最新技術紹介

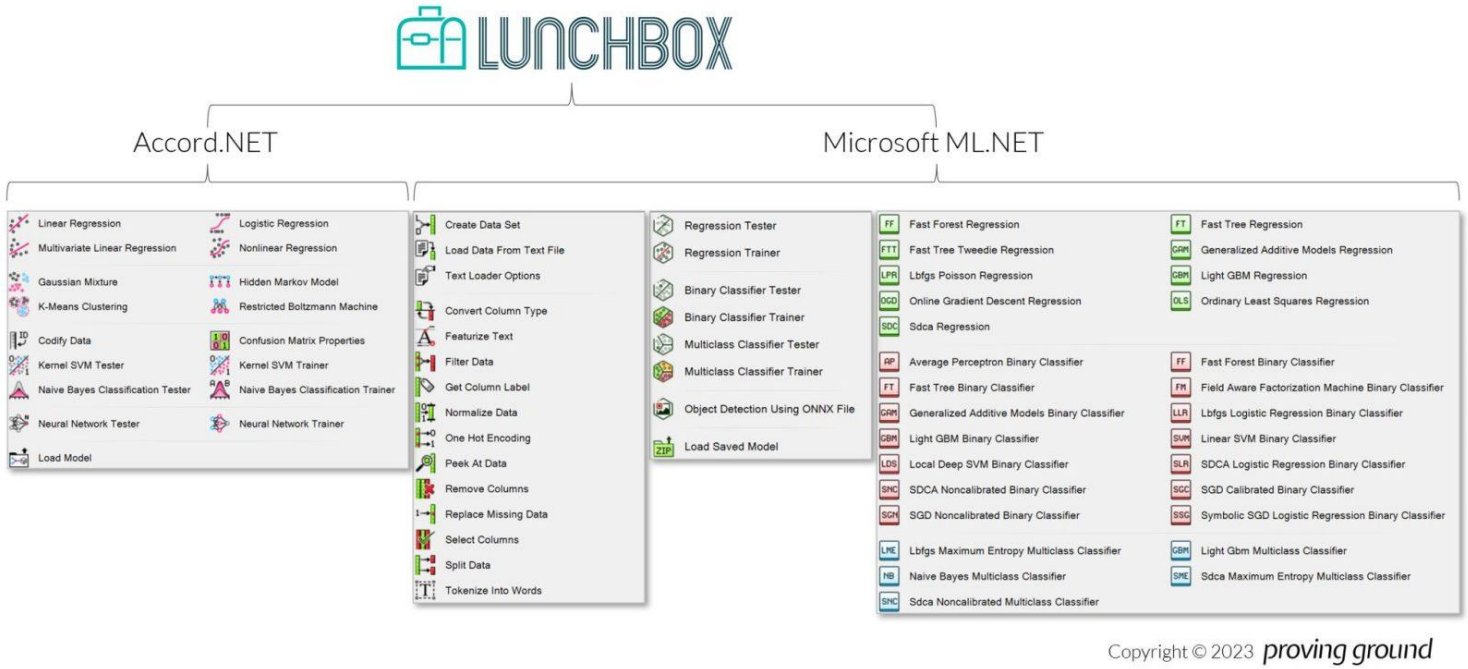
Lunchbox ML コンポーネントの紹介(Accord.NET編)

1 Lunchbox MLの概要

1.1 Lunchbox MLとは

LunchBoxMLは、RhinoGHやRevitDynamo内で汎用機械学習アルゴリズムをトレーニングし、使用するためプラグインです。 LunchBoxはAccord.NETやML.NETと呼ばれるC#の機械学習フレームワークに依存しています。今回は、 GHで使った例を紹介していきます。AIに注目が高まる今、GHから学習データをつくり、設計フローに活用していくアイディアを蓄積することは、大きな課題となってきます。

ご依頼にあったコンポーネントの説明で、まずは Accord.NETの使い方を紹介します。



2 Lunchbox ML コンポーネントの紹介

2.1 Lunchbox Accord.NETの回帰の紹介

Accord.NETのコンポーネントの説明をしていきます。 .NETプラットフォーム向けの機械学習、統計学、信号処理、画像処理などを扱うためのフレームワークです。以下はAccord.NETの主な特徴です。今回は、その中でも回帰系のコンポーネントの紹介をします。

回帰（Regression）とは、データ間の関係を理解し予測するための方法です。例えば、身長と体重の関係を考えてみましょう。一般的に、身長が高い人は体重も多い傾向にあります。ここで、身長から体重を予測することが「回帰」の一例です。

ある情報（例えば、身長）を使って、別の関連する情報（例えば、体重）を予測する方法です。このとき、身長と体重のようなデータの関係性を数学的なモデル（式）で表現します

回帰はデータのパターンを見つけ出し、それを基に新しいデータに対する予測を立てるための強力なツールです。

クラスタリング： データを自然なグループやクラスタに分ける手法です。 **K-平均法**や**ガウス混合モデル** はこのカテゴリーに属します。これらの手法はデータを分割し、類似の特徴を持つデータポイントを同じグループに分類します。

パターン認識： データからパターンを学習し、新しいデータにそれらのパターンを適用する手法です。 **隠れマルコフモデル** や**制限ボルツマンマシン** はこのカテゴリーに分類されます。これらは、データの隠れた構造や時間的な進行を捉えるのに適しています。

線形回帰（Linear Regression)

入力パラメーター

テストデータ（Test, 数値）:

モデルの精度を評価するために使用されるデータです。テストしたい値を入れます。

トレーニング入力（Inputs, 数値）:

トレーニングに使用される入力データのリストです。これらの入力は、線形回帰モデルが入力と出力の関係を学習するための独立変数または特徴量です。

トレーニング出力（Output, 数値）:

トレーニング入力に対応する出力データのリストです。線形回帰では、これはモデルが予測しようとする従属変数です。

シード（Seed, 整数）:

学習アルゴリズムのランダムシードは、再現性を保証するために使用されます。特定のシード値を設定することで、同じデータでモデルをトレーニングする際に、学習アルゴリズム内のランダムプロセス（例えば、重みのランダム初期化）が毎回同じ結果を生み出すことを保証します。

出力パラメーター

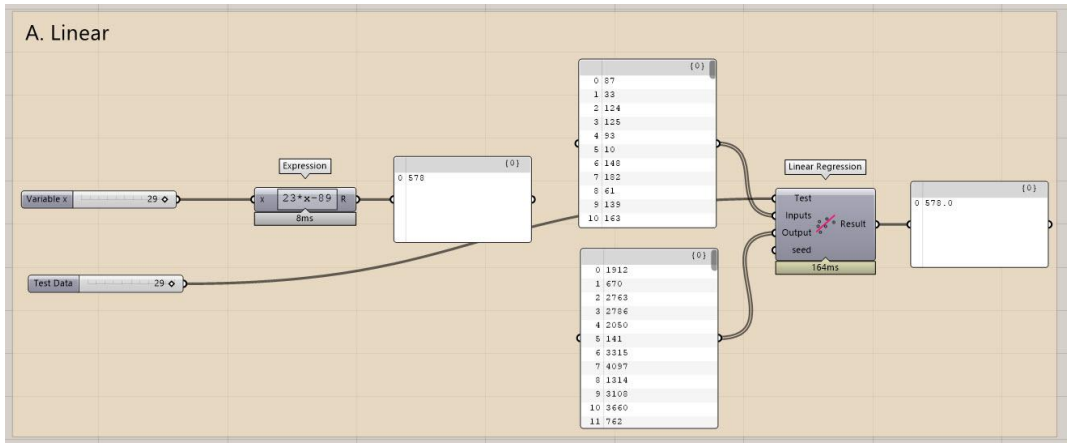
結果（Result, 汎用データ）:

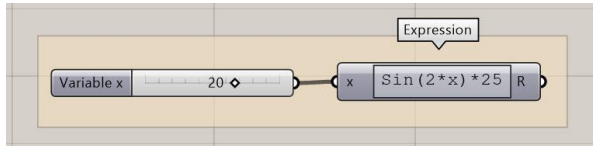
モデルの予測結果です。入力データで線形回帰モデルをトレーニングした後、テストデータを使用して予測を生成します。結果は通常、テストデータ入力に対応するモデルによって予測された値のセットです。

追加の注意点

線形回帰では、モデルは入力特徴量と出力の間の最適な　**直線関係**を見つけることを目指します。例えば「 23*x-29」などの例です。

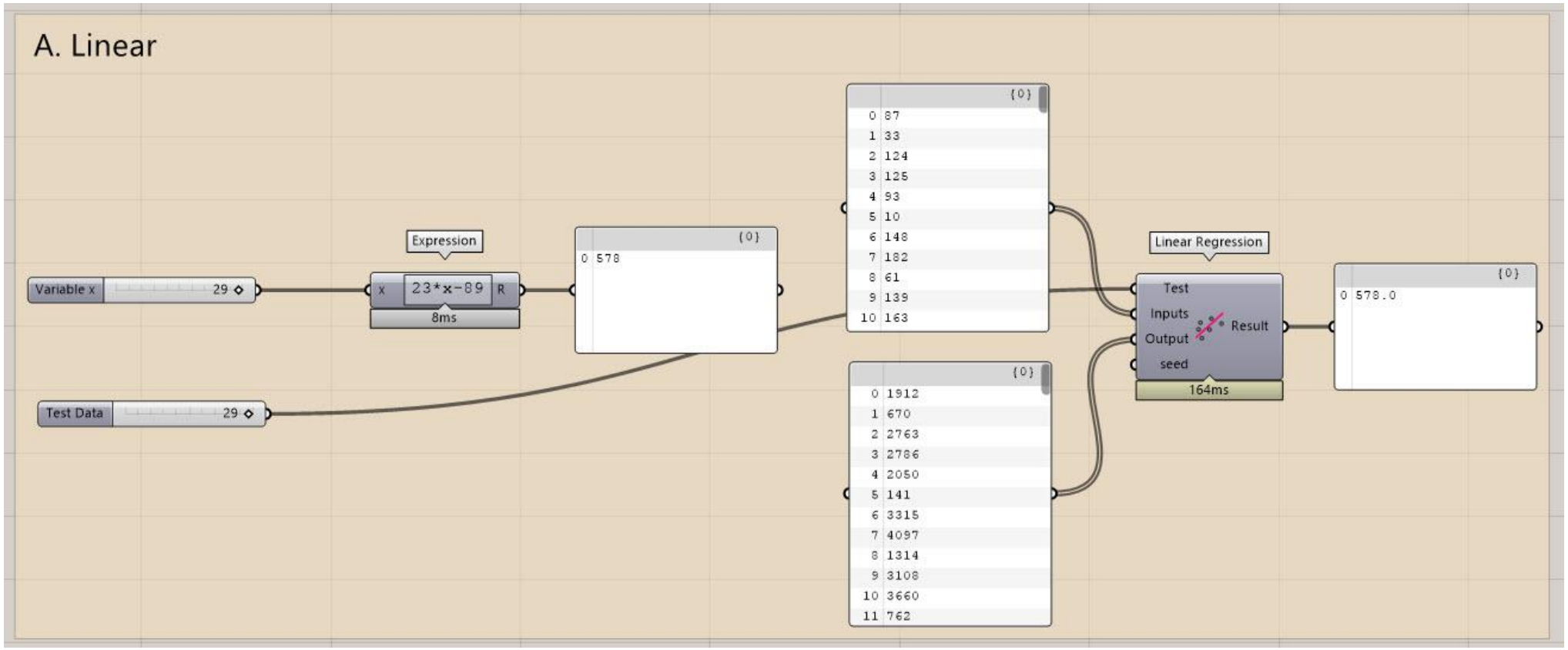
データが直線関係にないようなものを学習させても、思うような結果がでません。例えば、　Sin、Cos、Tanを伴うような関数の結果は出せないので注意が必要です。





GEOMETRY ENGINEERING LAB.

線形回帰（Linear Regression)



この例では、InputsとOutputにあらかじめ関数関係があります。その Inputはこの場合は、23*x-89という式のxの値を入れて、結果を Ouputsに入れています。10個のInputとOutputを入れることで、Linear Regressionのコンポーネントが、23*x-89という式を予測し、結果として、Testに29を入れた際に、23*x-89=578という正しい結果を予測しています。今回は、関係性のあるデータどおしを使ったので、 seedはつかっていませんが、このコンポーネントを使って導きだした式は、いくつかの式を推測する場合があるので、seedを用いてより正しい結果が出やすい式を使うことが可能です。

ロジスティック回帰（Logistic Regression)

入力パラメーター

テストデータ（Test, 数値）:

モデルの精度を評価するために使用されるデータです。このデータは、トレーニングデータとは別のものである必要があります。

トレーニング入力（Inputs, 数値）:

トレーニングに使用される入力データのリストです。これらは、モデルがパターンを学習するための独立変数です。

トレーニング出力（Output, ブーリアン）:

トレーニングデータに対応する出力（目標）のリストです。これは、通常、真または偽（例えば、スパムか非スパムか）のような二値の結果です。

許容誤差（Tol, 数値）:

アルゴリズムが収束したかどうかを判断するために使用される許容誤差の値です。

最大イテレーション数（ MaxIter, 整数）:

アルゴリズムによって実行される最大反復回数です。

正則化値（Regular, 数値）:

目的関数に加えられる正則化の値です。これは、過学習を防ぐために使用されます。

シード（Seed, 整数）:

学習アルゴリズムのランダムシードです。これにより、結果の再現性が保証されます。

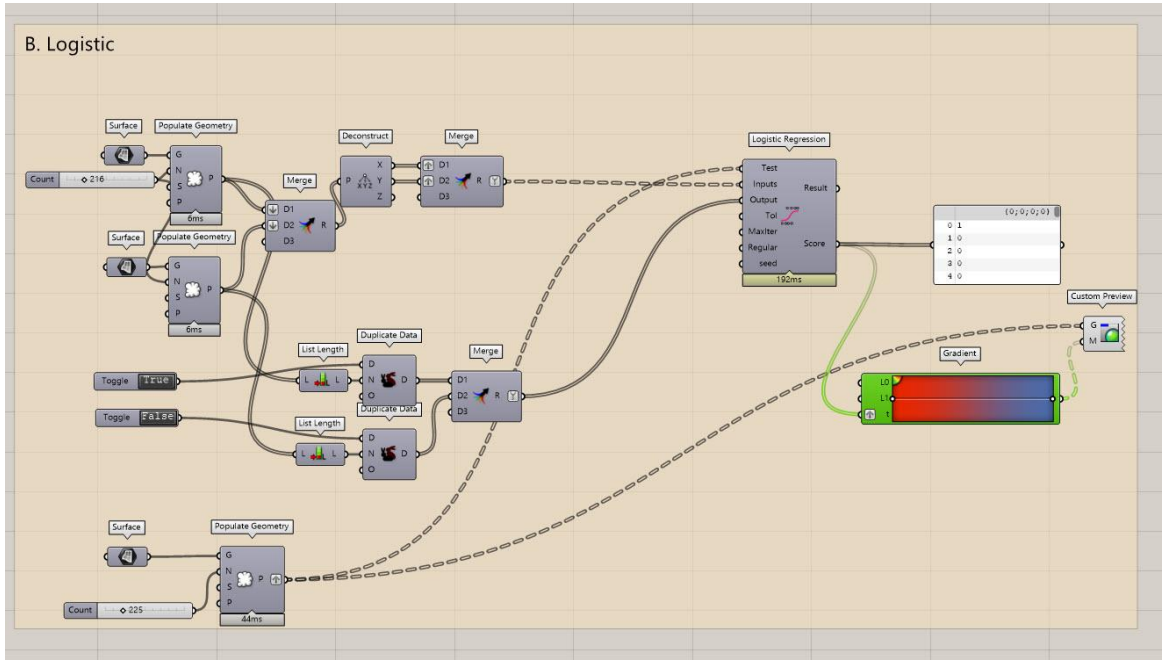
出力パラメーター

結果（Result, 汎用データ）:

