Wave-2, the

Upsell

What we know:

- 75,000 Sample Size
- Basic Demographics
- Purchased Tax Software
- Linear Regression Results

What we want to know:

- Who will buy out of the remaining 760,000
- New models

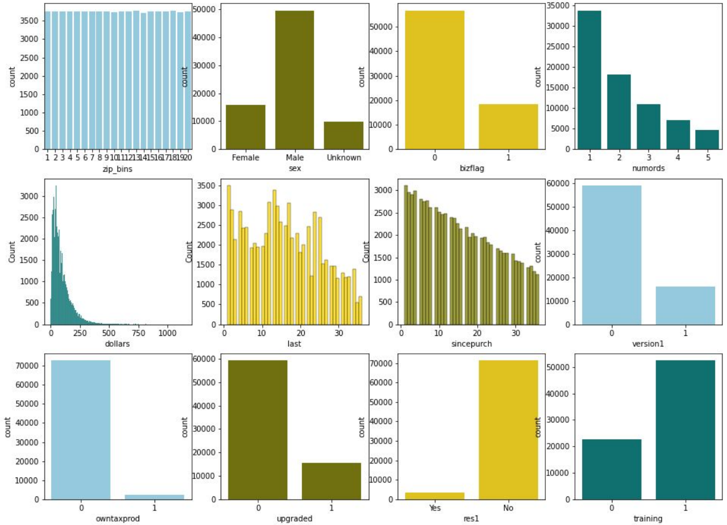
Table of Contents

01 Exploring the Data

02 Models and Hyperparameter Tuning

03 Model Performance

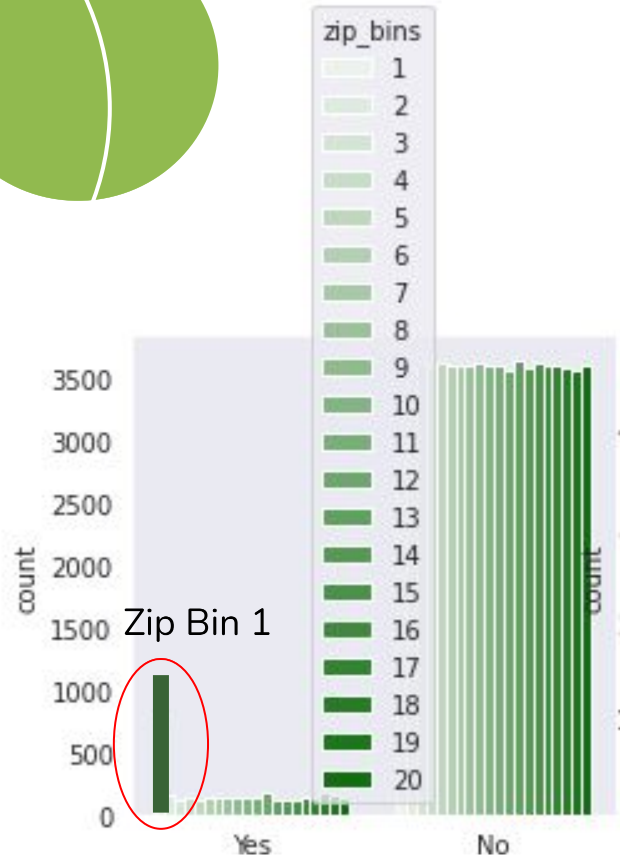
04 Decision & Next Steps



01 Exploring the Data

Dummy Variables / 3-Factor labels
