Thera Bank
Loan Offer Classification

Question

Can we reduce costs of marketing campaigns?

Question

Can we reduce costs of marketing campaigns?

Can we increase conversion rate on personal loans we offer?

Question

Can we reduce costs of marketing campaigns?

Can we increase conversion rate on personal loans we offer?

Can we use our data to predict which of our existing customers will accept a personal loan?

Machine Learning Solution

- Predictive - 98.3% accuracy on last year’s marketing campaign

Machine Learning Solution

- Predictive - 98.3% accuracy on last year’s marketing campaign
- Insightful - Observe which customers are likely to accept an offer and which aren’t

Machine Learning Solution

- Predictive - 98.3% accuracy on last year’s marketing campaign
- Insightful - Observe which customers are likely to accept an offer and which aren’t
- Strategic - Proactively market to the correct segment of customers

Current Process

- Compile information on all existing customers
- Send loan offer to all customers
- Track which customers accept and which reject

Current Process

- Compile information on all existing customers
- Send loan offer to all customers
- Track which customers accept and which reject

Issues

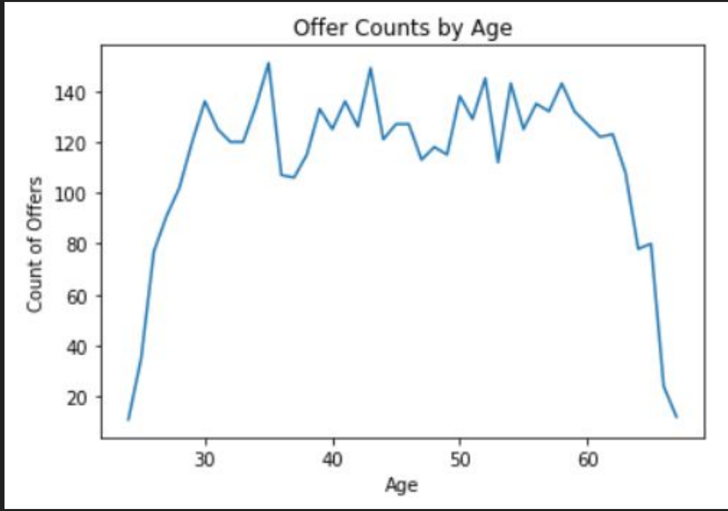
- High costs with sending to all of our customers
- Data is not driving marketing decisions

Data

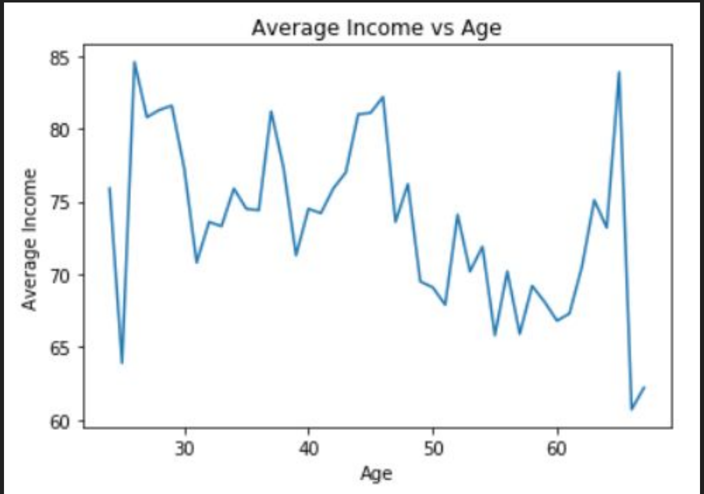
- Marketing campaign from last year (5,000 loan offers made)
- Details on age, number of years working, income, level of education, etc.
- Ignores zip code and observations with negative years of work experience

