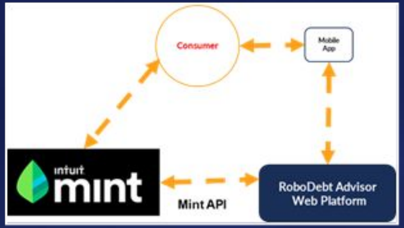


RoboDebt Advisor

Using Machine Learning for Financial Education

Gia R. NATHAN

11th Grade, James Logan High School, Union City
California 03/28/2023

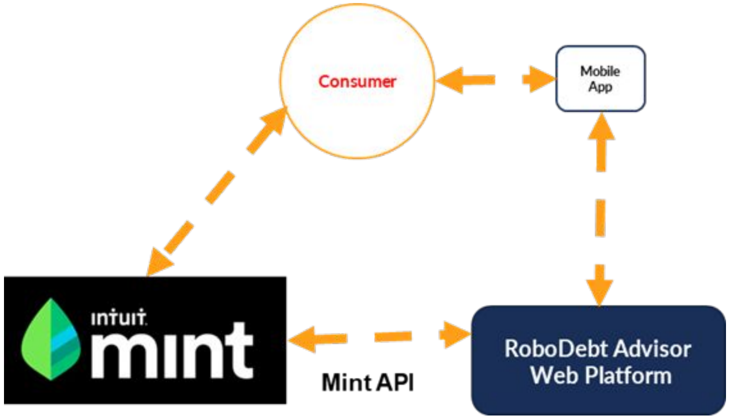
Humanize AI Challenge

AGENDA

- Application
- Problem Statement
- Possible Solution
- Solution Overview
- Methodology Used
- Machine Learning Model
- Results
- Summary

Project Purpose : RoboDebt Advisor: A Fintech Application for Financial Education

- Consumer Debt is a major problem in the USA
- However most consumers do not have the means to hire a financial advisor
- I want to leverage the power of AI/ML in building
  - A fintech application that acts like a virtual financial advisor
  - It woud help a consumer to recover from a financial debt by promoting financial education
  - This is one of the ways we can use AI/ML for society's benefit

