f1-score   support

           0       0.98      0.80      0.88     10188
           1       0.83      0.98      0.90     10205

    accuracy                           0.89     20393
   macro avg       0.91      0.89      0.89     20393
weighted avg       0.91      0.89      0.89     20393
```



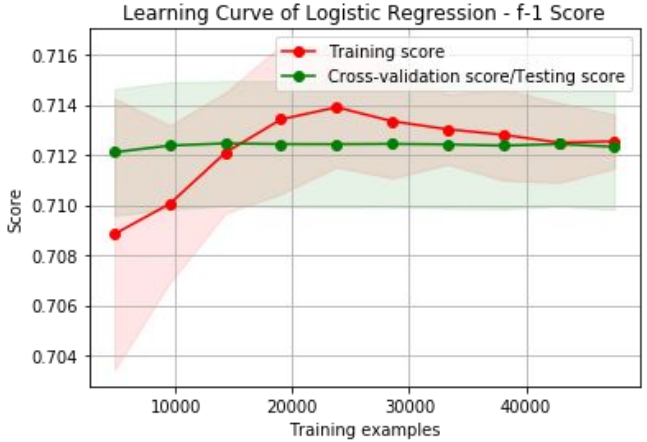
Model – Basic Optimized

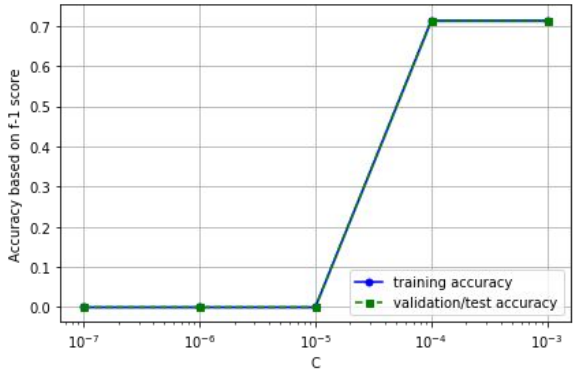
Logistic Regression

Logistic Regression - Parameter Tuning - Penalty and C

```
Non-nested CV f1:  0.7125148349888346
Optimal Parameter:  {'C': 0.001, 'penalty': 'l1'}
Optimal Estimator:  LogisticRegression(C=0.001, class_weight=None, dual=F
alse, fit_intercept=True,
                   intercept_scaling=1, l1_ratio=None, max_iter=100,
                   multi_class='warn', n_jobs=None, penalty='l1',
                   random_state=42, solver='warn', tol=0.0001, verbose=0,
                   warm_start=False)
Nested CV f1: 0.7119024364831615  +/-  0.0033207986498625473
```

```
Accuracy (out-of-sample): 0.71
Accuracy (in-sample): 0.72
F1 score (out-of-sample):  0.712186318772102
F1 score (in-sample)    :  0.7154982860992815
Kappa score (out-of-sample):  0.42439139132807313
Kappa score (in-sample)    :  0.43102868773100245
              precision    recall  f1-score   support

           1       0.71      0.72      0.71      8488
           0       0.71      0.71      0.71      8506

    accuracy                           0.71     16994
   macro avg       0.71      0.71      0.71     16994
weighted avg       0.71      0.71      0.71     16994
```

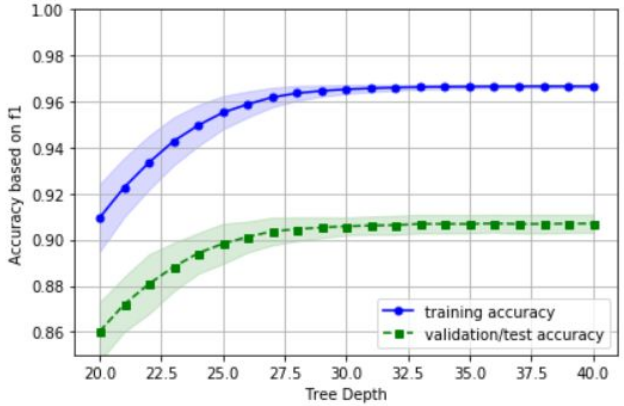




Model – Basic Optimized

Decision Tree

