機械学習のための特徴量最適化

特徴量選択の基本

特徴量選択を実施することで、機械学習モデルを大幅に改善できます。ここでは、特徴量選択について知っておくべきことをすべて概説します。なぜ特徴量選択が重要なのか、そして特徴量選択がなぜ難しい問題なのかを説明します。また、特徴量選択を行うために使用されているさまざまなアプローチについて実例をもとに詳しく解説します。

なぜ特徴量選択が重要なのか？

特徴量設計は、モデルタイプ（決定木や回帰などアルゴリズム）やそのパラメータよりもモデルの品質に大きな影響を与えるという共通認識があります。しかしながら、カーネル関数を用いたサポートベクタマシンや隠れ層を用い暗黙的な特徴空間を生成するディープラーニングなどが使えるようになった時代においても特徴量選択は依然として重要な問題なのでしょうか？

特徴量選択が重要な問題である理由として、下記の 2 つが挙げられます。

1 つ目の理由として、実際のパターンを隠してしまうほどの余分な特徴量で学習してしまうと、本来のパターンを見つけるのが難しくなります。この場合、モデルは不要な特徴量のノイズパターンを使用し始め、結果としてパフォーマンスの低下につながります。ノイズパターンにフィットしすぎると、新しいデータ・ポイントで予測（推論）が上手くいかず、パフォーマンスがさらに悪化する可能性があります。これは、特に次元数の多いデータセットで起こりやすくなります。特に、決定木などのアルゴリズムは多層ニューラルネットワークと同様にノイズパターンを過度に学習してしまう傾向があります。不要な特徴量を削除することで、モデルが本来のパターンに焦点を合わせ学習することができ、高いパフォーマンスを維持することができます。

2 つ目の理由は、特徴量の数を減らすと、一般的にモデルの学習（トレーニング）が大幅に速くなります。そして大抵の場合、結果として得られるモデルは単純で理解しやすいものとなります。常に単純なモデルになるように心がけ、ノイズを削除し、ロバストなモデル作成を心がける必要があります。

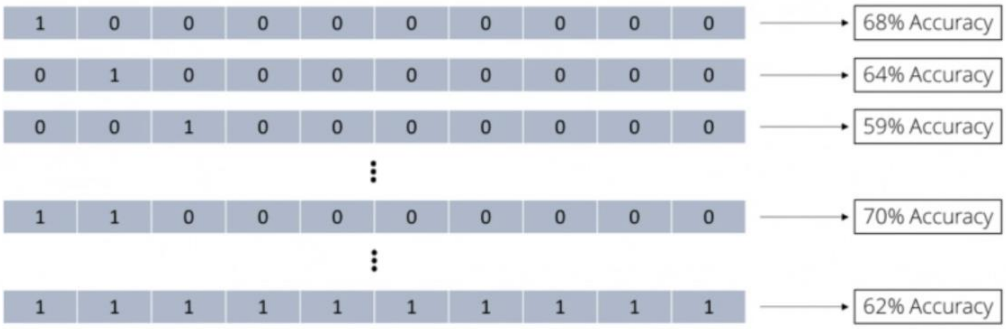
特徴量選択の難しさ

特徴量選択は、ノイズのパターン学習の回避や学習スピード向上の観点からも重要であることが確認で

きました。しかし、特徴量選択はなぜ難しい問題なのでしょうか？例を使って説明してみたいと思います。

10 個の属性(説明変数、列)と 1 個のラベル(目的変数、ターゲット)を持つデータセットがあるとします。ラベル列は予測したい列です。このデータに基づいてモデルを学習（トレーニング）し、データに基づいて構築されたモデルの精度が 62%であることを確認しました。ただ、10 個全ての属性を用いて学習したモデルの方が正答率は高いのでしょうか？それともいくつかの属性を選択した上でモデルを構築した方が正答率は高くなるのでしょうか？後者の場合、どの属性を使えばいいのでしょうか？