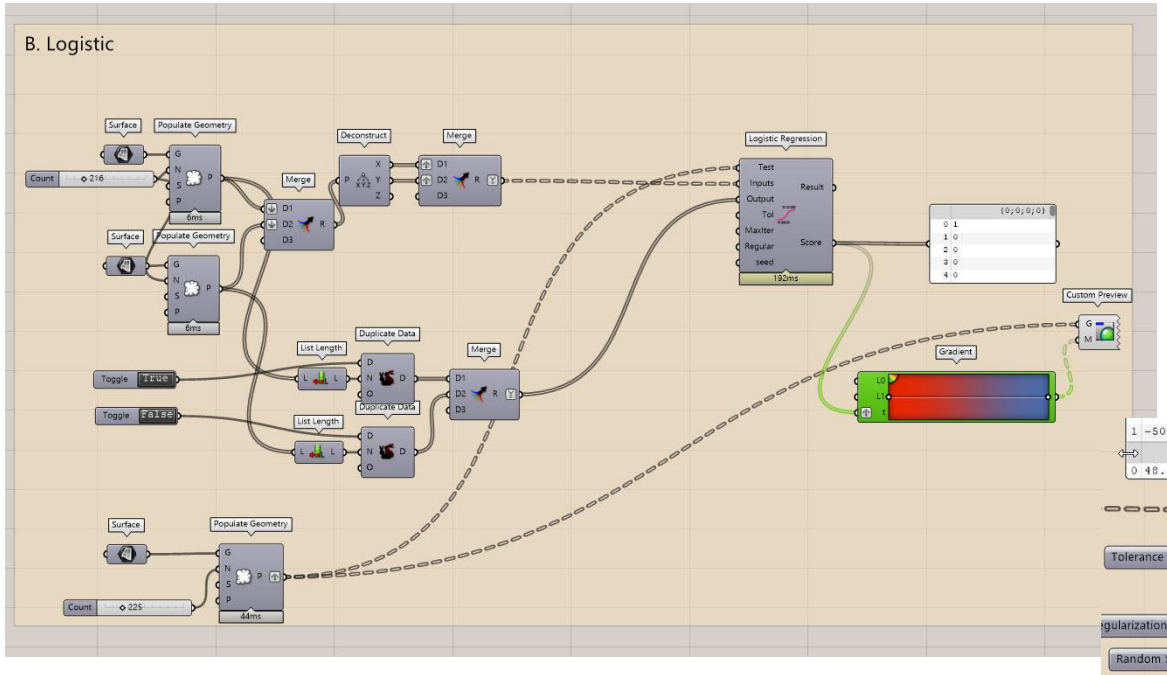
モデルによって生成された予測結果です。これは、テストデータに対する予測されたカテゴリ（例えば、スパムか非スパムか）を表します。

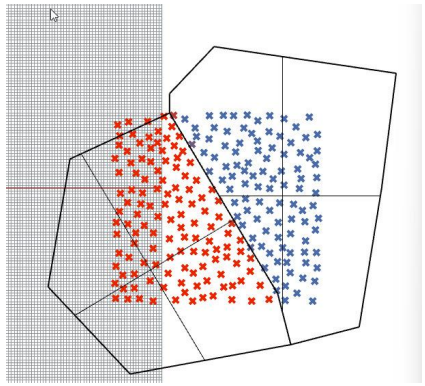
スコア（Score, 汎用データ）:

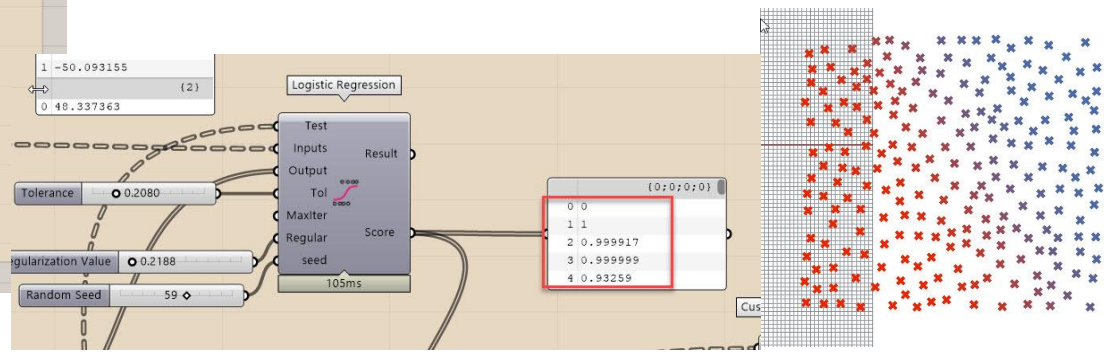
予測されたスコアです。これは、各テストデータポイントが特定のカテゴリに属する確率を表すことが多いです。



ロジスティック回帰（Logistic Regression)







この例では、InputsとOutputにあらかじめ関数関係があります。2つのSurfaceの上に点を充填し、そのサーフェス上の点をTrueとFalseに指定し、サーフェスから外に出ている点の真偽を予測するような例です。ロジスティック回帰は、出力がカテゴリ（はいいいえ、真/偽）である場合に特に有効です。

許容誤差(Tol, 数値)を大きくすると(0.4など)、境目の数値がグラデーショナルに変わります。正則化値(Regular)の適切な設定は、データや問題によって異なります。正則化は、モデルの複雑さを制御し、過学習を防ぐために使用されます。過学習とは、モデルがトレーニングデータに過度に適合し、新しい未知のデータに対してうまく一般化できない状態を指します。数値としては、正則化値を増やしてみたり（例えば1.0や10.0など）、減らしてみたり（例えば、0.001や0.0001など）します。今回は、学習できているので使用しませんでした。学習させてみて、テストをしながら数値を探る工程になります。

非線形回帰（NonLinear Regression)

入力パラメーター

テストデータ（Test, 数値）:

トレーニングデータと比較してテストを行うためのデータです。これはモデルの性能を評価するために使われます。

トレーニング入力（Inputs, 数値）:

モデルのトレーニングに使用される入力データのリストです。これらはモデルがパターンを学習するための基本情報源です。

トレーニング出力（Output, 数値リスト）:

トレーニング入力に対応する出力データのリストです。これはモデルが予測を行う際の目標値となります。

シグマ（Sigma, 整数）:

予測曲線のシグマ（分散）を指定する整数値です。このパラメータはモデルの予測における不確実性の程度を表します。

複雑性（Complex, 数値）:

予測の複雑さを指定する数値です。この値が大きいほど、モデルはより複雑なパターンを捉えることができますが、過学習のリスクも高まります。

シード（Seed, 整数）:

数値生成器のためのシード値です。この値を設定することで、結果の再現性を高めることができます。

出力パラメーター

結果（Result, 汎用データ）:

トレーニングデータに基づいてモデルが生成した予測結果です。

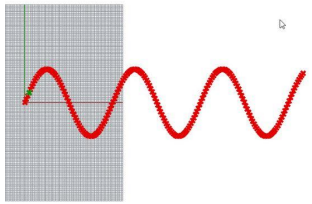
スコア（Score, 汎用データ）:

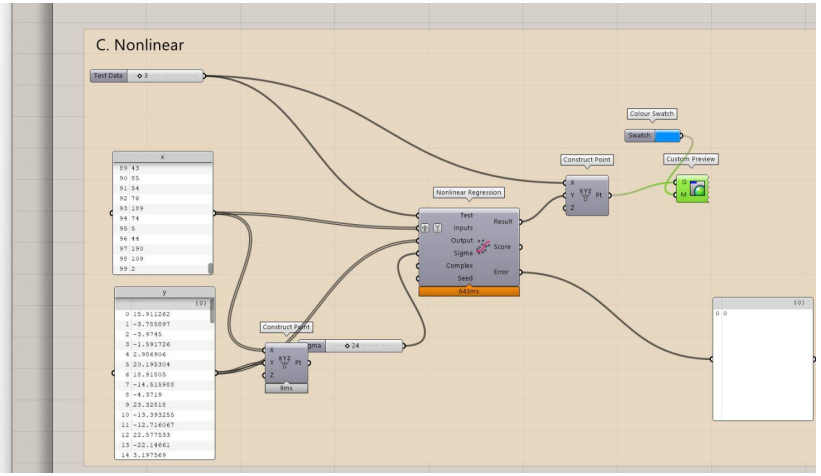
モデルによって予測されたスコアです。これは、各テストデータ点に対する予測値を表します。

エラー（Error, 汎用データ）:

期待される値と予測された値の間の誤差を表します。

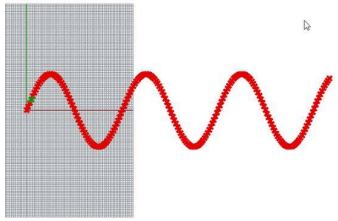
非線形回帰問題を解くためのシーケンシャル・ミニマル・オプティマイゼーション（　SMO）を使用します。

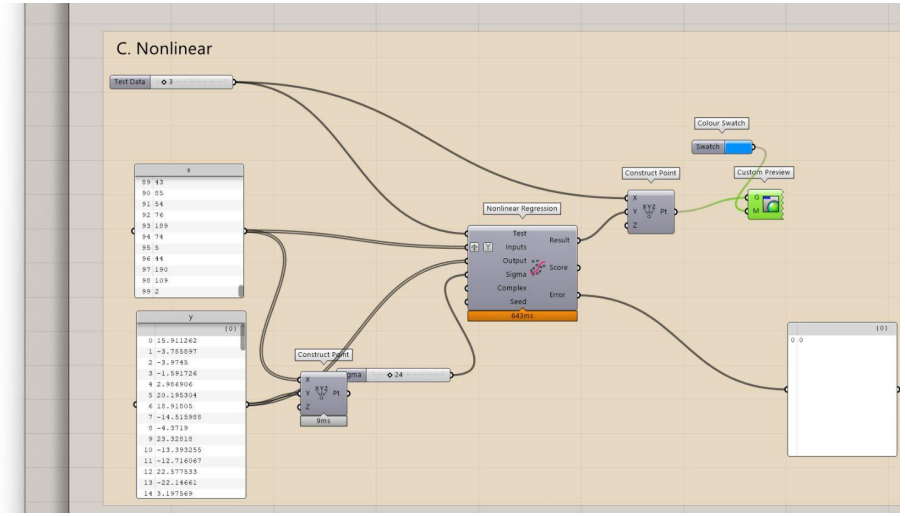


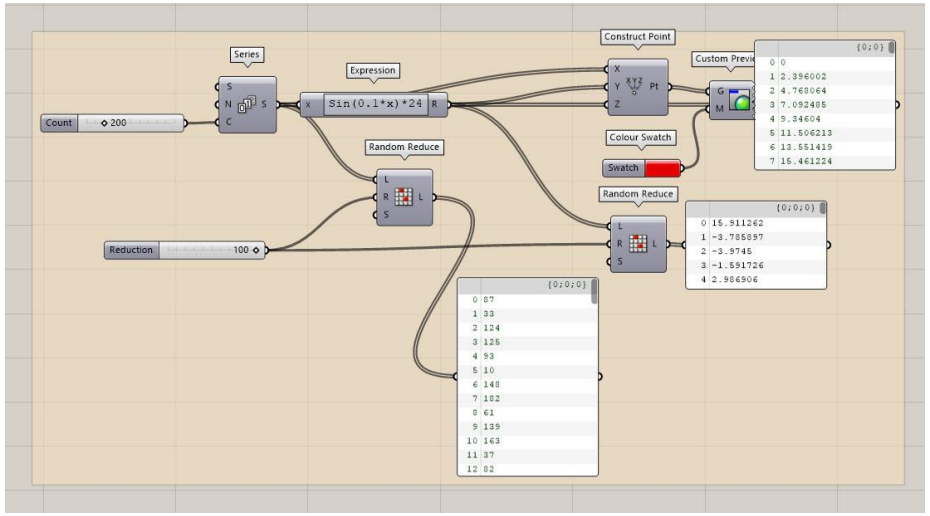


GEOMETRY
ENGINEERING
LAB.

非線形回帰（NonLinear Regression)







この例では、InputsとOutputにあらかじめ関数関係があります。Sin（0.1*x)*24という式のInputと式の結果のOutputをつくります。（右の図）
コンポーネントのInputs、Outputに作った数値を入れて、任意のTestの数値を入れると、結果が正しいことが確認できました。スライダーを動かして、結果を比較します。

シードに関しては、線形の説明時と同じです。「シグマ(Sigma）」と「複雑性（Complex）」の具体的な数値を決めるには、データと問題の性質に応じて様々な要素を考慮する必要があります。

シグマは、予測曲線の分散を制御します。この値が大きいほど、曲線はデータにより柔軟に適応しますが、ノイズにも敏感になります。具体的な数値は、データの分布やノイズのレベルに依存します。通常は、数値実験を通じて最適な値を見つけます。例えば1から始めて、必要に応じて増減させることができます。
テストでの実際の数値の兆候を見て、シグマの値を調整します。

複雑性は、モデルが捉えることのできるパターンの複雑さを表します。高い値は、より複雑なパターンをモデル化できることを意味しますが、過学習のリスクも増加します。
この値もまた、データセットに依存します。簡単なデータセットでは低い値（例えば0.1や1など）から始めることができます。複雑なデータセットでは、より高い値を試す必要があります。

多変量線形回帰（Multivariate Linear Regression)

入力パラメーター

テストデータ（Test, 数値）:

トレーニングデータと比較してテストを行うためのデータです。これはモデルの性能を評価するために使われます。

トレーニング入力（Inputs, 数値）:

モデルのトレーニングに使用される入力データのリストです。これらはモデルがパターンを学習するための基本情報源です。

トレーニング出力（Output, 数値リスト）:

トレーニング入力に対応する出力データのリストです。これはモデルが予測を行う際の目標値となります。

シグマ（Sigma, 整数）:

予測曲線のシグマ（分散）を指定する整数値です。このパラメータはモデルの予測における不確実性の程度を表します。

複雑性（Complex, 数値）:

予測の複雑さを指定する数値です。この値が大きいほど、モデルはより複雑なパターンを捉えることができますが、過学習のリスクも高まります。

シード（Seed, 整数）:

数値生成器のためのシード値です。この値を設定することで、結果の再現性を高めることができます。

出力パラメーター

結果（Result, 汎用データ）:

トレーニングデータに基づいてモデルが生成した予測結果です。

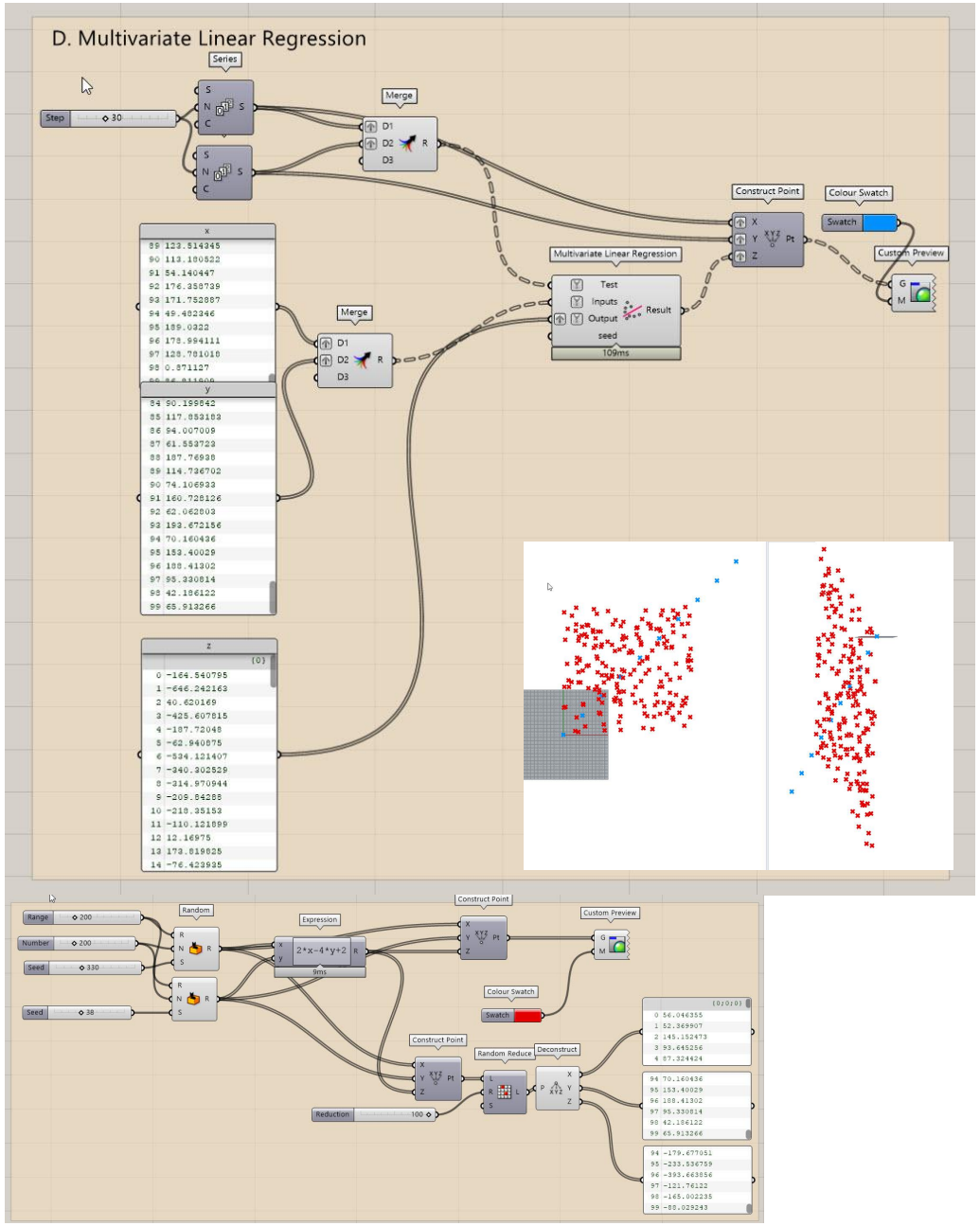
スコア（Score, 汎用データ）:

モデルによって予測されたスコアです。これは、各テストデータ点に対する予測値を表します。

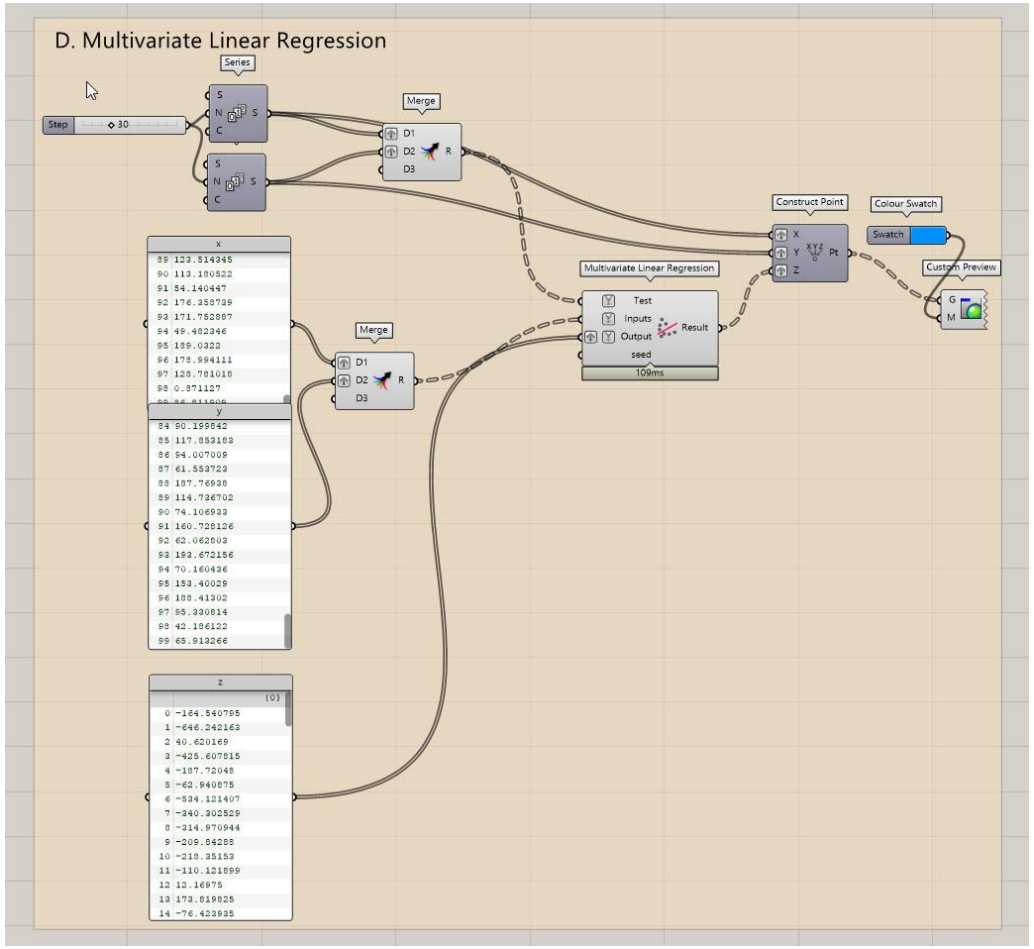
エラー（Error, 汎用データ）:

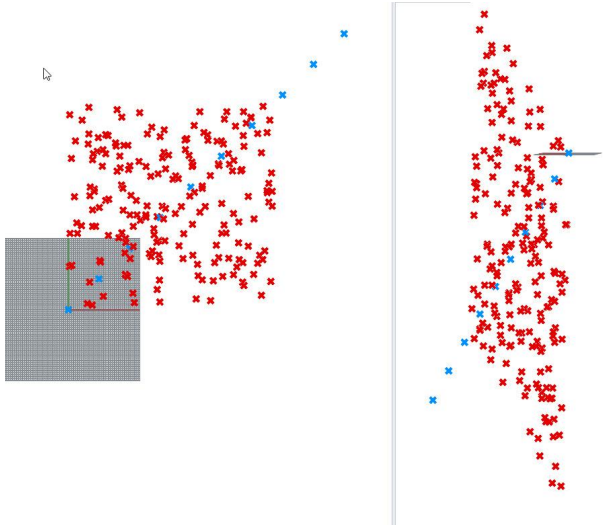
期待される値と予測された値の間の誤差を表します。

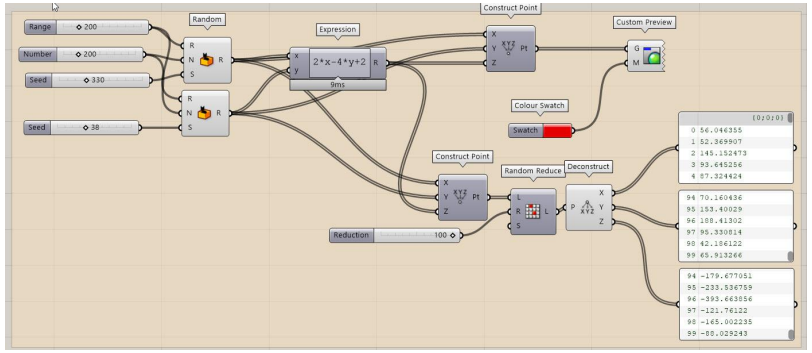
非線形回帰問題を解くためのシーケンシャル・ミニマル・オプティマイゼーション（　SMO）を使用します。