```
Parameter Tuning Decision Tree
Non-nested CV Accuracy:  0.9228161613986926
Optimal Parameter:  {'criterion': 'entropy', 'max_depth': 38, 'min_samples_split': 10}
Optimal Estimator:  DecisionTreeClassifier(class_weight=None, criterion='entropy', max_depth=38,
            max_features=None, max_leaf_nodes=None,
            min_impurity_decrease=0.0, min_impurity_split=None,
            min_samples_leaf=1, min_samples_split=10,
            min_weight_fraction_leaf=0.0, presort=False, random_state=42,
            splitter='best')
Nested CV Accuracy:  0.9227531952564112  +/-  0.0023144015823847214
```



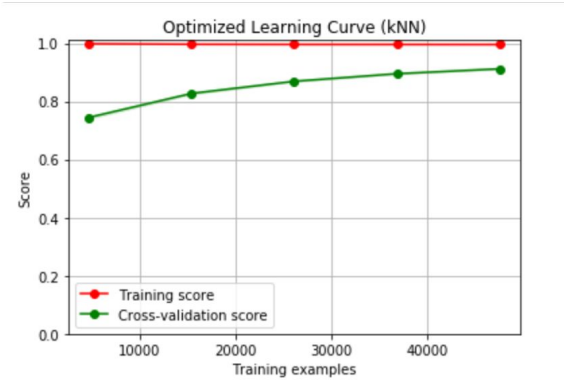

```
Accuracy (out-of-sample): 0.91
Accuracy (in-sample): 0.97
F1 score (out-of-sample):  0.9124466486923817
F1 score (in-sample)    :  0.9704930156347895
Kappa score (out-of-sample):  0.8253314202305714
Kappa score (in-sample)    :  0.9409970306825134
              precision    recall  f1-score   support

           0       0.96      0.86      0.91      8488
           1       0.88      0.96      0.92      8506

   micro avg       0.91      0.91      0.91     16994
   macro avg       0.92      0.91      0.91     16994
weighted avg       0.92      0.91      0.91     16994
```

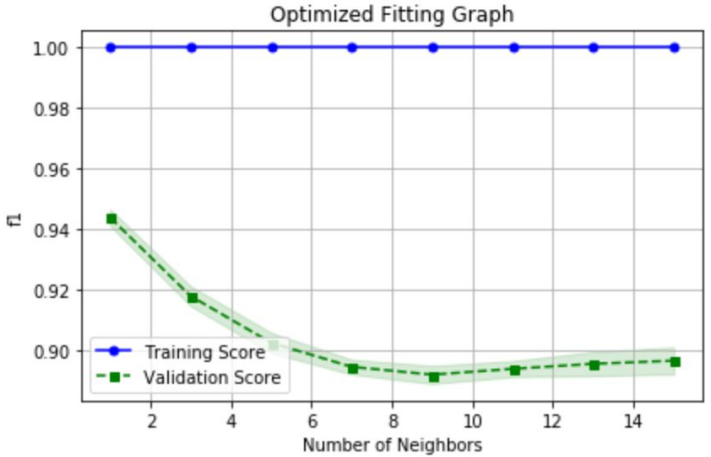
Model – Basic Optimized

k-NN

```
 Parameter Tuning
Non-nested CV f1:  0.924627051158157
Optimal Parameter:  {'n_neighbors': 3, 'p': 1, 'weights': 'distance'}
Optimal Estimator:  KNeighborsClassifier(algorithm='auto', leaf_size=30, metric='minkowski',
                    metric_params=None, n_jobs=None, n_neighbors=3, p=1,
                    weights='distance')
Nested CV f1:  0.9241512824406785  +/-  0.0012833274629558433
```

```
Accuracy (out-of-sample): 0.91
Accuracy (in-sample): 1.00
F1 score (out-of-sample):  0.9091105176721648
F1 score (in-sample)    :  0.9955023549271698
              precision    recall  f1-score   support

           0       0.99      0.83      0.90     10188
           1       0.85      0.99      0.92     10205