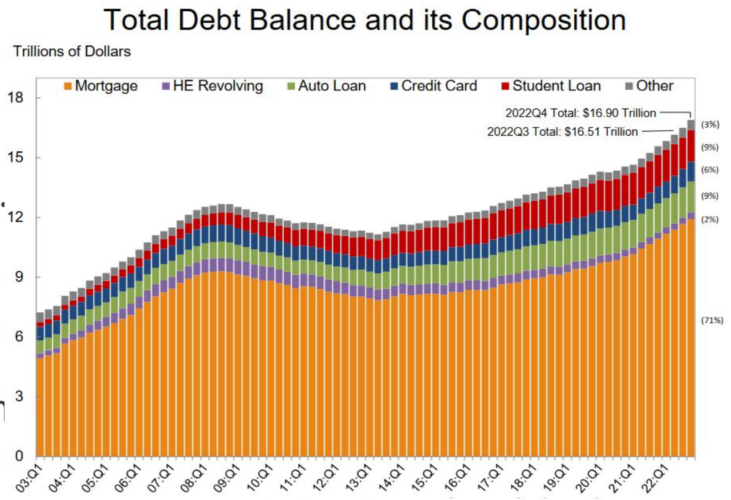


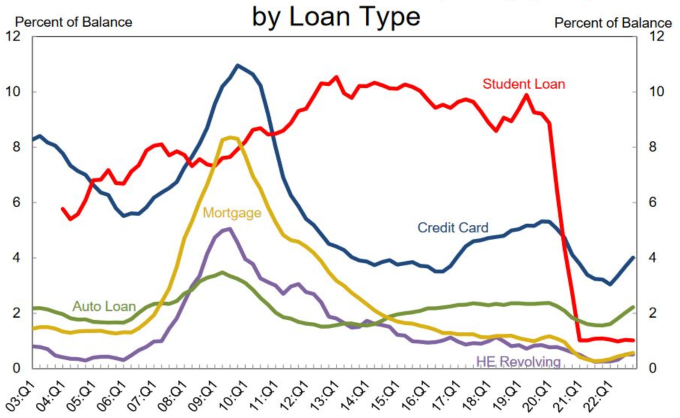
Problem Statement

Consumer Debt in USA

Problem Statement: Consumer Debt in the USA

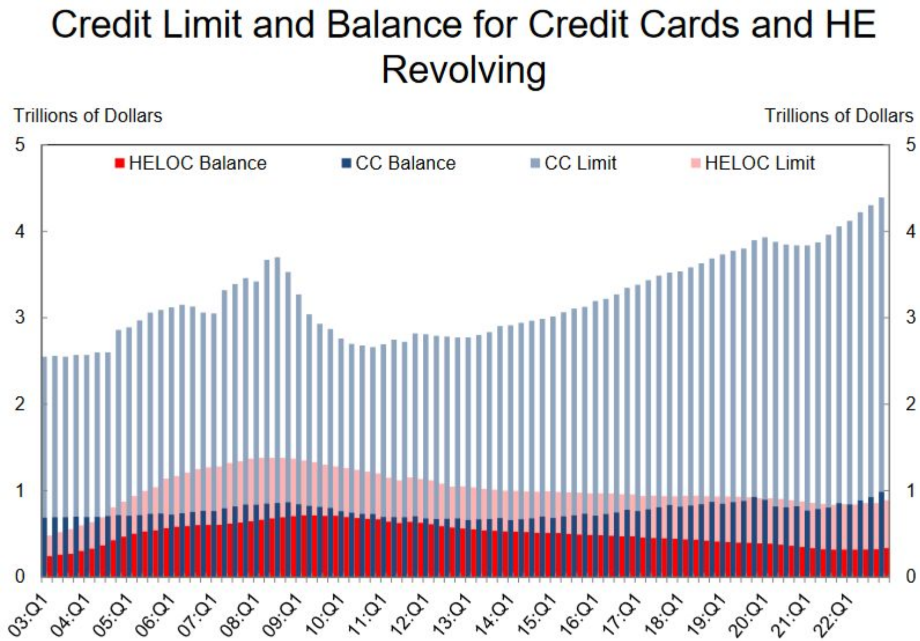
- Consumer debt in America was approximately $16.51 Trillion in Q3 2022*
- The Credit Card debt across USA, was roughly $0.93 Trillion.
- Majority of the consumers with credit card debt have their finances out of control.
- Often end up making decisions that leads them into taking on more debt.
- Given a chance they would like to get out this situation
- For putting in a mitigation plan those consumers:
  - Just don't have the financial education
  - Or funds necessary to hire a financial planner