Transforming Dollars

ZIP Bins & Dummy Variables

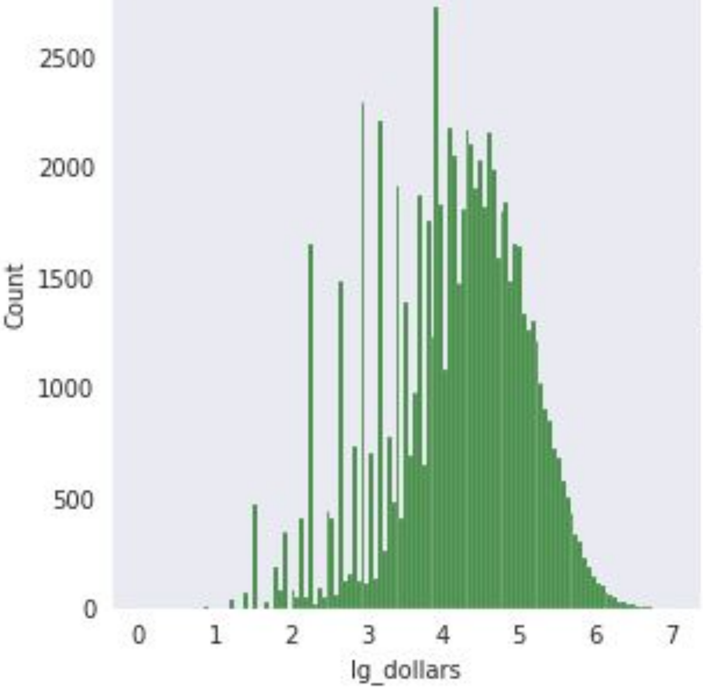


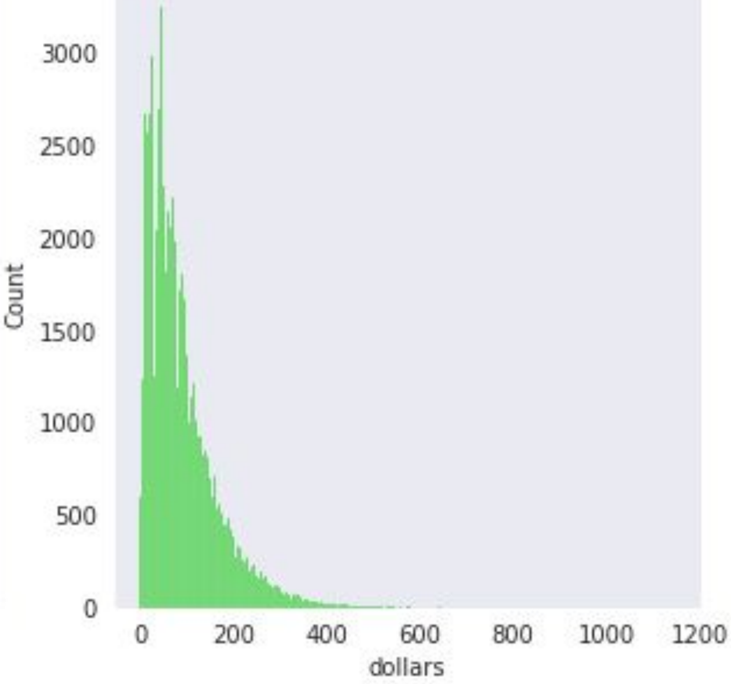
- Zip Bins
- Zip ‘00801’ & ‘00804’
- As New Dummy Variables

‘Upgraded’ & ‘Version 1’ 3 Factor Label

	Quickbooks Version 1 Only	Version 1 to Version 2	Quickbooks Version 2 Only
	👤	👤 → 👤	👤
	Upgraded = 0 **Version 1 = 0**	**Upgraded = 1** **Version 1 = 0**	**Upgraded = 1** **Version 1 = 1**
New Variable:	only_ver02	from_v1_to_v02	only_ver01