    accuracy                           0.91     20393
   macro avg       0.92      0.91      0.91     20393
weighted avg       0.92      0.91      0.91     20393
```





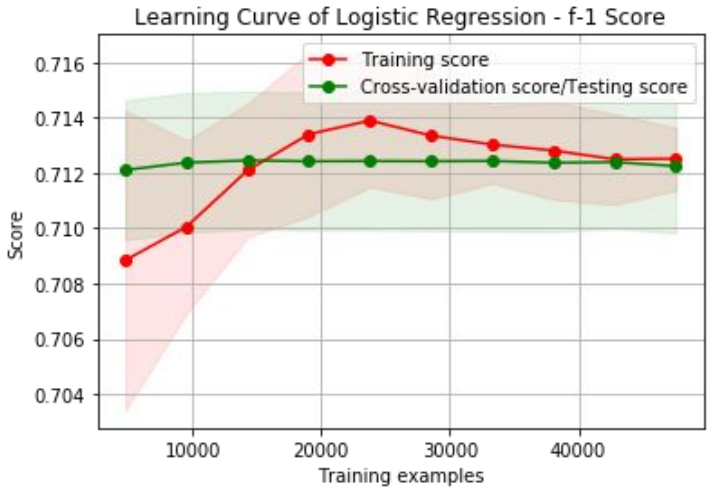
Model – Transformed Optimized

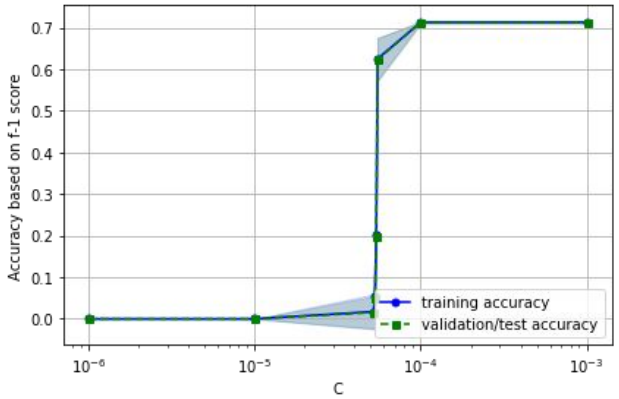
Logistic Regression

Logistic Regression - Parameter Tuning - Penalty and C

```
Non-nested CV f1:  0.7125342436071481
Optimal Parameter:  {'C': 0.001, 'penalty': 'l1'}
Optimal Estimator:  LogisticRegression(C=0.001, class_weight=None, dual=F
alse, fit_intercept=True,
                   intercept_scaling=1, l1_ratio=None, max_iter=100,
                   multi_class='warn', n_jobs=None, penalty='l1',
                   random_state=42, solver='warn', tol=0.0001, verbose=0,
                   warm_start=False)
Nested CV f1: 0.7120749976817567  +/-  0.0031682187286711473
```

```
Accuracy (out-of-sample): 0.71
Accuracy (in-sample): 0.72
F1 score (out-of-sample):  0.7122450059115002
F1 score (in-sample)    :  0.7156545110892298
Kappa score (out-of-sample):  0.4245091503703403
Kappa score (in-sample)    :  0.43134247498793743
              precision    recall  f1-score   support

           1       0.71      0.72      0.71      8488
           0       0.72      0.71      0.71      8506

    accuracy                           0.71     16994
   macro avg       0.71      0.71      0.71     16994
weighted avg       0.71      0.71      0.71     16994
```

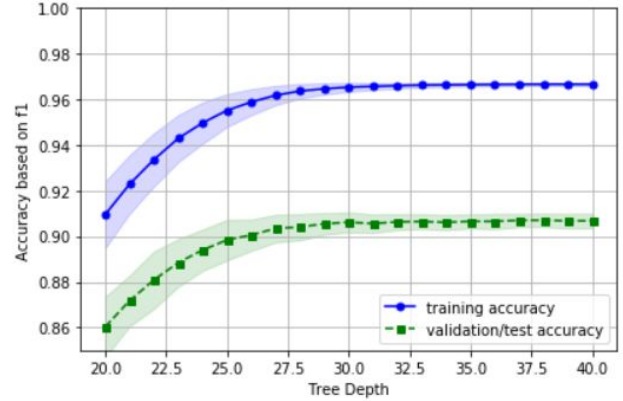




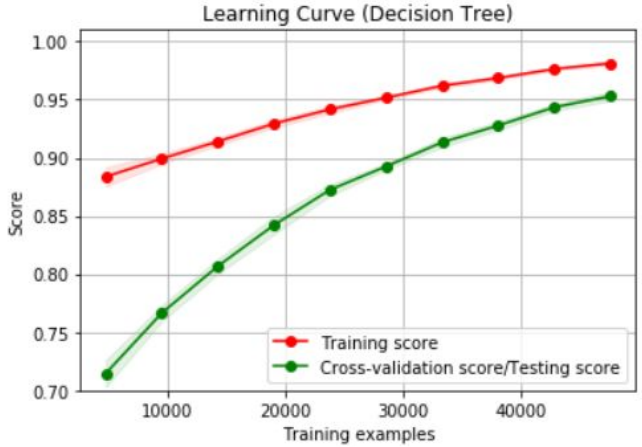
Model – Transformed Optimized

Decision Tree