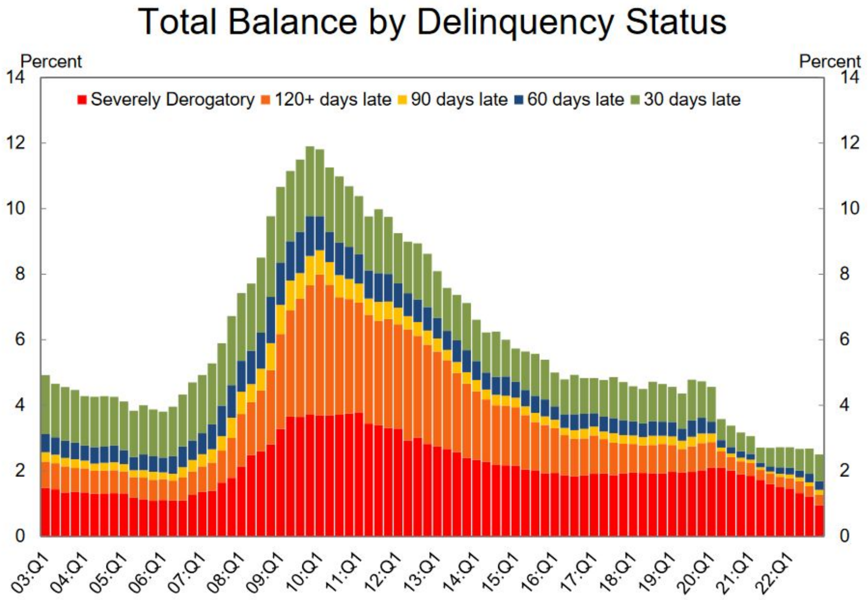




*n.b:Data & Diagrams Source: Federal Reserve Bank of New York

Problem Statement: Consumer Debt in the USA





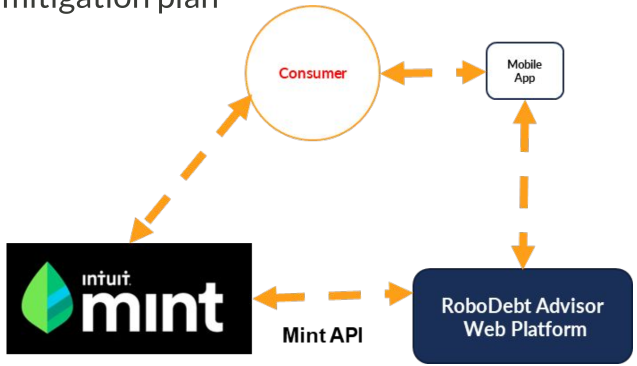
*n.b:Data & Diagrams Source: Federal Reserve Bank of New York

Solution Overview

RoboDebt Advisor

My Solution: RoboDebt Advisor & What it Does

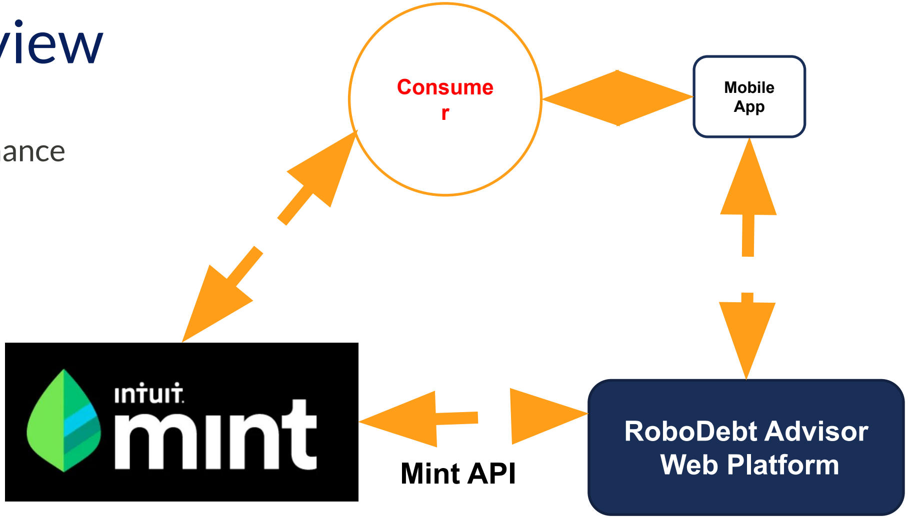
- RoboDebt Advisor Functions:
  - Aggregate a Consumer's financial transaction data from any existing free application api (Mint)
  - Develops an ML Model on the consumer transactions
  - Generates Recommendations on Debt Reduction Strategies and financial mitigation plan
  - Generate a daily report on the consumer's financial health



- RoboDebt Advisor Features:
  - Suggest potential ways to prevent the customer default:
    - By flagging high interest debt by with a set of actions/recommendations
    - Real time monitoring with a workflow engine to trigger alerts and actions
    - Auto-scheduling payments based on pre-determined thresholds.
  - Simple intuitive Interface via Phone App

