Theme 1

1. The explanation for the Data Scientist says that, the NumSatisfactoryTrades (number of satisfactory credit account) applicant with more than 17 satisfactory credit accounts has the ExternalRiskEstimate (higher is better) threshold much less than that of the ones with fewer satisfactory account.

Loan Officer: (Prototypical Explanation)Needs a set of similar user profiles to an applicant in question that a bank employee such as a loan officer may be interested in. This helps the officer understands why the application was rejected or accepted in the first place based on the other user profiles. The explanation for the Data Scientist will not make sense for the Loan Officer.

Customer: (Contrastive Explanation)Needs reason to why they do not qualify for a line of credit and if so what changes in their application would qualify them. On the other hand, if they qualified, they might want to know what factors led to the approval of their application. The explanation for the Data Scientist will not make sense for the Customer as well.

2. Output for id = 3. Explanation: The above table depicts the five closest user profiles to the chosen applicant. Based on the importance weight outputted by the method, we see that the prototype under column zero is the most representative user profile by far. This is (intuitively) confirmed from the feature similarity table above where more than 50% of the features (12 out of 23) of this prototype are very similar to that of the chosen user whose prediction we want to explain. Also, the bank employee looking at the prototypical users and their features surmises that the approved applicant belongs to a group of approved users that have no number of revolving trades with balance (NumInstallTradesWBalance), and NumBank2NatlTradesWHighUtilization. This justification gives the employee more confidence in approving the user's application.

```
vishal — IPython: Users/vishal — ipython — 114×35
   ...: dfw = pd.DataFrame.from_records(np.around(fwt.astype('double'), 2))
   ...: dfw.columns = df.columns[:-1]
   ...: dfw.transpose()
   ...:
Chosen Sample: 3
Prediction made by the model: Good
Prediction probabilities: [[-0.42241824  0.52840394]]

Out[15]:
                                      0     1     2     3     4
ExternalRiskEstimate               0.83  0.69  0.13  0.40  1.00
MSinceOldestTradeOpen              0.38  0.27  0.26  0.18  0.19
MSinceMostRecentTradeOpen          0.61  0.78  0.72  0.10  0.80
AverageMInFile                     0.88  0.48  0.20  0.27  0.42
NumSatisfactoryTrades              0.77  0.18  0.77  0.33  0.39
NumTrades60Ever2DerogPubRec        1.00  0.29  0.29  0.29  0.29
NumTrades90Ever2DerogPubRec        1.00  1.00  0.08  1.00  1.00
PercentTradesNeverDelq             0.78  0.61  0.18  0.29  0.61
MSinceMostRecentDelq               0.87  0.64  0.62  0.11  0.56
MaxDelq2PublicRecLast12M           1.00  1.00  0.21  0.21  1.00
MaxDelqEver                        1.00  0.60  0.22  0.22  0.60
NumTotalTrades                     0.76  0.18  0.84  0.34  0.34
NumTradesOpeninLast12M             1.00  0.13  0.13  1.00  0.13
PercentInstallTrades               0.87  0.12  1.00  0.43  0.59
MSinceMostRecentInqexcl7days       1.00  1.00  1.00  0.38  0.08
NumInqLast6M                       0.00  0.00  0.00  0.00  0.00
NumInqLast6Mexcl7days              0.02  0.13  0.13  0.13  0.02
NetFractionRevolvingBurden         0.13  1.00  1.00  1.00  0.13
NetFractionInstallBurden           0.08  1.00  1.00  1.00  1.00
NumRevolvingTradesWBalance         1.00  1.00  0.13  0.13  1.00
NumInstallTradesWBalance           0.26  0.26  1.00  0.26  1.00
NumBank2NatlTradesWHighUtilization 1.00  1.00  1.00  1.00  1.00
PercentTradesWBalance              0.33  0.26  0.81  0.21  0.45

In [16]:
```

