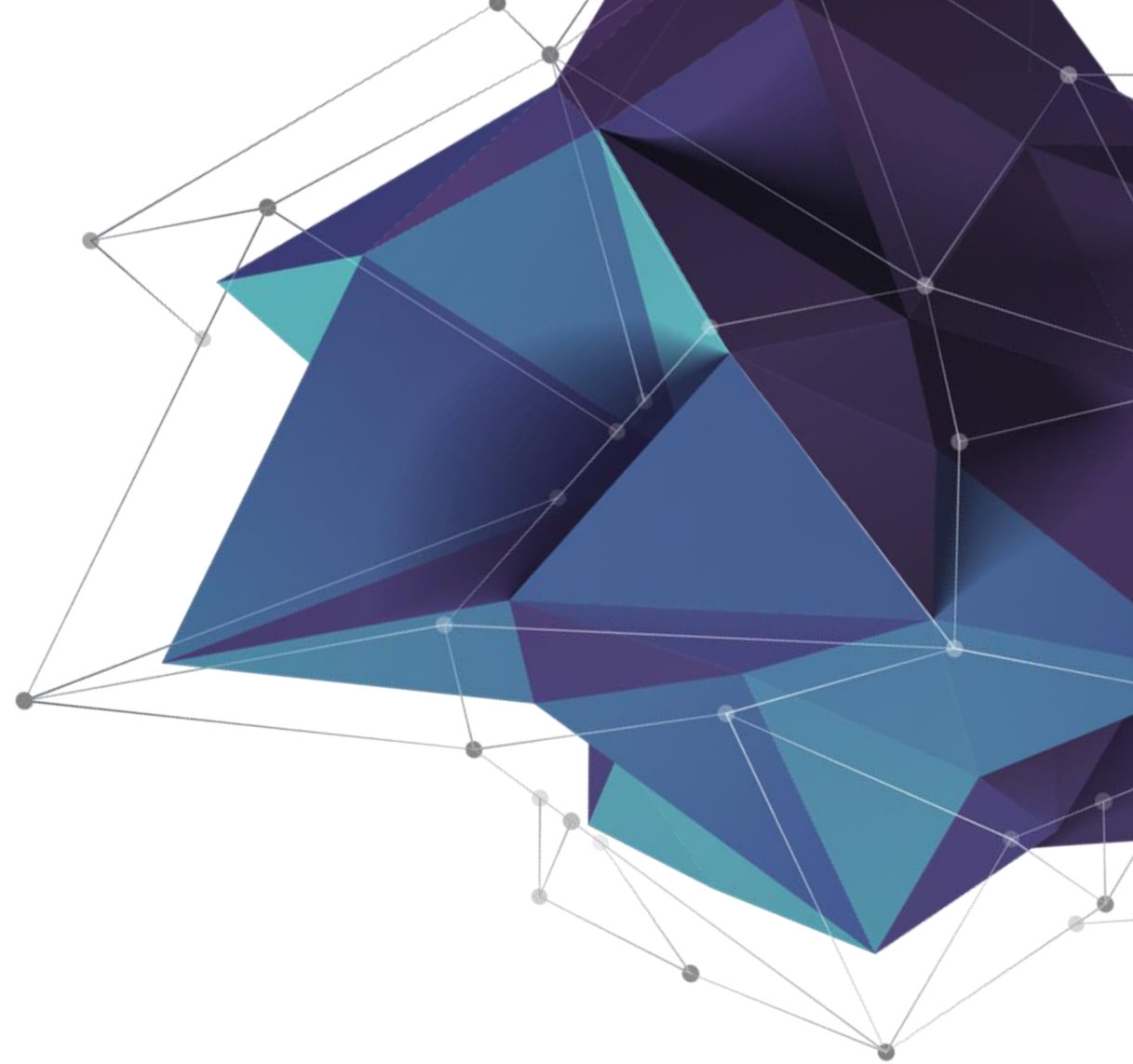


Home Equity Line of Credit

# Risk Prediction

Chen Cao, Xu liu, Shuyu Huang, Yi Huang, Shuyi Chen  
MSMA Group 34





# Roadmap

**01.**Overview

**02.**Data Cleaning

**03.**Model Selection

**04.**Interface





# Overview

HELOC Applications

1

Develop Predictive  
Models To Access Risk

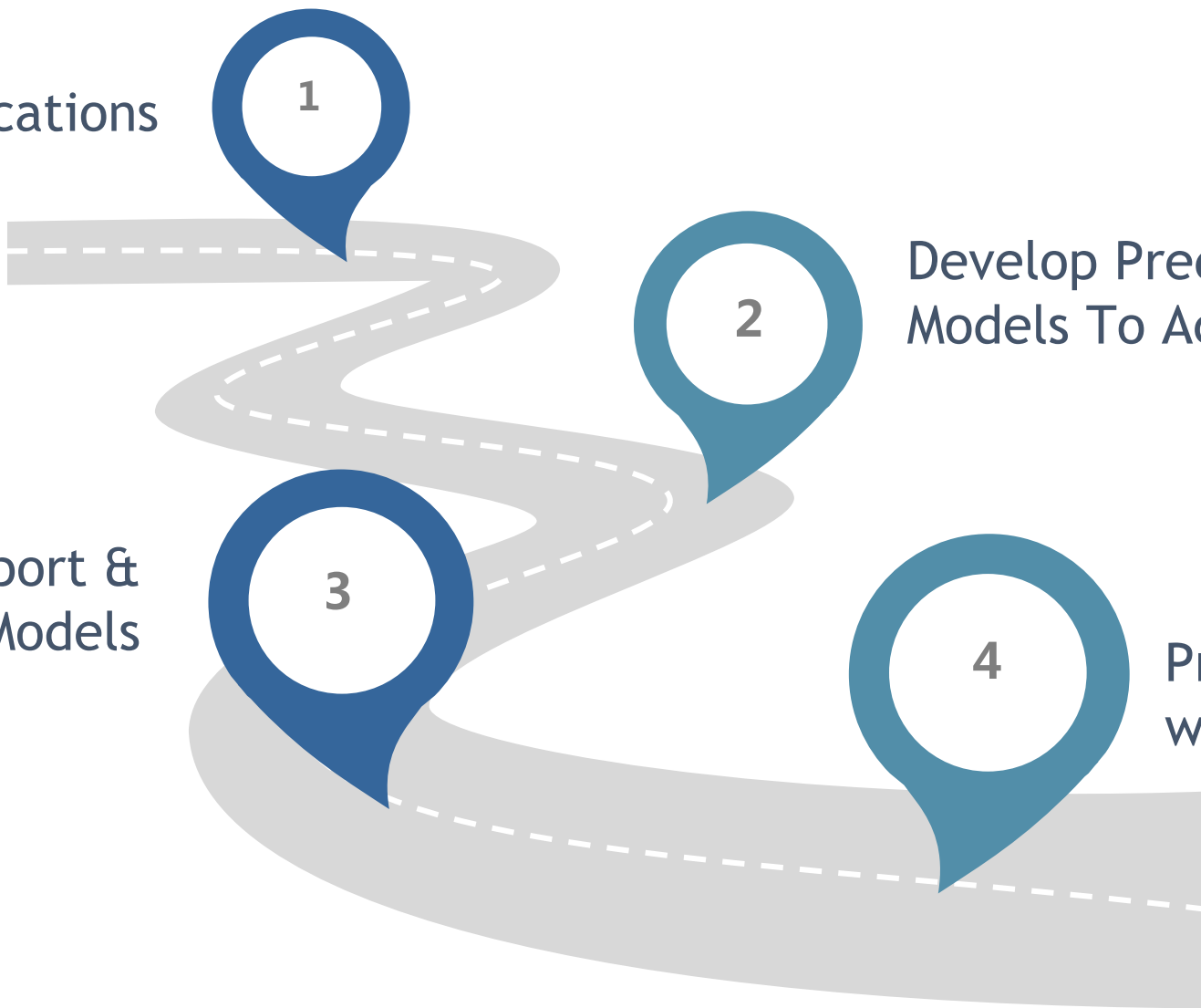
2

Report &  
Compare Models

3

Present the model  
with best performance

4





# Data Cleaning

# 23 Variables

## Consolidation of Synonymous

```
data['MaxDelq2PublicRecLast12M']  
5&6 ; 8&9
```

## Datatype Transfer

```
data['MaxDelqEver'] .astype (object)  
data['MaxDelq2PublicRecLast12M'] .astype (object)  
data['RiskPerformance'] .astype (bool)
```

## Value Imputer

```
data.replace(-7,150)  
data.replace(-8,np.nan)  
Imputer(strategy="mean") or  
Imputer(strategy="median")  
data['ExternalRiskEstimate'].isin([-9])
```