RoboDebt Advisor: High Level Overview

- Mint is a popular free personal finance application for personal finance
- This Mobile application and web platform leverages Mint's API
- How this solution works:
  - Customer's sign up for Mint and enter their financial data
  - Mint aggregates the Consumer's financial Data
  - Customer downloads RoboDebt Advisor application
  - Logs in to this application with their Mint ID/Password
  - The Phone app creates the account on RoboDebt Web Platform
  - Pulls in customer data from Mint via intuit customer data API *
  - Goes through the following phases

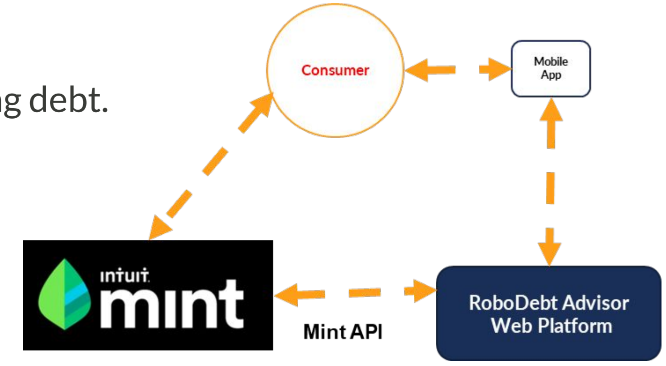


Algorithm Overview

RoboDebt Advisor

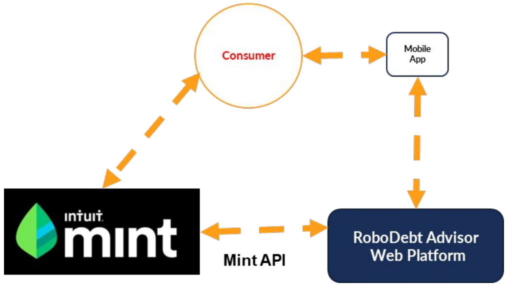
RoboDebt Advisor: Algorithm Overview Data Acquisition Phase

- Aggregate the Bill data from Check Module (Mint Bills) & data from income sources
  - Add up all the bills and income
  - Figure out the difference to arrive at the money left for servicing debt.
  - Rank the obligations into
    - High interest debt
    - Non-deductible low-interest debt
    - Tax-deductible debt
  - Pull in the most recent credit score from Mint's built in VantageScore® interface
  - Record the most up to date information of the consumer's creditworthiness



RoboDebt Advisor: Algorithm Overview Data Analysis Phase 1

- Provide a review day-to-day expenditure. Provide recommendations to cut back.
- Provide a review of subscriptions along with recommendations for cut backs.
- Based on the ranking of the debt provide recommendations for a debt avalanching*
  - high-interest debt (first), suggest doubling up on payments to speed up the payback period
  - non-deductible, low-interest debt (next), again suggestions for doubling up payments
  - tax-deductible debt (last).
  - Provide recommendation to the user for alternatives or encourage quitting the habit of using high-interest debt.

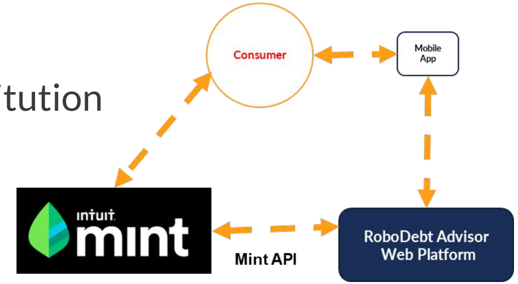


- Based on the credit rating
  - Provide lender recommendations for larger, lower-interest loan, and roadmap to consolidate all the consumer's debts into this loan.

