**Original data**

\*\*\*Support Vector Machine\*\*\*

accuracy 0.778222222222

average\_precision\_score 0.228639920666

Kappa 0.0199367901784

Precision Score 0.578947368421

\*\*\*Stochastic Gradient Descent\*\*\*

accuracy 0.775333333333

average\_precision\_score 0.22319616271

Kappa 0.00282691282379

Precision Score 0.292682926829

\*\*\*Nearest Centroid Classifier\*\*\*

accuracy 0.531333333333

average\_precision\_score 0.257792545966

Kappa 0.108053947205

Precision Score 0.274781192754

\*\*\*Decision Tree Classifier\*\*\*

accuracy 0.730333333333

average\_precision\_score 0.298149133136

Kappa 0.237159746286

Precision Score 0.400565504241

\*\*\*Multi-layer Perceptron\*\*\*

accuracy 0.766222222222

average\_precision\_score 0.231931646722

Kappa 0.045871818502

Precision Score 0.362880886427

\*\*\*K-Nearest Neighbor Classifier\*\*\*

accuracy 0.776666666667

average\_precision\_score 0.243180772449

Kappa 0.0764239994976

Precision Score 0.491909385113

**With pre-processed data**

\*\*\*Support Vector Machine\*\*\*

accuracy 0.781555555556

average\_precision\_score 0.237702489807

Kappa 0.0362414396572

Precission Score 0.809523809524

\*\*\*Stochastic Gradient Descent\*\*\*

accuracy 0.776555555556

average\_precision\_score 0.222729010806

Kappa -0.000778549200107

Precission Score 0.125

\*\*\*Nearest Centroid Classifier\*\*\*

accuracy 0.531333333333

average\_precision\_score 0.257792545966

Kappa 0.108053947205

Precision Score 0.274781192754

\*\*\*Decision Tree Classifier\*\*\*

accuracy 1.0

average\_precision\_score 1.0

Kappa 1.0

Precision Score 1.0

\*\*\*Multi-layer Perceptron\*\*\*

accuracy 0.749

average\_precision\_score 0.257281443267

Kappa 0.137198617888

Precision Score 0.385171790235

\*\*\*K-Nearest Neighbor Classifier\*\*\*

accuracy 0.778444444444

average\_precision\_score 0.245493026693

Kappa 0.0809662298875

Precision Score 0.518518518519

\*\*\*Random Forest\*\*\*

accuracy 0.828777777778

average\_precision\_score 0.402643668606

Kappa 0.318823439271

Precision Score 1.0

**ACCURACY SCORES**

SVM Accuracy: 0.78 (+/- 0.00)

KNN Accuracy: 0.78 (+/- 0.00)

RF Accuracy: 0.83 (+/- 0.01)

SDG Accuracy: 0.57 (+/- 0.33)

NCC Accuracy: 0.54 (+/- 0.00)

DT Accuracy: 1.00 (+/- 0.00)

MLP Accuracy: 0.71 (+/- 0.17)

[The CreditOne Project and Writing Your Report](https://ut.daacertificate.com/mc/poa?productID=3650&taskID=3824" \l "collapsepoa7548)

Now you are ready to dive into the CreditOne data and apply what you've already learned to a classification problem; much of the work you have already done can be applied to this last task so know you're not starting completely from the beginning.

One of the main objectives that separates Data Science from Data Analytics is creative problem-solving skills and many times Data Scientists will have to learn new skills just to solve a problem; you'll be doing the same thing here and stretching your existing knowledge by learning things that you have not been introduced to yet to solve the problem at hand.

In addition to applying your new Python and Sci-Kit Learn skills you will also need to demonstrate that you can use the Sci-Kit Learn resources to perform some additional Data Science related skills.

Here is the list of requirements that your data science process should include for your final report:

1. Cleaning and [Pre-processing](http://scikit-learn.org/stable/modules/preprocessing.html#preprocessing)
2. [Covariance Estimation](http://scikit-learn.org/stable/auto_examples/index.html#covariance-estimation)
3. EDA
4. [Feature Engineering](http://scikit-learn.org/stable/auto_examples/index.html#feature-selection) (either PCA or RFE) and [Dimensionality Reduction](http://scikit-learn.org/stable/modules/decomposition.html#decompositions)
5. [One-Hot Encoding](http://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.OneHotEncoder.html)(if needed)
6. [Classification](http://scikit-learn.org/stable/supervised_learning.html#supervised-learning) (Build three model and choose the best)
7. [Model Tuning](http://scikit-learn.org/stable/modules/grid_search.html) (Tune at least two parameters for each model you build)
8. Model Evaluation

You can use the learning resources to assist you with your work.

**This task requires that you prepare one deliverable and one Juypter Notebook:**

1. Customer Default Identification Report that addresses:

Problem:

An increase in customer default rates is bad for Credit One since its business is approving customers for loans in the first place. This is likely to result in the loss of Credit One's business customers.

Questions to Investigate:

1. How do you ensure that customers can/will pay their loans?
2. Can we approve customers with high certainty?

As you progress through the task, begin thinking about how to solve the company's problem.

Here are some lessons the company learned from addressing a similar problem last year:

1. We cannot control customer spending habits
2. We cannot always go from what we find in our analysis to the underlying "why"
3. We must focus on the problems we can solve:
   1. Which attributes in the data can we deem to be statistically significant to the problem at hand?
   2. What concrete information can we derive from the data we have?
   3. What proven methods can we use to uncover more information and why?

Guido is expecting a report in a few days:

1. Your report should be a one to three page Word document that includes rules you believe provide insights, any relevant visualizations, and the answers to the company's questions.
2. It should also include any observations that you've made and any recommendations you might have, supported by evidence uncovered in your analysis.

2. Your Data Science work should be submitted as a Juypyter Notebook and in your GitHub account.