Log Transform Dollars





02

Models and hyperparameters

Modeling Methods / Tuning Hyperparameters

Methods we used

Keras Neural Network

MLP Neural Network

Decision Tree Classifier

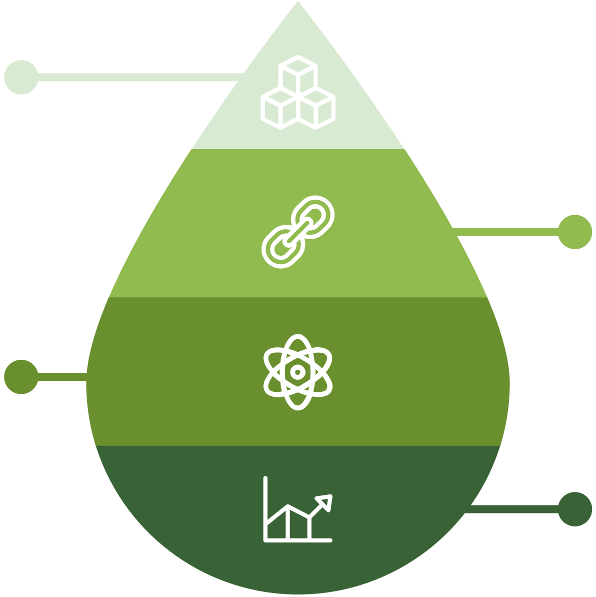
Random Forest Classifier

XGBoost Classifier

Neural Networks to find and capture underlying relationships

Tree based models to capture predictions on highly non-linear and complex relationships

Method 1: Keras Neural Network



1. Specify Architecture

- ***Keras Sequential*** Model
- 100 to 300 nodes each layer
- 4 to 6 dense layers
- ***Relu*** activation
- ***Softmax*** activation

2. Compile

- ***Adam*** optimizer
- ***Categorical_crossentropy*** loss function
- ***Accuracy*** metrics

3. Fit

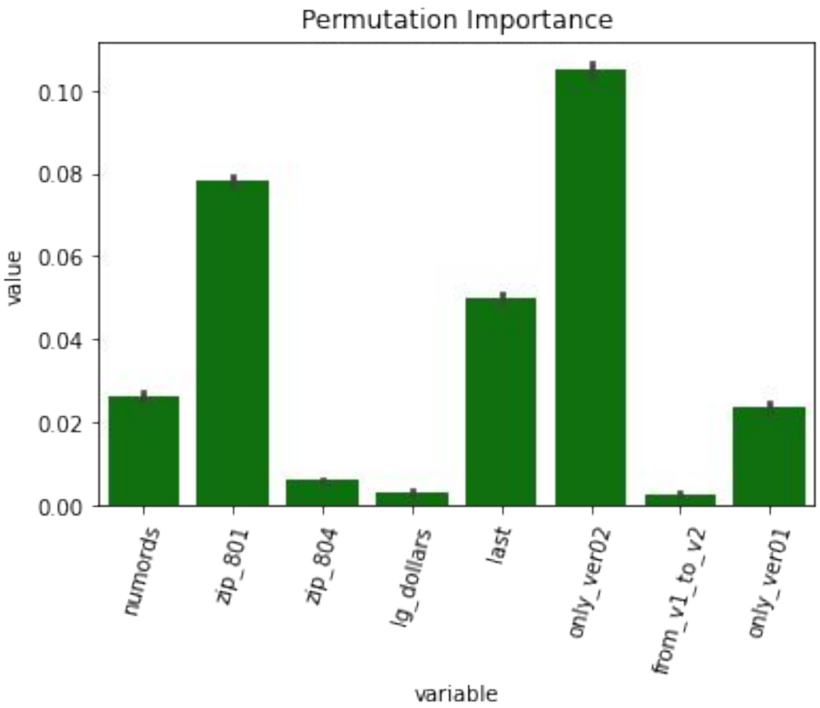
- Train on 52,500 data
- 30% validation split
- ***Early Stopping*** callbacks
- 20 epochs

4. Predict

- Predict 22,500 test data
- Breakeven threshold

Best AUC Score: 76.5%

Method 2: MLP Neural Network



Activation = "tanh"
Solver = "lgfbs"
Max iter = 1000

Hidden Layer Sizes?

1 Layer: (1,), (2,), **(3,)**, (4,)

2 Layers: (1, 1), (2, 2), (3, 3), (4, 4)

Alpha?