***n.b: Debt Avalanche: Please refer this Investopedia link for details**

RoboDebt Advisor: Algorithm Overview Data Analysis Phase 2

- Recommend a Debt renegotiation plan:
  - Rank the customers lending institutions with whom they have outstanding debts
  - If customer owes more than one bank, start with the bank with which they have the best history.
  - Provide a damage report, a new cash-based budget
  - List out the steps to take to avoid defaulting on the existing debts
  - Suggest renegotiation terms for the debt the customer has with the institution

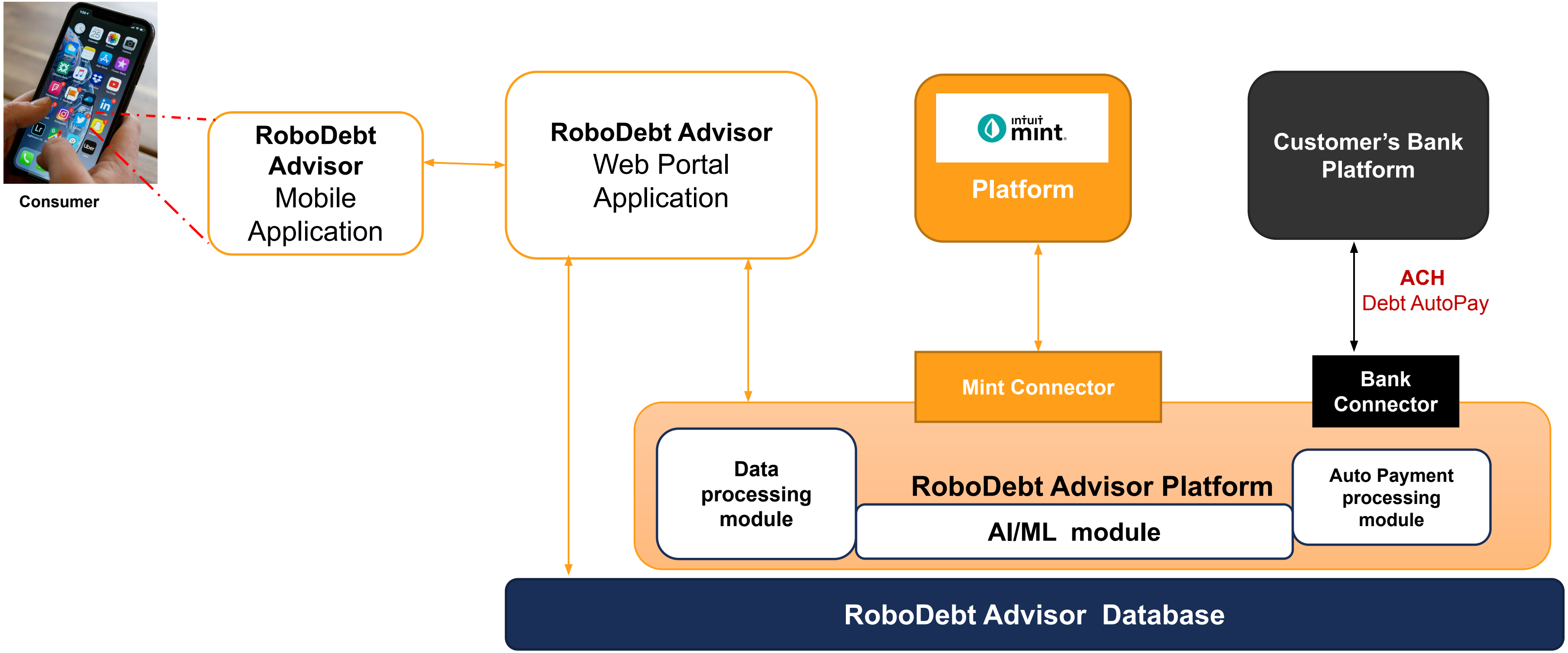


Software Architecture Overview

RoboDebt Advisor

RoboDebt Advisor: Software Platform Building Blocks

- Functional Block Diagram



Automated Clearing House is a payment system that handles online Direct Deposits & Payments across all U.S. bank and credit union accounts.