どの属性を使用するかについては、10 個（属性）の二進数 0 または 1 のベクトルとして表現することができます。0 は特定の属性が使用されないことを意味し、1 は特定の属性が使用されることを意味します。10 個の属性をすべて使用する場合は、ベクトル(1 1 1 1 1 1 1 1 1 1)となります。特徴量選択は,最も高い正答率（accuracy）を生み出すビットベクトルの探索とも言えます。現実的なアプローチとしては、すべての可能な組み合わせを試すことです。まず、1 つ目の属性のみを使用した場合、68%の正答率、2 つ目の属性のみを使用した場合、64%の正答率、3 つ目の属性のみを使用した場合、59%の正答率というように全ての組み合わせを試し、最も高い正答率となる属性の組み合わせを探索します。



10 個の属性に対して、いくつの組み合わせを試すことはできそうですが、10 の属性全てに対して、使用する、使用しないという組み合わせを決定しようとすると、2x2x2x …=2^{10}=1,024 の異なる結果が存在します。ただ、全て使用しない（0）組み合わせには意味がありませんので、2^{10}−1=1,023 のサブセットを試すことになります。10 属性程度の小さなデータセットでも、多数の変数選択の組み合わせがあることが分かります。また、これらの組み合わせのすべてに対してモデル検証（交差検証など）を実行する必要があります。10 回の交差検証を実施する場合、10,230 モデルをトレーニングする必要があるということになります。

より現実的なデータセットについて考えてみると、データセットが 10 属性だけではなく 100 属性ほどある場合もあるでしょう。その場合は、2^{100}-1 組み合わせが存在し、数の組み合せを計算すると下記の通りとなります。

1,267,650,600,228,229,401,496,703,205,375

相当スペックの良いコンピュータでも、これを実行するのは容易ではありません。ご覧のように属性の数が増えれば増えるほど、試行回数は多くなり、特徴量選択が難しくなるはずです。ここに特徴量選択の難しさがあります。

経験的解決法

すべての組み合わせを調べることは、現実的なアプローチとは言えません（上の例で言えば、2^{100}-1 通り、正答率を求める）。しかし、より正確なモデルを導く可能性の高い組合せのみに焦点を絞ることで、探索スペースを削減し、良いモデルを生成しそうにない属性を無視することもできます。ただ、これ以上最適なソリューションが見つかる保証はありませんし、最適な解も無視してしまう可能性もありますが、これらの経験則は、私たちの力ずくのアプローチ（全ての組み合わせを試行する）よりもはるかに現実的であり、高速です。最終的には良い結果が得られることもあれば、より短時間で最適な結果が得られることもあります。機械学習における経験則的な特徴量選択には、Forward Selection、Backward Selection の 2 つが広く使われています。

Forward Selection

Forward Selection は非常に単純です。まず、1 つの属性のみでモデル作成を試し、最も正答率が高かった属性を保持します。次に 2 つの属性を持つすべての選択可能な組み合わせを試す代わりに、特定の 2 属性を使って試します。前回のラウンドで最も正答率が高かった属性を含む 2 つの属性で試します。改善しなければ、1 つ前の最善の結果、つまり単一属性で停止して、結果を提供します。精度（正答率）が向上した場合は、これまでで一番良かった属性を保持して、さらに 1 つ追加し、正答率が改善しなくなるまでこれを繰り返し続けます。

まずは一属性の 10 個のサブセットから始める。その後、他の属性と 9 通りの組み合わせを試します。改善がない場合は中止するか、より良い正答率が得られた場合は最良の 2 つのサブセットを保持する。このように変数選択を進めていくことで可能性のある 1,023 のサブセットすべてに対して力ずくで進めるのではなく、効率的に変数選択を進めることで計算量を小さくすることができます。