0.001, 0.01, 0.05

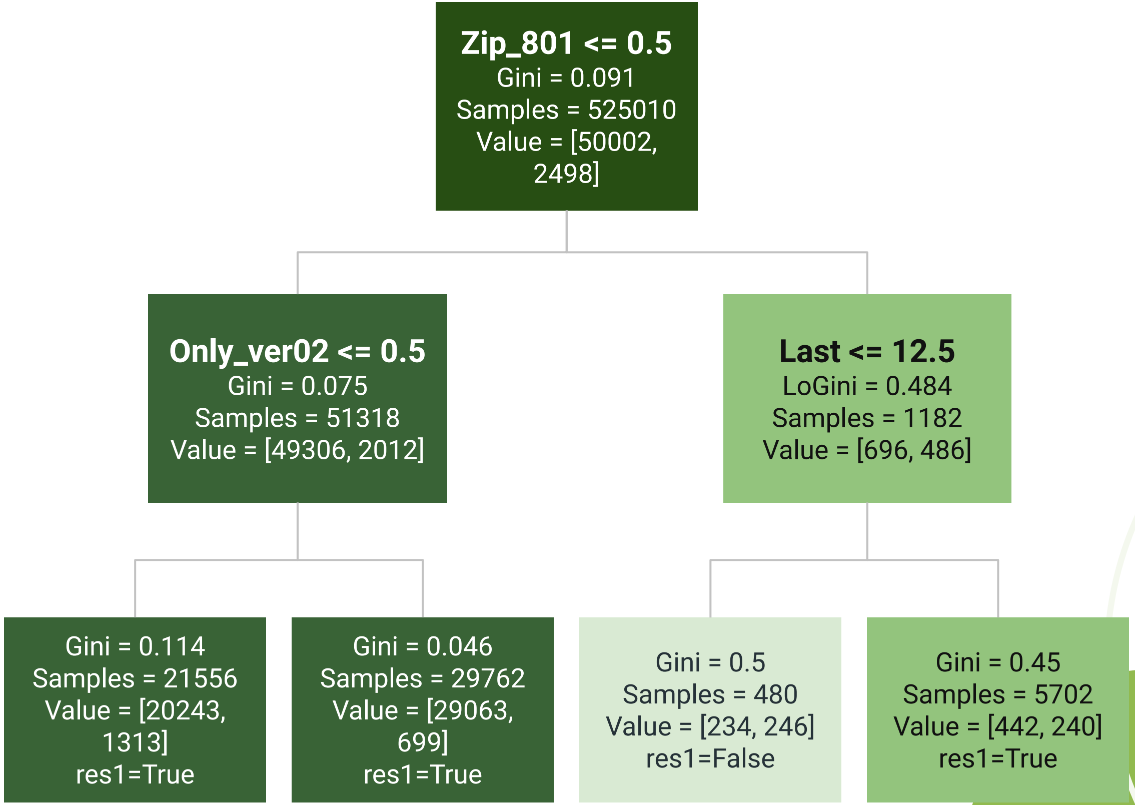
Best AUC Score: 77.02%

Method 3: Decision Tree Classifier

**GridSearchCV:
5 Folds**

AUC = 75%

Max depth = 6



Zip_801 <= 0.5
Gini = 0.091
Samples = 525010
Value = [50002, 2498]

Only_ver02 <= 0.5
Gini = 0.075
Samples = 51318
Value = [49306, 2012]

Last <= 12.5
LoGini = 0.484
Samples = 1182
Value = [696, 486]