RoboDebt Advisor: Implementation of Credit Card Default Predictor

- In this challenge round I plan on implementing the Credit Default Predictor
- I used the Credit Card Default Analysis from UC Irvine from Kaggle, for this ML program
  - The variables are:
  - Default Payment (0 = No, 1 = Yes)
  - Amount of Given Credit
  - Gender (1 = Male, 2 = Female)
  - Education (1 = Graduate School, 2 = University, 3 = High School, 4 = Other, 5 = Unknown)
  - Marital Status (1 = Married, 2 = Single, 3 = Others)
  - Age
  - History of Past Payment from April 2005 to September 2005 where -2=no consumption, -1=pay duly, 0=the use of revolving credit, 1=payment delay for one month, 2=payment delay for two months, … 9=payment delay for nine months and above
  - Amount of Bill Statement from April 2005 to September 2005
  - Amount of Previous Payment April 2005 to September 2005

Credit Card Default Predictor

RoboDebt Advisor

RoboDebtAdvisor: Machine Learning for Credit Card Default Analysis

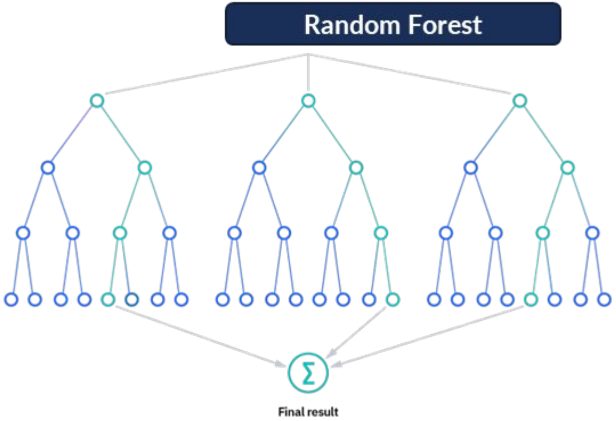
- In this challenge round I am implementing a Credit Default Predictor
- In this challenge round I implemented a Credit Default Predictor Application
- This will be a subapplication in the RoboDebt Advisor application
- I Chose Credit Card debt as it is the most common type of Debt in USA
- Data Source used for Development: Kaggle UCI Credit Card Data
  - https://www.kaggle.com/datasets/uciml/default-of-credit-card-clients-dataset
  - The data contains 24 variables and a total of 30,000 individual instances of customer data
  - The last column indicates credit card default for a customer
  - I modified the:
    - Random Forest code sample from O-Reilly’s Introduction to Machine Learning with Python
    - This is a technology demonstrator as should be treated as such.
    - The development was done on windows running Anaconda Python
    - Pandas and Sklearn library was used

```
C:\Users\gia.nathan\Desktop>python
Python 3.6.3 |Anaconda, Inc.| (default, Nov  8 2017, 15:10:56) [MSC v.1900 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>>
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 30000 entries, 0 to 29999
Data columns (total 24 columns):
 #   Column     Non-Null Count  Dtype
---  ------     --------------  -----
 0   LIMIT_BAL  30000 non-null  float64
 1   SEX        30000 non-null  int64
 2   EDUCATION  30000 non-null  int64
 3   MARRIAGE   30000 non-null  int64
 4   AGE        30000 non-null  int64
 5   PAY_0      30000 non-null  int64
 6   PAY_2      30000 non-null  int64
 7   PAY_3      30000 non-null  int64
 8   PAY_4      30000 non-null  int64
 9   PAY_5      30000 non-null  int64
 10  PAY_6      30000 non-null  int64
 11  BILL_AMT1  30000 non-null  float64
 12  BILL_AMT2  30000 non-null  float64
 13  BILL_AMT3  30000 non-null  float64
 14  BILL_AMT4  30000 non-null  float64
 15  BILL_AMT5  30000 non-null  float64
 16  BILL_AMT6  30000 non-null  float64
 17  PAY_AMT1   30000 non-null  float64
 18  PAY_AMT2   30000 non-null  float64
 19  PAY_AMT3   30000 non-null  float64
 20  PAY_AMT4   30000 non-null  float64
 21  PAY_AMT5   30000 non-null  float64
 22  PAY_AMT6   30000 non-null  float64
 23  Default    30000 non-null  int64
dtypes: float64(13), int64(11)
memory usage: 5.5 MB
```