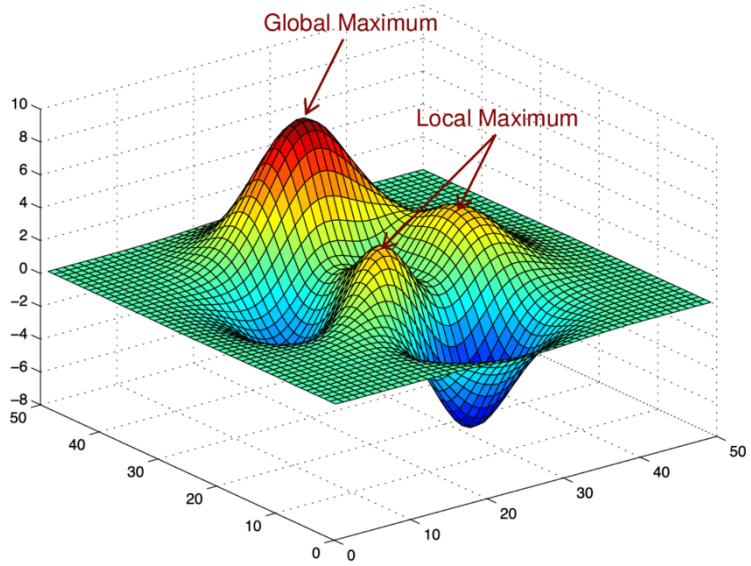
Backward Elimination

Backward Elimination も Forward Selection と考え方は似ており、方向を変えるだけです。まず、すべての属性で構成されるサブセットから始めます。次に、1 つの属性を除外します。1 属性を除外するこ

とにより正答率が改善すれば、さらに属性を除外し、除外可能なすべての組み合わせを調べます。これを改善が見られなくなるまで続けます。

Backward Elimination の手法も総当たりで変数選択を行うよりは、高速に属性を発見してくれます。ある場合には、これらの手法は非常に優れた属性サブセット（データセット）を提供します。

しかし、上記でアウトプットされた結果が最適な属性サブセットになっている可能性は低いと言えます。その理由は、ほとんどのデータセットのモデル精度はいわゆる“multi-modal fitness landscape”を形成するからです。これは、1 つの全体最適値の他に、複数の局所最適値が存在することを意味します。Forward Selection も Backward Elimination の方法も、基準となる属性を決め、そこから正答率（精度）が改善されれば、属性を追加（または削除）し続けます。下の図で言えば、最も近い丘に登り、それ以上登ることができなくなり、そこに閉じ込められてしまうという局所的な最適条件を見つけたに過ぎないのです。これらの方法は、高い丘を探すことさえしません。簡単に手に入るものは何でも取り、正答率（精度）が上がるかを試し続けます。貪欲に改善を進め、改善を止めたとき、最も高い丘の上にたどり着けている可能性は非常に低いと言えるでしょう。言い換えれば、私たちが本当に求めている全体最適値を見逃している可能性ははるかに高いということです。



確かに力ずくで全ての属性に対してアプローチするより Forward Selection や Backward Elimination というアプローチが経験的に有効であり、現実的であることは理解できました。しかし、残念なことに、局所的最適条件を得たに過ぎないということです。

ただ、あきらめる必要はありません。次に、大規模なデータセットでも実行可能な別のアプローチについて説明します。多くの場合、上記アプローチよりもはるかに優れた結果が得られます。次項では、進化アルゴリズムを使用したアプローチを紹介します。

Evolutionary Algorithm

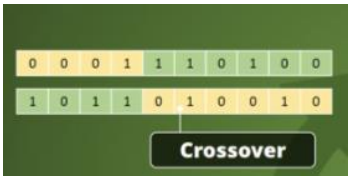
進化的アルゴリズムは、自然進化の考えを模倣する一般的な最適化手法である。3 つの基本的な概念で構成されています。まず、親は子を作るという考え方です（クロスオーバー）。第 2 に、個体が小さな変化（突然変異）を受ける可能性があるということです。第 3 に、生存の可能性は体力のある個体の方が高い（選択される）ということです。

進化アルゴリズムの最初のステップは、個体の集団をつくることから始まります。この個体群は時間とともに進化します。これを進化アルゴリズムの初期化段階と呼びます。出発個体群の個体はランダムに生成され、その結果に応じて、この個人属性を含めるかどうかを決定します。開始母集団のサイズに固定ルールはありませんが、集団中に少なくとも 2 人の個体が存在しなければならず、経験的には、属性の数の 5%から 30%を母集団の大きさとして使用することをお勧めします。