```
Parameter Tuning Decision Tree
Non-nested CV Accuracy:  0.9229513440866125
Optimal Parameter:  {'criterion': 'entropy', 'max_depth': 36, 'min_samples_split': 10}
Optimal Estimator:  DecisionTreeClassifier(class_weight=None, criterion='entropy', max_depth=36,
            max_features=None, max_leaf_nodes=None,
            min_impurity_decrease=0.0, min_impurity_split=None,
            min_samples_leaf=1, min_samples_split=10,
            min_weight_fraction_leaf=0.0, presort=False, random_state=42,
            splitter='best')
Nested CV Accuracy:  0.922594831084852  +/-  0.0022775259142253606
```



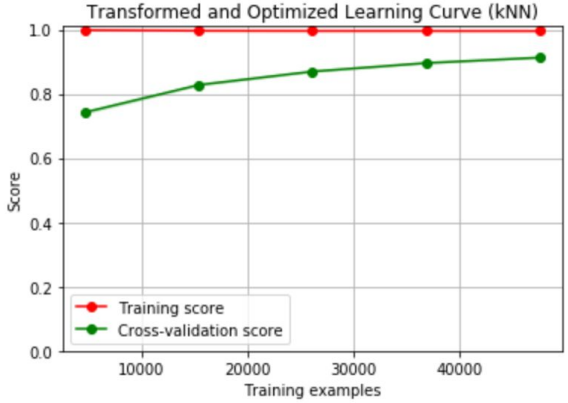

```
Accuracy (out-of-sample): 0.91
Accuracy (in-sample): 0.97
F1 score (out-of-sample):  0.9124466486923817
F1 score (in-sample)    :  0.9704930156347895
Kappa score (out-of-sample):  0.8253314202305714
Kappa score (in-sample)    :  0.9409970306825134
              precision    recall  f1-score   support

           0       0.96      0.86      0.91      8488
           1       0.88      0.96      0.92      8506

   micro avg       0.91      0.91      0.91     16994
   macro avg       0.92      0.91      0.91     16994
weighted avg       0.92      0.91      0.91     16994
```



Model - Transformed Optimized

k-NN

```
 Parameter Tuning
Non-nested CV f1:  0.925409737700338
Optimal Parameter:  {'n_neighbors': 3, 'p': 1, 'weights': 'distance'}
Optimal Estimator:  KNeighborsClassifier(algorithm='auto', leaf_size=30, metric='minkowski',
                     metric_params=None, n_jobs=None, n_neighbors=3, p=1,
                     weights='distance')
Nested CV f1:  0.9254097596296372  +/-  0.001775138837325355
```




```
Accuracy (out-of-sample): 0.91
Accuracy (in-sample): 1.00
F1 score (out-of-sample):  0.9096211462899034
F1 score (in-sample)    :  0.9954603258981869
              precision    recall  f1-score   support

           0       0.99      0.83      0.90     10188
           1       0.85      0.99      0.92     10205

    accuracy                           0.91     20393
   macro avg       0.92      0.91      0.91     20393
weighted avg       0.92      0.91      0.91     20393
```

I). About potential bank client characteristics:

1. “age” (numeric)
2. “job” : type of job (categorical: 'admin.', 'blue-collar', 'entrepreneur', 'housemaid', 'management', 'retired', 'self-employed', 'services', 'student', 'technician', 'unemployed', 'unknown')
3. “marital”: marital status (categorical: 'divorced'(meaning divorced or widowed), 'married', 'single', 'unknown’)
4. “education” (categorical: 'basic.4y', 'basic.6y', 'basic.9y', 'high.school', 'illiterate', 'professional.course', 'university.degree', 'unknown')
5. “default”: has credit in default or not? (categorical: 'no','yes','unknown')
6. “housing”: has housing loan or not? (categorical: 'no','yes','unknown')
7. “loan”: has personal loan or not? (categorical: 'no','yes','unknown')

II). About the last contact of the current campaign:

8. “contact”: contact communication type (categorical: 'cellular','telephone')
9. “month”: last contact month of year (categorical: 'jan', 'feb', 'mar', ..., 'nov', 'dec')
10. “day_of_week”: last contact day of the week (categorical: 'mon','tue','wed','thu','fri')

III). About historical campaigns:

11. “campaign”: number of contacts performed during this campaign and for this client (numeric, includes last contact)
12. “pdays”: number of days that passed by after the client was last contacted from a previous campaign (numeric; 999 means client was not previously contacted)
13. “previous”: number of contacts performed before this campaign and for this client (numeric)
14. “poutcome”: outcome of the previous marketing campaign (categorical: 'failure','nonexistent','success')

IV). About social and economic context:

15. “emp.var.rate”: employment variation rate - quarterly indicator (numeric)
16. “cons.price.idx”: consumer price index - monthly indicator (numeric)
17. “cons.conf.idx”: consumer confidence index - monthly indicator (numeric)
18. “euribor3m”: euribor 3 month rate - daily indicator (numeric)
19. “nr.employed”: number of employees - quarterly indicator (numeric)