Gini = 0.114
Samples = 21556
Value = [20243, 1313]
res1=True

Gini = 0.046
Samples = 29762
Value = [29063, 699]
res1=True

Gini = 0.5
Samples = 480
Value = [234, 246]
res1=False

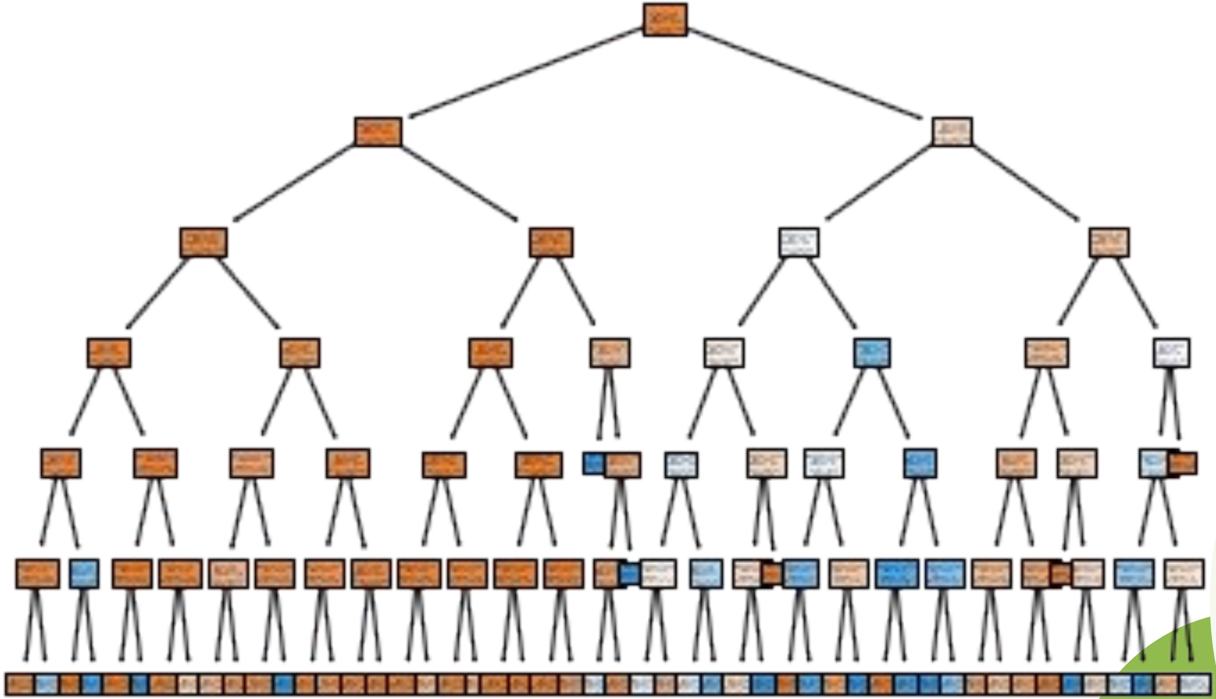
Gini = 0.45
Samples = 5702
Value = [442, 240]
res1=True

Method 3: Decision Tree Classifier

GridSearchCV: 5 Folds

AUC = 75%

Max depth = 6



Method 4: Random Forest Classifier

1. Bootstrapping
2. Ensembling

max_features

[**auto**, sqrt]

n_estimators

[**50**, 100]

max_depth

[2, ,**5** 10]

min_samples_split

[**2**, 5]

bootstrap

[**True**, False]

min_samples_leaf

[**1**, 2, 4]

GridSearchCV with 4 folds:

Best AUC Score: 76.64%

Method 5: XGBoost Classifier

- ***Residuals*** *= target – prediction*
- ***Output*** *= average of residuals*
- $\mathbf{Pred}_i = Pred_{(j-1)} + lr * output$

Tuning parameters

- **objective**: reg:logistic
- **colsample_bytree**: 0.3, 0.7, 1
- **n_estimators**: 50, 100, 150
- **max_depth**: 2, 5, 7, 10
- **eta**: 0.01,0.1,0.5,0.9

Cross validation

- GridSearchCV
- 4 folds

Best parameters

- **objective**: reg:logistic
- **colsample_bytree**: 1
- **n_estimators**: 100
- **max_depth**: 2
- **eta**: 0.1