# Output Risk Performance



# Model Selection

## Model Selection - SVC got the highest score

		SVC	LDA-svd	LDA-lsqr	AdaBoost	Logistic Regression	KNN	Decision Tree	QDA
Imputer= mean	CV_Score	70.34%	73.17%	72.42%	73.53%	72.34%	69.80%	70.32%	70.23%
	Test_Score	71.52%	71.87%	71.70%	72.07%	71.75%	69.74%	71.02%	69.31
Imputer= median	CV_Score	72.51%	71.48%	71.50%	72.24%	70.17%	71.41%	70.59%	70.29%
	Test_Score	72.42%	70.94%	70.94%	72.32%	70.83%	70.41%	69.69%	69.98%



# Future Potential- SNN

```
137 class SimpleNN(nn.Module):
138     def __init__(self, in_dim=IN_DIM, hidden_dim_1=HIDDEN_DIM, hidden_dim_2=HIDDEN_DIM):
139         super(SimpleNN, self).__init__()
140         self.fc1 = nn.Linear(IN_DIM, HIDDEN_DIM)
141         self.fc1_bn = nn.BatchNorm1d(HIDDEN_DIM)
142         self.dropout1 = nn.Dropout(p=0.5)
143         self.fc2 = nn.Linear(HIDDEN_DIM, HIDDEN_DIM)
144         self.fc2_bn = nn.BatchNorm1d(HIDDEN_DIM)
145         self.dropout2 = nn.Dropout(p=0.5)
146         self.fc3 = nn.Linear(HIDDEN_DIM, 1)
147         self.act = nn.Sigmoid()
148
149     def forward(self, x):
150         x = F.relu(self.fc1_bn(self.fc1(x)))
151         x = F.relu(self.fc2_bn(self.fc2(self.dropout1(x))))
152         x = self.fc3(self.dropout2(x))
153         return self.act(x)
154
```

## Accuracy:

Epoch 1, train loss: 0.586499, dev acc: 0.727053  
Epoch 2, train loss: 0.577453, dev acc: 0.731401  
Epoch 3, train loss: 0.568702, dev acc: 0.740097  
Epoch 4, train loss: 0.565585, dev acc: 0.738164  
Epoch 5, train loss: 0.564307, dev acc: 0.742995  
Epoch 6, train loss: 0.565393, dev acc: 0.736715  
Epoch 7, train loss: 0.563329, dev acc: 0.739614  
Epoch 8, train loss: 0.564499, dev acc: 0.739130  
Epoch 9, train loss: 0.562446, dev acc: **0.733333**





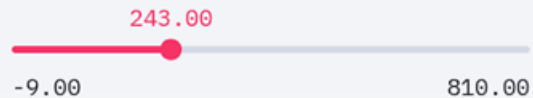
Interface



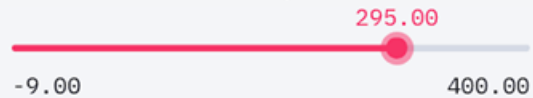
ExternalRiskEstimate



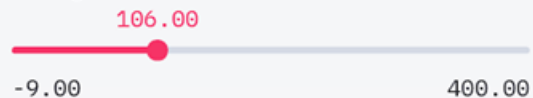
MSinceOldestTradeOpen



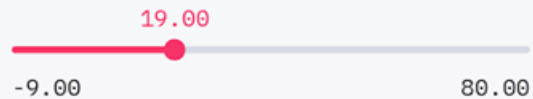
MSinceMostRecentTradeOpen



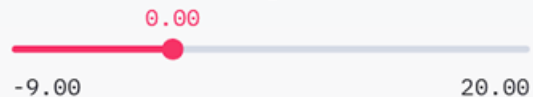
AverageMInFile



NumSatisfactoryTrades



NumTrades60Ever2DerogPubRecs



NumTrades90Ever2DerogPubRec



PercentTradesNeverDelq

95.00

# Credit Risk Report

☐ Show dataframe

Choose a row of information in the dataset (0~10458):

23

Which algorithm?

Support Vector Machine

Prediction: Bad

<http://10.5.63.150:8501>

Risk Prediction

# Thank You !

Chen Cao, Xu liu, Shuyu Huang, Yi Huang, Shuyi Chen  
MSMA Group 34