多変量線形回帰（Multivariate Linear Regression)







この例では、多変量線形回帰は、複数の入力変数（特徴量）を使用して、1つの出力変数を予測する手法です。　InputsとOutputにあらかじめ関数関係があります。その Inputsは
この場合は、2*x-4*y+2という式のx、yの値を入れて、結果を Ouputに入れています。100個のInputとOutputを入れることで、NonLinear Regressionのコンポーネントが、
2*x-4*y+2という式を予測し、結果として、Testに2つの数値を入れた際に、2*x-4*y+2という正しい結果を予測しています。今回は、関係性のあるデータどおしを使ったので、
seedはつかっていませんが、このコンポーネントを使って導きだした式は、いくつかの式を推測する場合があるので、　seedを用いてより正しい結果が出やすい式を使うことが可能です。

K-平均法 (K-Means Clustering)

トレーニング入力（Inputs, 数値）:

これはモデルのトレーニングに使用されるデータのリストです。各データポイントは、クラスタリングまたはモデルによって分析されます。

クラスタの数（Clusters, 整数）:

データを分割するクラスタの数を指定します。この数は、分析したいデータ内の異なるグループまたはセグメントの数に対応します。

シード値（Seed, 整数）:

Accordアルゴリズムや他のランダムプロセスにおいて、結果の再現性を確保するためのシード値です。

出力パラメーター

結果（Result, 汎用データ）:

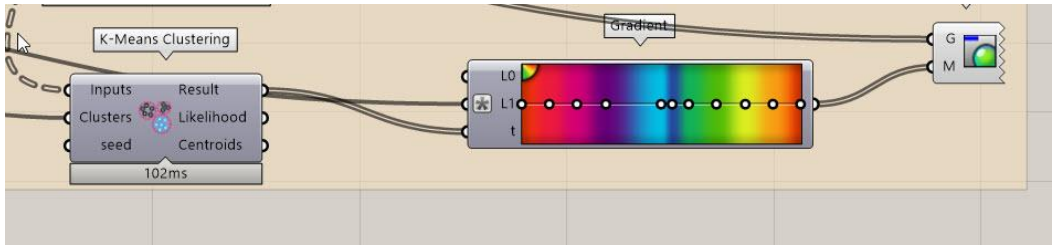
クラスタリングアルゴリズムによる予測結果です。通常、各データポイントがどのクラスタに属するかを示します。

尤度（Likelihood, 汎用データ）:

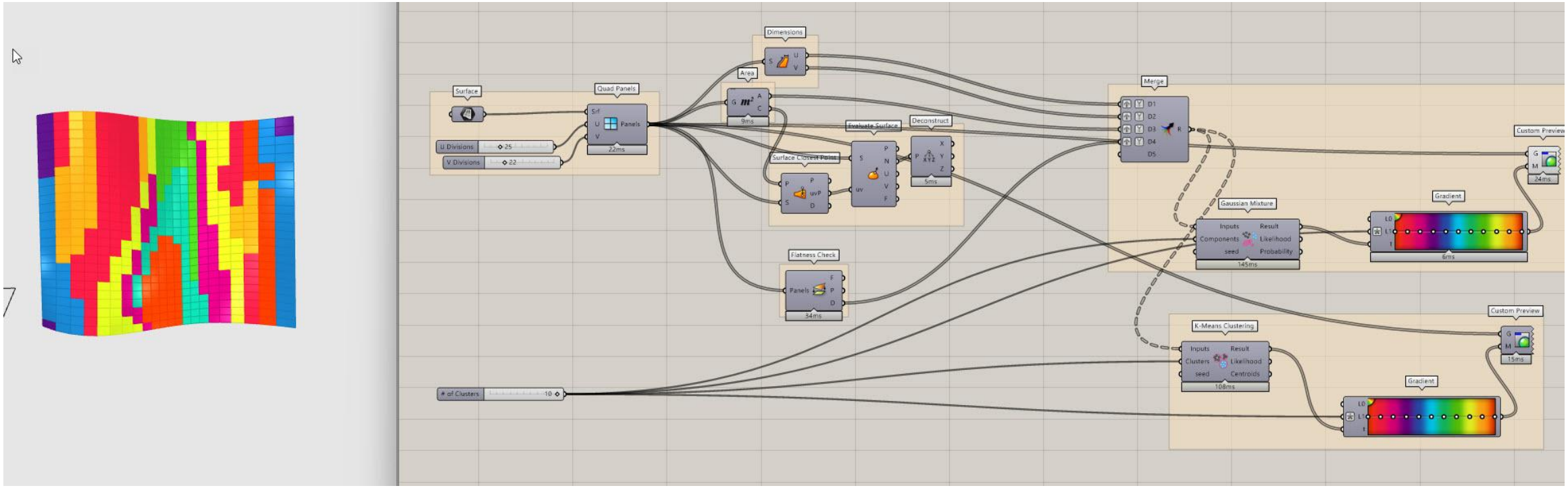
各入力が特定のクラスタに属するログ尤度です。これは、入力が特定のクラスタに属する確率の尺度です。

セントロイド（Centroids, 汎用データ）:

各クラスタの中心点（セントロイド）です。セントロイドは、クラスタ内の全データポイントの平均点として定義されます。



K-平均法 (K-Means Clustering)



ガウス混合モデル (Gaussian Mixture Model, GMM):

入力パラメーター

トレーニング入力（Inputs, 数値）:

これはモデルのトレーニングに使用されるデータのリストです。各データポイントは、クラスタリングまたはモデルによって分析されます。

クラスタの数（Components, 整数）:

データを分割するクラスタの数を指定します。この数は、分析したいデータ内の異なるグループまたはセグメントの数に対応します。

シード値（Seed, 整数）:

Accordアルゴリズムや他のランダムプロセスにおいて、結果の再現性を確保するためのシード値です。

出力パラメーター

結果（Result, 汎用データ）:

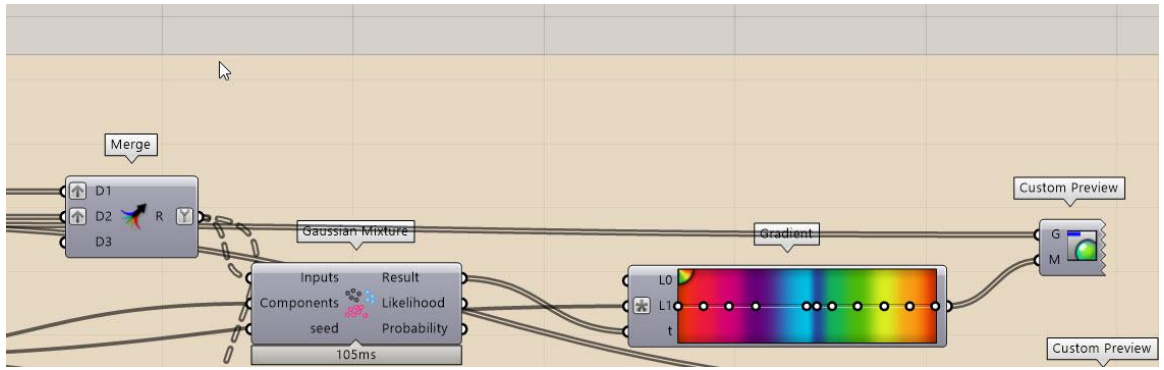
モデルが生成した予測結果です。これは通常、各入力データポイントがどのクラスタに属するかを示します。