最初の母集団が形成できた後、実行を継続するために必要な手順があります。それは停止基準に達するまでです。クロスオーバーのみを選択する場合もありますし、最初に新世代を選び、その後クロスオーバーを行う人もいます。子供だけを変異させる人もいれば、両親を変異させる人もいる。以下では、一例をステップ毎に分けて説明します。

Step.1

最初のステップは、集団から無作為に両親を選ぶことから始まります。すべての親が 1 回だけクロスオーバーします。選択の段階では、次の世代ではより適合（正答率が高い）した親だけが交配できるようにするという考え方で進めます。これを実装するには多くのバリエーションがあります。例えば、最初の親の前方部分と 2 番目の親の後方部分を組み合わせて子を作成し、進化的アルゴリズムを象徴するジャンプを可能にします。これにより特定地域の極端な現象に巻き込まれる可能性を低減します。



Step.2

次のステップでは、突然変異を起こします。突然変異とは、1 つのビットを 0 から 1 に、あるいはその逆に反転させることを意味します。m が属性の数の場合、1 つの属性の選択を反転する可能性は 1/m です。つまり、突然変異の各個体について平均して 1 つの属性を反転させることになります。元の個体を残すのか、変異した個体だけ残すのかを決めることもできます。

私たちは最初に両親のそれぞれのペアから 2 人の新しい子供が生まれ、個体数が 2 倍になった。そして

それぞれの個体に 1 つの変異を起こしました。次の段階では、属性の数を再び小さい数に縮小しなければなりません。この時もより正答率の高い個体（属性）が選択されることになります。

Step.3

Step3 は、進化アルゴリズムの評価段階です。選択された特徴量の部分集合でトレーニングされた交差検証済みモデルの精度（正答率）を使用します。これらの精度とその際に選択された特徴量を一緒に保存することで、次の手順へと進み新たな選択を実行することができます。

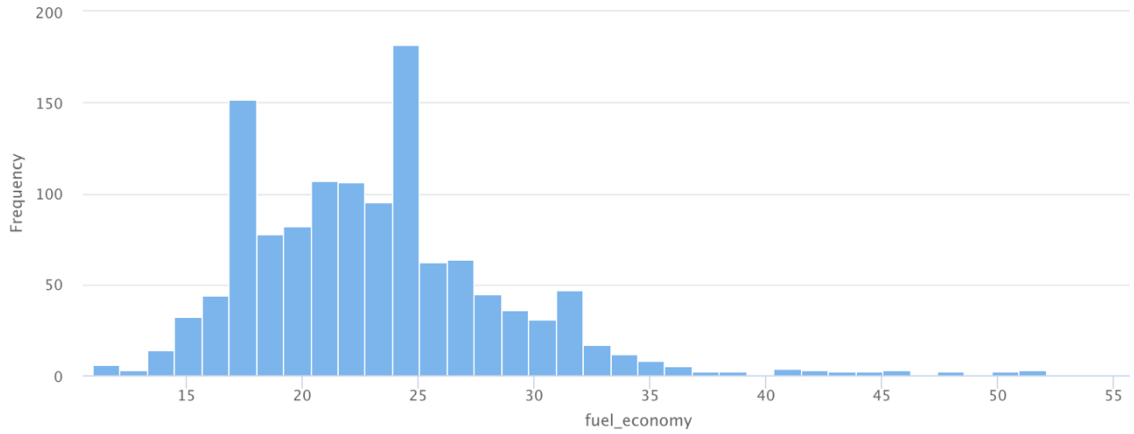
Step.4

最後のステップは選択です。このステップでは、「適者生存」の概念を用います。分かりやすく言えば、より高い精度を出せるモデルを導く属性、すなわち属性部分集合がより高い生存可能性を持つということです。そして、生き残りは、次世代の新しい属性部分集合の基礎を構築します。選択を実行する方法は多数ありますが、ここでは最も広く使用されている選択方式の 1 つであるトーナメント選択を使用します。