Analysis

Approx 100-140 offers were made for each age between 30 and 60

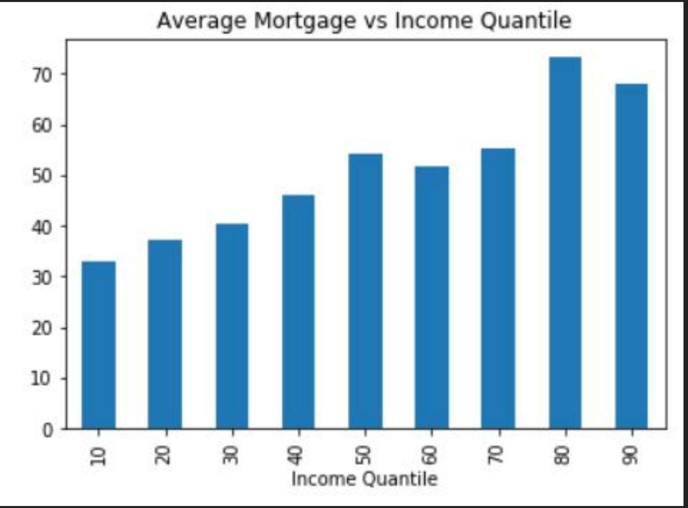


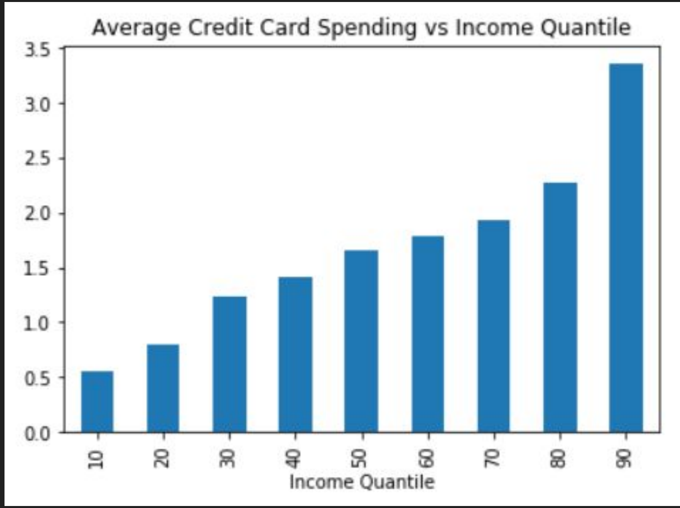
Large variability in income by age



Analysis

- Higher income customers have larger mortgages and credit card spending





Model

- Predict whether or not customer will accept or reject loan offer
- Gradient boosting with hyperparameter tuning
  - Learning rate
  - Number of estimators

Model

- Data is split with 75% being used for training and 25% for testing
- Data is scaled appropriately
- To address class imbalance, synthetic observations are incorporated into the model

Results

After hyperparameter tuning, training set performs with 99.1% accuracy

```
param_grid = {'learning_rate':[0.05, 0.1, 0.25, 0.5, 0.75, 1], 'n_estimators':[25, 50, 75, 100, 150, 200, 250]}
gb_cv = GridSearchCV(GradientBoostingClassifier(random_state = 5), param_grid = param_grid)
gb_cv.fit(X_train_res, y_train_res)
print("Best Score:" + str(gb_cv.best_score_))
print("Best Parameters: " + str(gb_cv.best_params_))
```

```
Best Score:0.991314449752181
Best Parameters: {'learning_rate': 1, 'n_estimators': 75}
```

Results

- When applied to testing set, we correctly identify 1,216 observations out of 1,237.
- Of the 107 accepted loan offers, model correctly identifies 99 (92.5%)
- Conversion rate of offers made increases to 88.4% (99/112)

	0	1
0	1117	8
1	13	99

Conclusion

- Classifying responses reduces costs and increases conversion
    - Loan offers sent to approximately 9% of customers
    - Conversion rate increases to 88.4%
- Opportunities for improvement - update data on existing customers, incorporate cost function into score optimizing