RoboDebtAdvisor: What the program does

- The program takes in Customer Credit Card information
- The raw data
  - Is in the form of a comma separated value file
  - It is read and cleaned using Pandas Library
  - This is the raw input used by sklearn library to build a model
- We then teach the computer to detect patterns in the data
- This is stored in a file called “Model”
- The Model is trained to detect certain types of patterns.
  - The data is divided into a training and test set to verify the accuracy.
  - The model is trained using algorithms that allows the computer to learn the patterns in data
  - When new data is presented to the data the model is able to detect or ‘predict’ based on what it learned
  - One such algorithm is Decision Tree.
  - A Decision Tree is a graph representation of all possible solutions to a decision based on certain conditions.
  - It can be used for classification tasks (credit card default or not)
  - We can concatenate a number of decision trees using an algorithm can Random Forest. (diagram)
  - This is a supervised algorithm as we are teaching the algorithm with manually cleaned data



Picture Source: IBM Blog on Machine Learning

RoboDebtAdvisor: What the program does

- We can implement the Random Forest algorithm using the sklearn package like so:
  - I divide the data into training and test set
  - The y represents the variable I want to predict
    - 1 represents a default
    - 0 represents normal credit
  - RAN (random forest) is the model
- Sklearn is able to predict with 82% accuracy in few lines of code.
- This is a rudimentary implementation
- The algorithm can definitely be improved

```
from sklearn.ensemble import RandomForestClassifier

Ran = RandomForestClassifier(criterion= 'gini', max_depth= 6,
                                     max_features= 5, n_estimators= 150,
                                     random_state=0)
Ran.fit(X_train, y_train)
y_pred = Ran.predict(X_test)
print('Accuracy:', metrics.accuracy_score(y_pred,y_test))

## 5-fold cross-validation
cv_scores =cross_val_score(Ran, X, y, cv=5)

# Print the 5-fold cross-validation scores
print()
print(classification_report(y_test, y_pred))
print()
print("Average 5-Fold CV Score: {}".format(round(np.mean(cv_scores),4)),
      ", Standard deviation: {}".format(round(np.std(cv_scores),4)))

plt.figure(figsize=(4,3))
ConfMatrix = confusion_matrix(y_test,Ran.predict(X_test))
sns.heatmap(ConfMatrix,annot=True, cmap="Blues", fmt="d",
            xticklabels = ['Non-default', 'Default'],
            yticklabels = ['Non-default', 'Default'])
plt.ylabel('True label')
plt.xlabel('Predicted label')
plt.title("Confusion Matrix - Random Forest");
```

```
Accuracy: 0.8171666666666667
```

RoboDebtAdvisor: What the program does

- Next I implemented:
  - A simple method to use this model for predictions.
  - I then created a pandas dataframe to hold/mimic new customer data
  - This is data the model has never seen before

```
def RF_Prediction(creditCardData):

    crd=pd.DataFrame(creditCardData)

    results = Ran.predict(crd)

    print("The predicted Credit  status is: $", results)
    print("")
    print("Here is how to interpret the credit default status of a customer:")
    print("")
    print("An outcome of  '1' indicates that there is high likely hood of a default on their credit card debt")
    print("")
    print("An outcome of '0' indicates that the customer has high probability of paying their credit card debt" )
    return
```

- We can see that the model
  - is able to predict
  - Credit default 1
    - or
  - No default 0