トーナメント選択方式では、少人数のグループで対戦し、グループの中で最も適任な人が生き残り、属性部分集合を作り、次の世代に伝えていきます。トーナメントグループを作るために、サンプリングを置き換え、個人を複数回選択することも許容しつつ、次世代の個体群が再び求める精度に達するまで、個体を追加、削除し続けます。

例題：MPG の予測モデル

アメリカ政府が公表している燃費データ（2019）を用いて、MPG を予測するモデルを作成します。燃費は、日本では 1L あたりの走行距離（km）が用いられるのに対し、アメリカでは Mile /(per) ガロンで MPG が用いられます。MPG（fuel economy）をラベル（目的変数）に設定し、分析を進めていくことにします。下記の図は、MPG の頻度分布（ヒストグラム Number of bins = 40）。



使用する特徴量

モデルの説明に使用している特徴は、エンジン排気量（大きさ）、シリンダーの数、トランスミッションタイプ、ギア数、エアインスパイア方法、回生ブレーキタイプ、バッテリー容量 Ah、ドライブトレイン、気筒休止、可変バルブ、燃料タイプです。

上記 11 の属性（特徴量）を用いてモデルを作成するにあたり、様々な関心事が想定されますが、本節では下記の 4 つに焦点を絞り、答えを考えていきます。

（１） エンジン排気量と燃費の間には関係があるのか。

（２） どの特徴量が最も燃費に影響を与えるのか。

（３） 新たな車種の燃費をどのくらいの精度で予測できるのか。

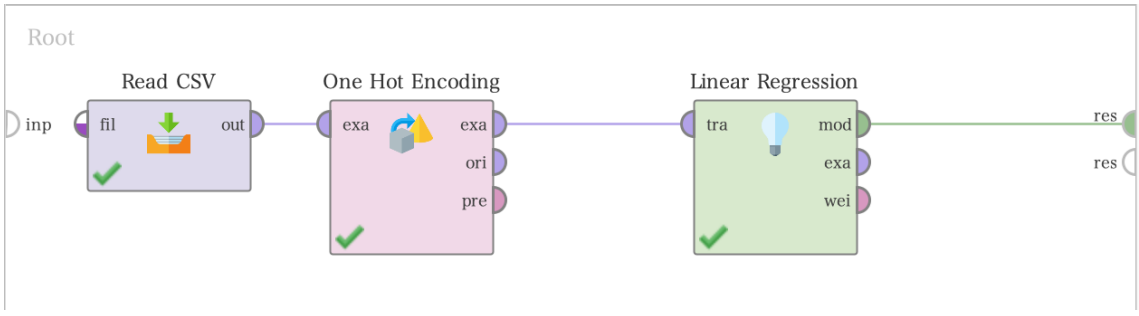
（４） 最も精度良く予測できる時、どの特徴量が選択されるのか。

線形回帰の係数算出

（１）、（２）の問題に関しては、Linear Regression オペレーターを使用することで特徴量の関係を探ることができます。まず、Read CSV のオペレーターを使って対象となる csv ファイルを読み込みます。読み込む際に、fuel economy をラベル設定することを忘れないようにします。また、今回の予測モデル生成に直接的に関係がない Model Year や Mfr Name、Division などの変数を対象から外しておくように設定する必要があります。読み込んだデータをそのまま Linear Regression に繋ぎたいところですが、トランスミッションタイプ、エアインスパイア方法、燃料タイプなどの属性はカテゴリカルな特徴量ですので、One Hot Encoding のオペレーターを使って、1,0 の形に変更します。1 つの列（transmission）に入っていた AMS、SA、M、A、CVT、SCV がそれぞれ列を形成し、1,0 の数値が与えられたことが確認できます。

下記の図のようにオペレーターを並べ、実行すると mod ポートから実行結果（回帰係数）がアウトプットされます。

Mfr Name	fuel_economy	transmission = AMS	transmission = SA	transmission = M	transmission = A	transmission = CVT	transmission = SCV
Honda	21	1	0	0	0	0	0
FCA US LLC	28	0	0	0	0	0	0
aston martin	21	0	1	0	0	0	0