Output for id = 2385. Explanation: The above table depicts the five closest user profiles to the chosen applicant. Based on importance weight outputted by the method, we see that the prototype under column zero is the most representative user profile by far. This is (intuitively) confirmed from the feature similarity table above where more than 50% of the features (14 out of 23) of this prototype are very similar to that of the chosen user whose prediction we want to explain. Also, the bank employee looking at the prototypical users and their features surmises that the approved applicant belongs to a group of approved users with number of trades 60+ and 90+ (NumTrades90Ever2DerogPubRec, and NumTrades60Ever2DerogPubRec). This justification gives the employee more confidence in approving the users application.

```
vishal — IPython: Users/vishal — ipython — 114×35
    ...: dfw = pd.DataFrame.from_records(np.around(fwt.astype('double'), 2))
    ...: dfw.columns = df.columns[:-1]
    ...: dfw.transpose()
    ...:
Chosen Sample: 2385
Prediction made by the model: Good
Prediction probabilities: [[-0.16362362  0.23259817]]

Out[16]:
                                       0     1     2     3     4
ExternalRiskEstimate                0.82  0.14  0.25  0.17  0.67
MSinceOldestTradeOpen               0.65  0.16  0.26  0.50  0.55
MSinceMostRecentTradeOpen           0.45  0.34  0.58  0.58  0.20
AverageMInFile                      0.86  0.05  0.25  0.65  0.24
NumSatisfactoryTrades               0.86  0.13  0.33  0.82  0.52
NumTrades60Ever2DerogPubRec         1.00  1.00  1.00  0.08  1.00
NumTrades90Ever2DerogPubRec         1.00  1.00  1.00  0.08  1.00
PercentTradesNeverDelq              1.00  0.16  1.00  0.11  1.00
MSinceMostRecentDelq                1.00  0.65  1.00  0.08  1.00
MaxDelq2PublicRecLast12M            1.00  0.08  1.00  0.42  1.00
MaxDelqEver                         1.00  0.13  1.00  0.13  1.00
NumTotalTrades                      0.71  0.31  0.24  0.83  0.28
NumTradesOpeninLast12M              1.00  0.38  0.14  0.38  0.38
PercentInstallTrades                0.91  0.12  0.39  0.83  0.49
MSinceMostRecentInqexcl7days        1.00  1.00  0.60  0.08  1.00
NumInqLast6M                        0.32  0.32  0.32  0.18  0.32
NumInqLast6Mexcl7days               0.23  0.48  0.23  0.11  0.23
NetFractionRevolvingBurden          0.62  0.40  0.17  0.40  0.87
NetFractionInstallBurden            0.14  0.66  0.20  0.66  0.66
NumRevolvingTradesWBalance          0.72  0.14  1.00  0.07  0.19
NumInstallTradesWBalance            0.89  0.09  0.62  0.89  0.70
NumBank2NatlTradesWHighUtilization  0.58  0.34  0.20  0.34  0.58
PercentTradesWBalance               0.87  0.40  0.40  0.19  0.46

In [17]:
```

3. *In the python notebook.*

4. Code is in HELOC.ipynb. Attaching screenshot for obtained output:

```
vishal — IPython: Users/vishal — ipython — 80×24
cy: 0.7375
Epoch 495/500
7403/7403 [==============================] - 0s 5us/step - loss: 0.5300 - accura
cy: 0.7375
Epoch 496/500
7403/7403 [==============================] - 0s 5us/step - loss: 0.5300 - accura
cy: 0.7374
Epoch 497/500
7403/7403 [==============================] - 0s 5us/step - loss: 0.5300 - accura
cy: 0.7373
Epoch 498/500
7403/7403 [==============================] - 0s 5us/step - loss: 0.5300 - accura
cy: 0.7374
Epoch 499/500
7403/7403 [==============================] - 0s 5us/step - loss: 0.5300 - accura
cy: 0.7373
Epoch 500/500
7403/7403 [==============================] - 0s 5us/step - loss: 0.5300 - accura
cy: 0.7366
Train accuracy: 0.7394299507141113
Test accuracy: 0.7200161814689636
&lt;IPython.core.display.HTML object&gt;

In [13]:
```