尤度（Likelihood, 汎用データ）:

ある入力が特定のクラスタに属するログ尤度です。これは、その入力がそのクラスタに属する可能性の指標です。

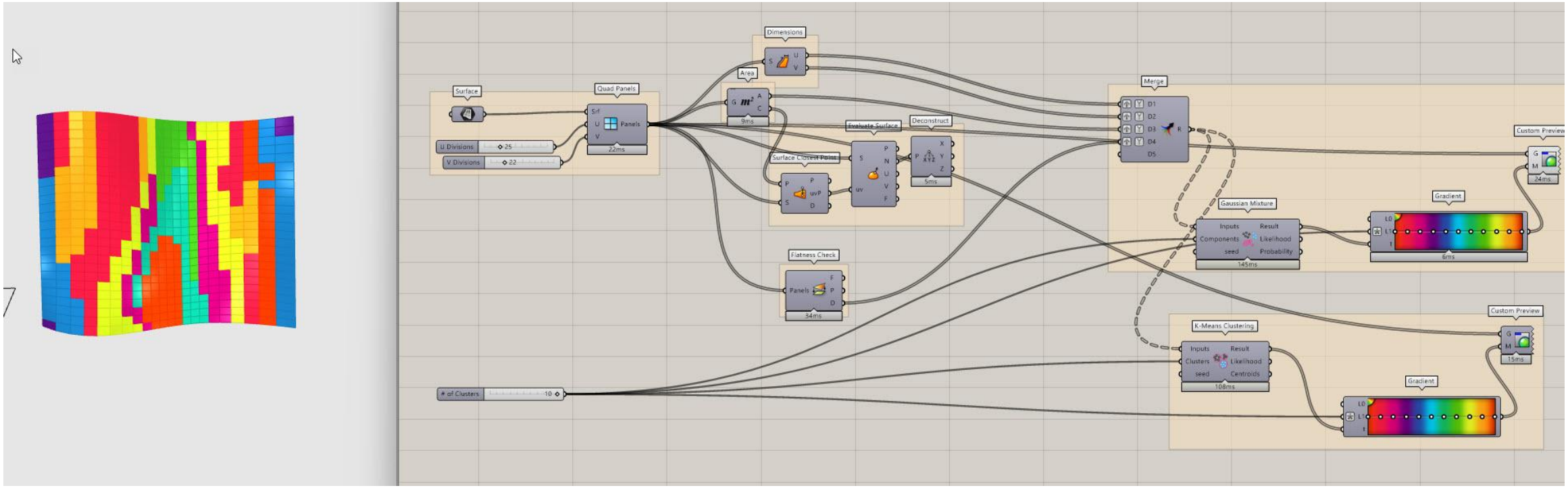
確率（Probability, 汎用データ）:

入力が特定のクラスタに属する確率です。これは、入力が各クラスタに属する確率を示す値です。

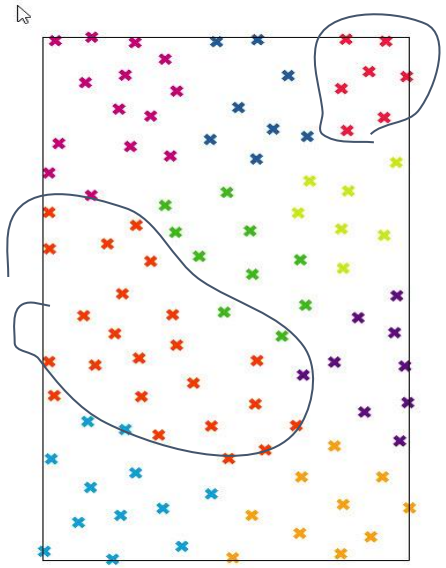


尤度と確率は、データポイントが特定のクラスタに属する可能性を数値化するために使用されます。これにより、データがどの程度明確にクラスタに分類されるかを理解することができます。

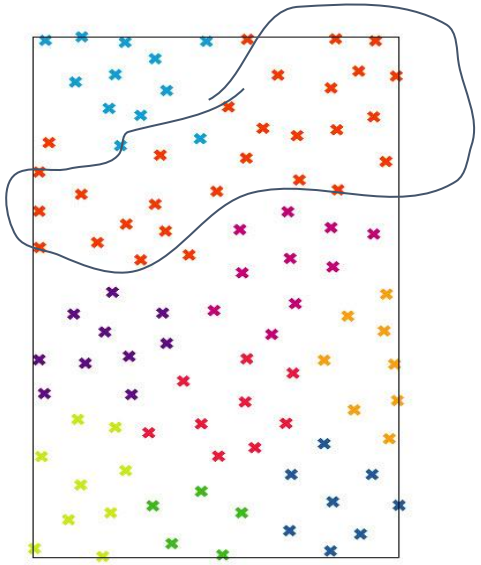
ガウス混合モデル (Gaussian Mixture Model, GMM):



K-平均法 (K-Means Clustering)　vs　ガウス混合モデル (Gaussian Mixture Model, GMM):



丸い感じに分類



複雑に分類

GMMはその柔軟性のために適しています。また、データが重なり合う場合や、各クラスタが異なる形状や方向性を持つ場合にも、GMMはK-平均法よりも優れた選択肢となります。K-平均法は、比較的単純で均一なデータ分布に適しているため、複雑さが少ない場合に選ばれます。

ナイーブベイズ分類

入力パラメーター

トレーニング入力データ(Inputs, 汎用データ) :

これは分類器のトレーニングに使用される入力データです。データは様々な形式で提供される可能性がありますが、通常は特徴量のセットとして表されます。

分類リスト(Classifications, 汎用データ) :

これは各トレーニングデータポイントに対応する分類(ラベル)のリストです。これらの分類は、トレーニングデータを基にして、アルゴリズムが学習するための「答え」を提供します。

出力パラメーター

分類器(Classifier, 汎用データ) :

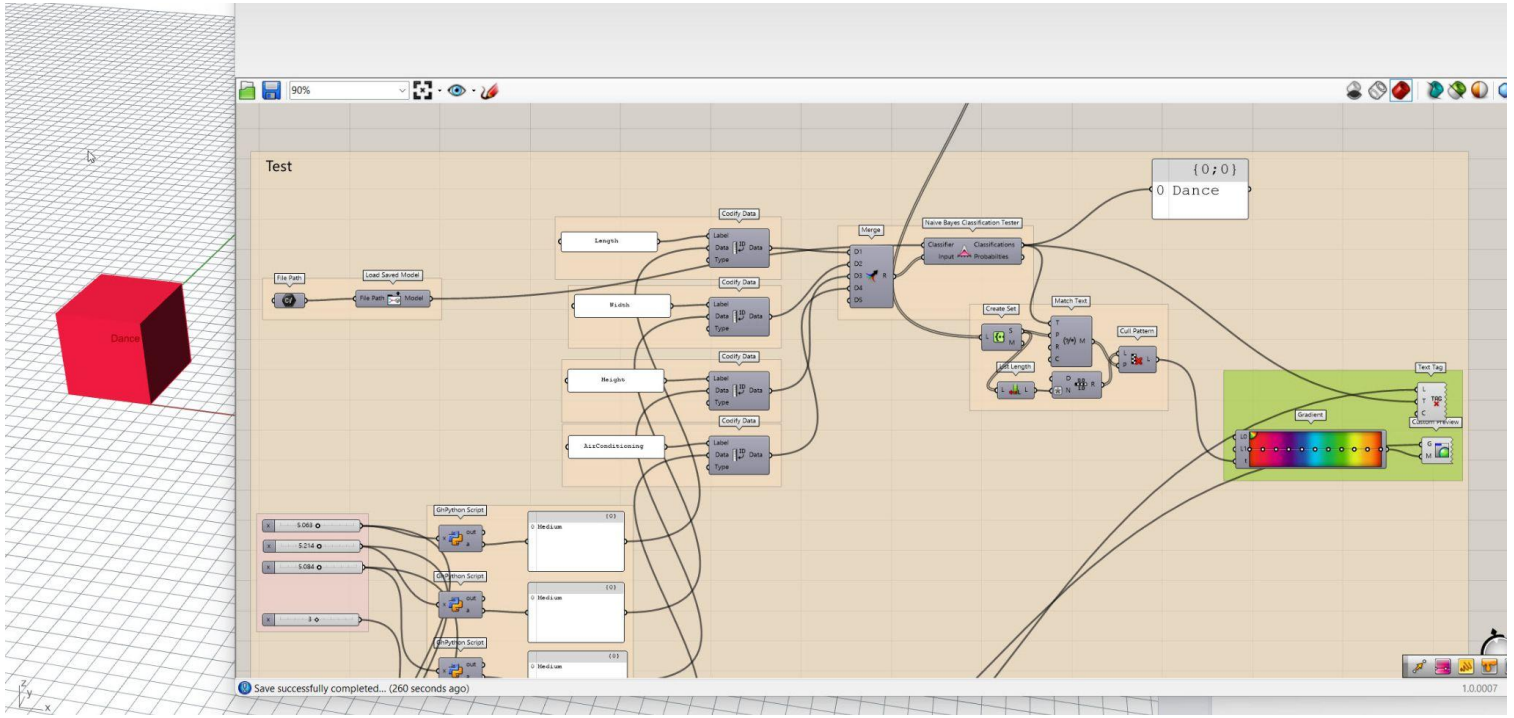
訓練されたナイーブベイズ分類器です。この分類器は、新しいデータポイントが与えられたときに、それがどの分類に属するかを予測するために使用されます。