Attribute	Coefficient	Std.Error	Std.Coefficient	Torerance	t-Stat	p-Value	code
transmission = AMS	-1.974	0.392	-0.085	1.000	-5.040	0.000	****
transmission = SA	-2.131	0.298	-0.179	0.996	-7.141	0.000	****
transmission = M	-1.704	0.355	-0.093	0.988	-4.807	0.000	****
transmission = A	-2.777	0.330	-0.200	0.941	-8.425	0.000	****
transmission = CVT	4.112	0.606	0.136	0.860	6.783	0.000	****
air_aspired_method = Turbocharged	2.264	1.898	0.191	1.000	1.193	0.233	
air_aspired_method = Supercharged	2.449	1.939	0.085	0.959	1.263	0.207	
air_aspired_method = Naturally Aspirated	4.042	1.910	0.336	0.996	2.117	0.034	**
air_aspired_method = Turbocharged+Supercharged	2.378	2.144	0.030	1.000	1.109	0.267	
regen_brake = Electrical Regen Brake	5.634	1.913	0.198	0.780	2.946	0.003	***
regen_brake = No	-6.242	1.927	-0.223	0.774	-3.240	0.001	***
Drive Desc = All Wheel Drive	0.823	0.464	0.062	0.993	1.775	0.076	*
Drive Desc = 2-Wheel Drive, Rear	1.382	0.462	0.104	0.888	2.991	0.003	***
Drive Desc = 2-Wheel Drive, Front	3.942	0.469	0.297	0.650	8.410	0.000	****
Drive Desc = 4-Wheel Drive	0.792	0.496	0.046	0.940	1.598	0.110	
Cyl Deact = N	-0.539	0.270	-0.031	0.879	-1.991	0.047	**
variable_valve = Y	1.994	0.594	0.059	0.996	3.358	0.001	****
fuel_type = Gasoline (Premium Unleaded Required)	-0.591	0.231	-0.043	0.952	-2.558	0.011	**
fuel_type = Gasoline (Regular Unleaded Recommended)	-0.986	0.232	-0.083	0.912	-4.258	0.000	****
fuel_type = Diesel, ultra low sulfur (15 ppm, maximum)	5.444	0.694	0.140	0.998	7.846	0.000	****
eng_disp	-2.796	0.086	-0.606	0.689	-32.632	0.000	****
num_gears	0.214	0.073	0.064	0.786	2.938	0.003	***
batt_capacity_ah	-0.372	0.110	-0.122	0.900	-3.390	0.001	****
(Intercept)	32.184	3.141	NaN	NaN	10.246	0.000	****

結果からは、ドライブトレインがフロント 2 駆動であることが、燃費にプラスの影響を与えていることが読み取れます。また、ブレーキ手法として、電気回生ブレーキの手法を備えていることも燃費にプラスの影響を与えているようです。一方、エンジン排気量は燃費にマイナスの影響を与えています。エンジン排気量が増えると燃費が悪化する傾向にあります。エンジン排気量の標準化係数（Std.Coefficient）の絶対値が最も大きい値を示していることから、エンジン排気量が燃費に最も影響を与える属性であると考えられます。また、トランスミッションタイプが「オートマ（A）」、「セミオートマ（SA）」である場合も燃費にマイナスの影響を与えていることが分かります。