  Fairly accurately

```
newCustomerCreditCardData=data[29991:]
newCustomerCreditCardData= newCustomerCreditCardData.drop("Default", axis=1)
```

```
RF_Prediction(newCustomerCreditCardData)
```

```
The predicted Credit  status is: $ [1 0 0 1 0 0 1 0 0]

Here is how to interpret the credit default status of a customer:

An outcome of  '1' indicates that there is high likely hood of a default on their credit card debt

An outcome of '0' indicates that the customer has high probability of paying their credit card debt
```

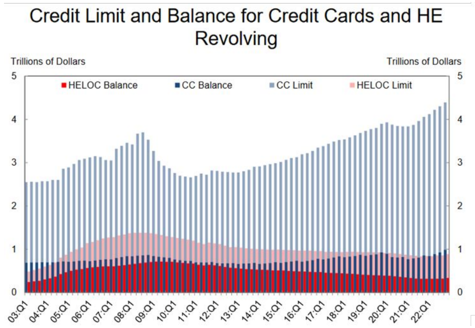
RoboDebtAdvisor: Some Thoughts on preventing Bias

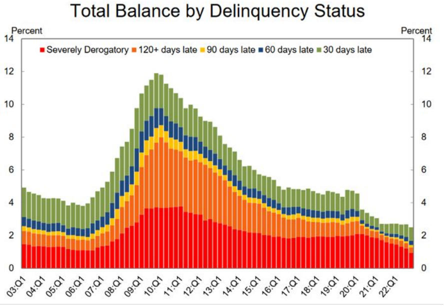
- **ML equivalent of "Blind Taste Test"**
  - We can begin by denying the algorithm the information suspected of biasing the outcome
  - This will ensure that it makes predictions blind to that variable.
  - This is similar to blind taste test used for food products
- **Getting Direct Input From End Users**
  - We can collect a sampling of personal experiences with our AI model by phone or email.
  - We can then focus on finding issues that need correcting based on actual customer feedback
- **Constant Monitoring and Transparency**
  - We can share data used and outcomes with openness, transparency with data providers
  - Solicit feedback from time to time to prevent bias from creeping in
  - Always make sure that we follow Lawful and Ethical practices when collecting data
- **Use of Awareness and Debiasing Tools for Supervised Learning Algorithms**
  - Companies like IBM have developed tools **(AI Fairness 360 (AIF360))**

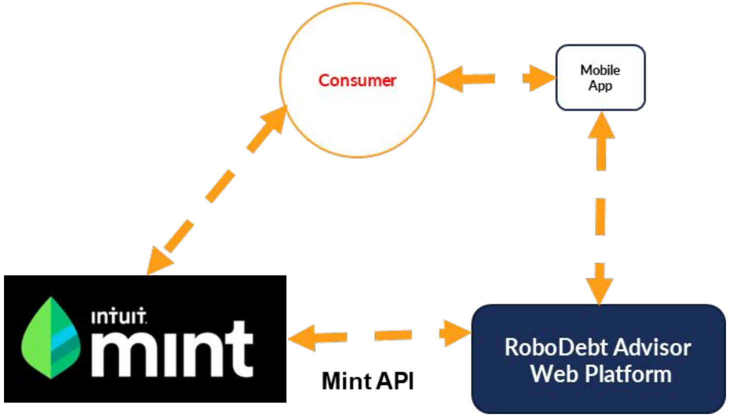
Source: IBM: https://github.com/Trusted-AI/AIF360

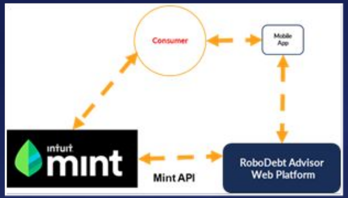
Summary: RoboDebt Advisor: A Fintech Application for Consumer Financial Education

- Consumer Debt is a major problem in the USA
- However most consumers do not have the means to hire a financial advisor
- We can leverage the power of AI/ML to build a fintech application that acts like a virtual financial advisor
- This application can help people recover from a financial debt by promoting financial education









THANK YOU !

gn1188@nhusd.k12.ca.us