ナイーブベイズ分類器は、特徴間の独立性を仮定します。これは、一つの特徴が他の特徴に影響を与えないという意味です。
このアルゴリズムは、特にテキスト分類(スパム検出や感情分析など)において効果的です。
単純な計算と強力な性能のため、ナイーブベイズは小さなデータセットにも適しています。

ナイーブベイズ分類

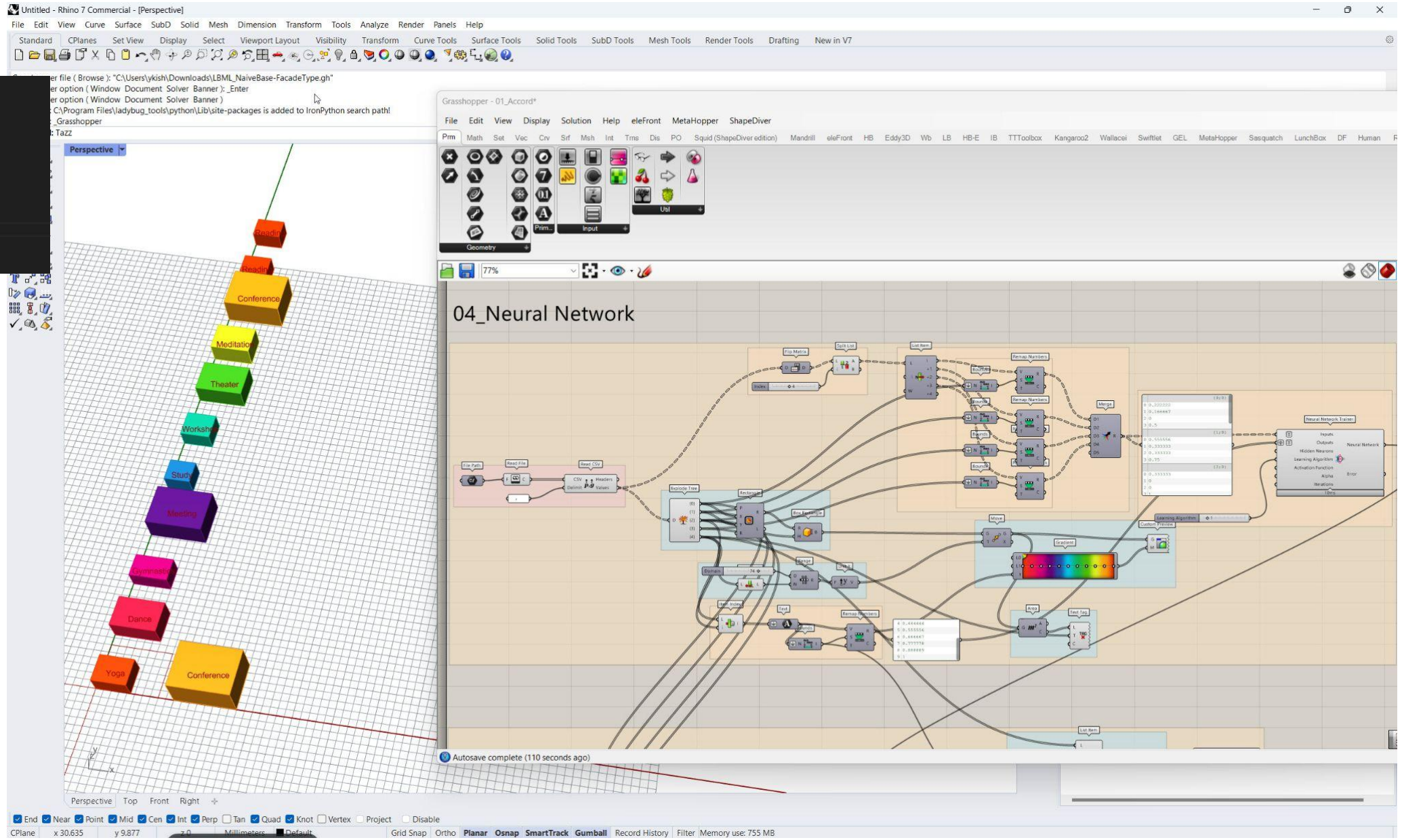
Users > ykish > Documents > GitHub > GEL_GH_Archive > 026_LunchBOXML > Activities >

```
Medium,Small,Small,Medium,Yoga
Small,Very Small,Small,Low,Yoga
Large,Medium,Medium,High,Yoga
Medium,Medium,Medium,High,Dance
Large,Large,Large,Very High,Dance
Small,Medium,Medium,Medium,Dance
Medium,Small,Small,Very High,Gymnastics
Large,Medium,Medium,High,Gymnastics
Small,Small,Small,Low,Gymnastics
Very Large,Large,Large,Low,Meeting
Medium,Medium,Medium,Medium,Meeting
Small,Small,Small,Very Low,Meeting
Small,Small,Small,Very Low,Study
Medium,Medium,Medium,Low,Study
Large,Large,Large,Medium,Study
Small,Small,Small,Medium,Workshop
Medium,Medium,Medium,High,Workshop
Large,Large,Large,Very High,Workshop
Large,Large,Medium,High,Theater
Medium,Medium,Medium,Medium,Theater
Small,Small,Small,Low,Theater
Large,Medium,Medium,Very High,Meditation
Medium,Medium,Medium,High,Meditation
Small,Small,Small,Medium,Meditation
Very Large,Very Large,Large,Low,Conference
Medium,Large,Large,Medium,Conference
Small,Medium,Medium,High,Conference
Small,Small,Small,Very Low,Reading
Medium,Medium,Medium,Low,Reading
Large,Large,Large,High,Reading
```



Neural Network:

```
Length,Width,Height,AirConditioning,Activity
4.5,3.5,2.5,3,Yoga
6.0,4.0,3.0,4,Dance
5.0,3.0,2.5,5,Gymnastics
7.5,5.5,3.5,2,Meeting
3.5,3.0,2.5,1,Study
4.0,3.5,2.5,3,Workshop
6.5,4.5,3.0,4,Theater
5.5,4.0,3.0,5,Meditation
8.0,6.0,4.0,2,Conference
4.0,3.0,2.5,1,Reading
```



GEOMETRY ENGINEERING LAB.