Best AUC score

- **76.98%**

03

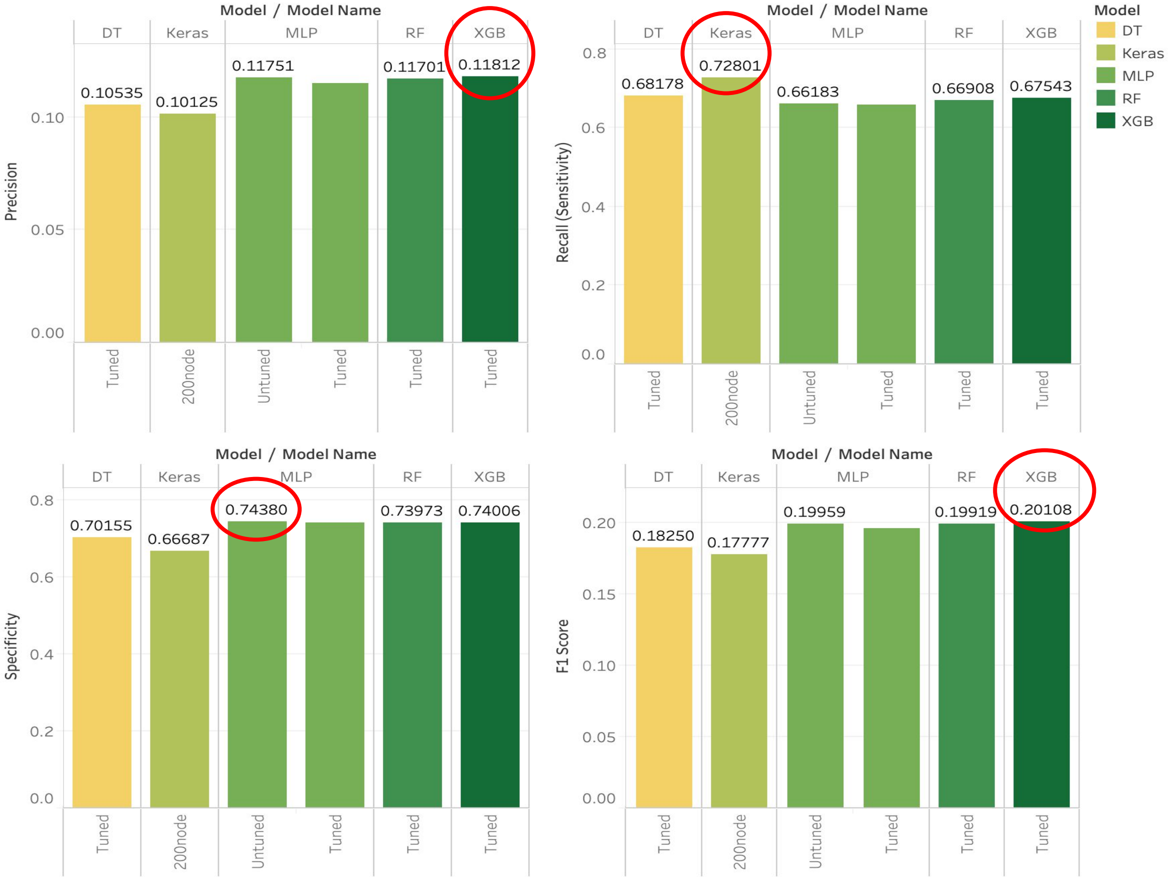
Model Performance

Accuracy / Gains / ROC / Test Set Profit

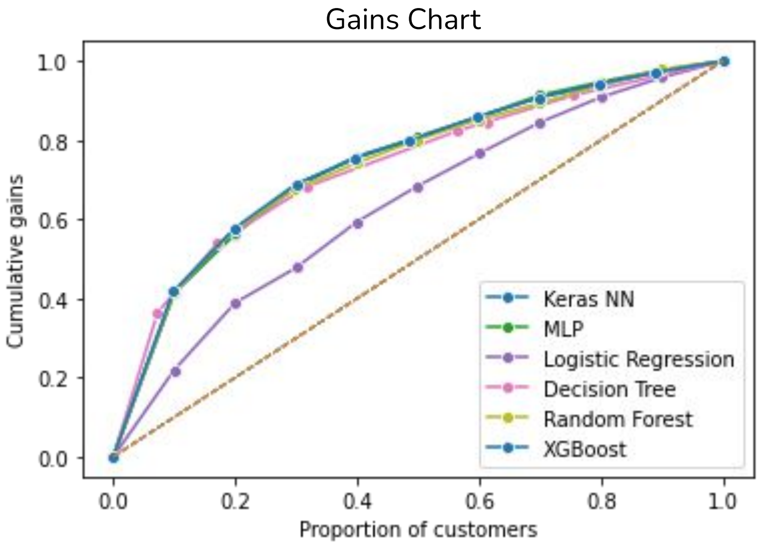
Confusion Matrices for Different Models

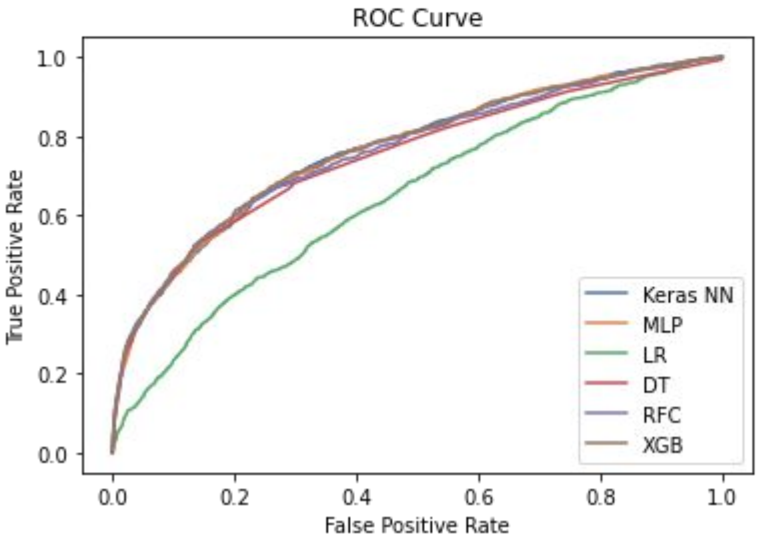
	Keras NN	MLP NN		XGBoost	Random Forest	Decision Tree
	200 Node 4 Layer	Untuned (1,) alpha= 0.01	Tuned (3,) alpha= 0.001	Tuned eta=0.1 max_depth=2 estimators=100	Tuned Max depth=5 estimators=50 bootstrap=True	Tuned Max depth = 6
TN %	63.42	70.73	70.29	70.38	70.35	66.72
TP %	3.57	3.24	3.23	3.31	3.28	3.34
FN %	1.33	1.66	1.68	1.59	1.62	1.56
FP %	31.68	24.36	24.81	24.72	24.75	28.38
ACU%	66.99	73.98	73.52	**73.69**	73.63	70.06

Model Performance Comparison



Gains Chart & ROC Curve


Gains Chart


ROC Curve

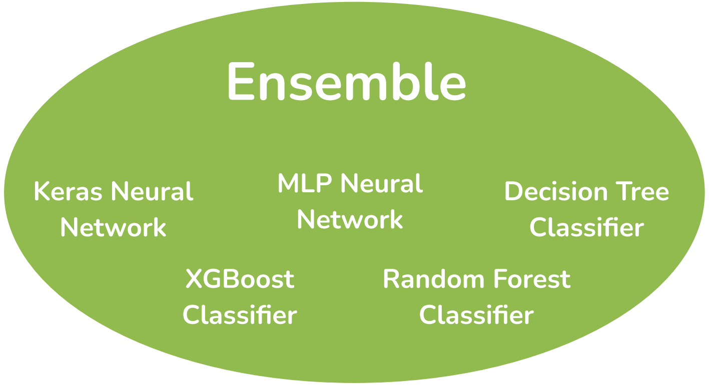
Profit Comparison

	Keras NN	MLP NN		XGBoost	Random Forest	Decision Tree
	200 Node 4 Layer	(1,) alpha=0.01	Tuned (3,) alpha= 0.001	Tuned eta=0.1 max_depth=2 estimators=100	Tuned Max depth=5 estimators=50 bootstrap=True	Tuned Max depth = 6
RR %	10.12%	11.7%	11.7%	11.8%	17.7%	10.5%
ROME %	115.42%	146.17%	148.7%	**151.3%**	149.0%	124.2%
$ Profit (test set)	12.3k	12.5k	12.2k	**12.8k**	12.6k	11.9k

Ensemble vs. XGBoost

XGBoost	**Ensemble Highest Prediction**
73.7%	73.1%
3.31%	3.37%
12.86k	**12.99k**

Lower Accuracy
Higher True Positive Rate
Higher Expected Profit



Ensemble

Keras Neural Network

MLP Neural Network

Decision Tree Classifier

XGBoost Classifier

Random Forest Classifier

04

Adopting the Model

Wave 2 Results / Future Projects

73%

Accurate

150%

ROME

$441,000

Dollars in Profit

Thank You

intuit
mint

intuit
Accountants

intuit
TurboTax®