Theme 2

5. The role best suited (as mentioned in the notebook) is that of a social scientist.

	8	132	690	1475	2449	2912	3899	5077	6895	7475
Respondent sequence number	73565.00	73689.00	74247.00	75032.00	76006.00	76469.00	77456.00	78634.00	80452.00	81032.00
Income from wages/salaries	1.00	1.00	2.00	1.00	1.00	1.00	1.00	1.00	1.00	2.00
Income from self employment	2.00	2.00	2.00	2.00	1.00	2.00	2.00	2.00	2.00	2.00
Income from Social Security or RR	2.00	2.00	2.00	2.00	2.00	2.00	1.00	2.00	2.00	1.00
Income from other disability pension	2.00	2.00	2.00	2.00	2.00	1.00	2.00	2.00	2.00	2.00
Income from retirement/survivor pension	2.00	2.00	2.00	2.00	2.00	1.00	2.00	1.00	2.00	2.00
Income from Supplemental Security Income	2.00	2.00	2.00	2.00	2.00	2.00	2.00	2.00	1.00	2.00
Income from state/county cash assistance	2.00	2.00	2.00	2.00	2.00	2.00	2.00	2.00	1.00	2.00
Income from interest/dividends or rental	2.00	1.00	2.00	1.00	2.00	2.00	2.00	2.00	2.00	1.00
Income from other sources	2.00	2.00	2.00	2.00	1.00	2.00	2.00	2.00	2.00	2.00
Monthly family income	12.00	11.00	8.00	4.00	1.00	6.00	7.00	2.00	5.00	3.00
Family monthly poverty level index	5.00	4.30	3.05	1.65	0.00	1.32	2.71	0.44	0.86	1.08
Family monthly poverty level category	3.00	3.00	3.00	2.00	1.00	2.00	3.00	1.00	1.00	1.00
Family has savings more than $5000	NaN	NaN	NaN	2.00	2.00	1.00	NaN	2.00	2.00	2.00
Total savings/cash assets for the family	NaN	NaN	NaN	3.00	1.00	NaN	NaN	1.00	1.00	1.00
Weights of Prototypes	0.18	0.07	0.09	0.06	0.15	0.07	0.12	0.07	0.09	0.10

The social scientist can make the following interpretation by looking at the above table which has prototypes alongwith their weights:

> Looking at the other questions in the questionnaire and the corresponding answers given by the prototypical people above, the social scientist realizes that most people are employed (3rd question) and work for an organization earning through salary/wages (1st two questions). Most of them are also young (5th question) and fit to work (4th question). However, they don't seem to have much savings (last question). The insights that the social scientist acquired from studying the prototypes could also be conveyed to the appropriate government authorities that affect future public policy decisions.

(Source: https://github.com/Trusted-AI/AIX360/blob/master/examples/tutorials/CDC.ipynb)

6. **Section 2 Summary:** The ProtoDash algorithm is used to approximate the distribution of a dataset with weighted samples from another. Samples that minimise Maximum Mean Discrepancy (MMD) are selected. MMD indicates how well the distribution of a dataset is represented by the sample of another distribution. A reformulation of MMD becomes the objective function to be maximised when selecting prototypes and weights. The methodology is as follows:
    a. Compute the value of the objective function given the current prototype set.
    b. Calculate the gradient of each example with respect to the objective function
    c. Choose the example with the largest gradient value and add this to the prototype set
    d. Compute the weights by optimizing the objective function using quadratic programming
    e. Repeat until the appropriate number of prototypes have been selected

Ref:
https://towardsdatascience.com/an-introduction-to-protodash-an-algorithm-to-better-understand-datasets-and-machine-learning-613c24b23719

Section 4 Summary:

ProtoDash	ProtoGreedy
The choice of next element is the one that has the maximum gradient value computed at the current weight vector.	The choice of the next element is the one that maximally increases the set function.
Exactly one function evaluation at each iteration to determine the weight vector.	$O(n^{(2)})$ function evaluation at each iteration to choose the element that maximally increases the set function.
Computationally cheaper compared to ProtoGreedy	Computationally expensive
Theoretical guarantees are weaker.	Stronger theoretical guarantees.