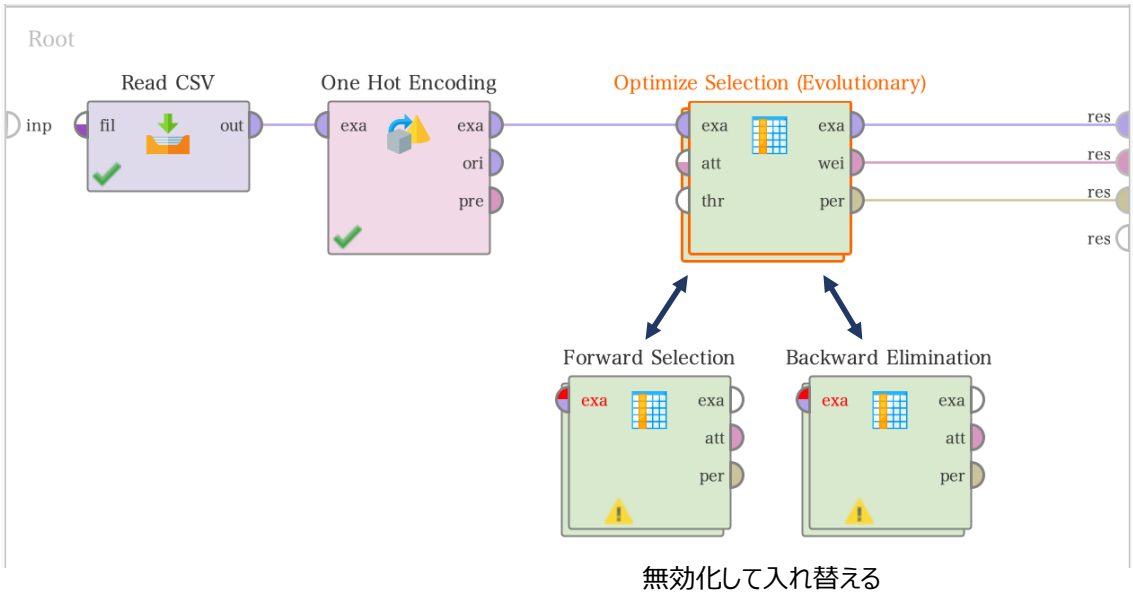
回帰における予測精度評価

機械学習における回帰問題の予測精度評価には、RMSE（二乗平均平方根誤差：Root Mean Squared Error）の指標が用いられることが一般的です。予測値と実際の値の誤差の二乗を平均して平方根をとったものであり、0 に近づけば近づくほど予測精度が高いことを意味しています。

$$\mathrm{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}$$

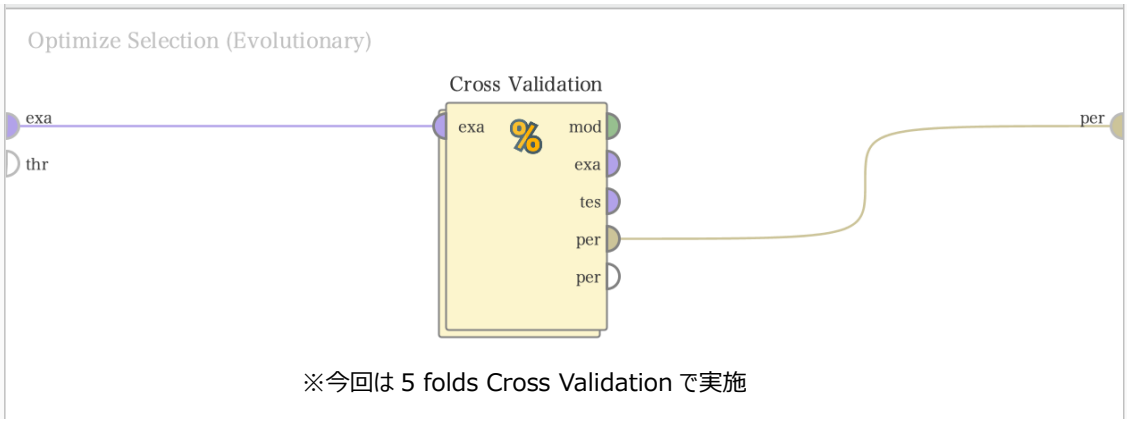
今回は RMSE を用いて、どのくらいの精度で新たな車種の燃費を予測できるのかを評価します。モデリングには、Linear Regression、SVM、k-NN のアルゴリズムを用い、変数選択手法は上記で解説を行った Forward Selection、Backward Elimination、Revolutionary Algorithm を用います。

■1 層目



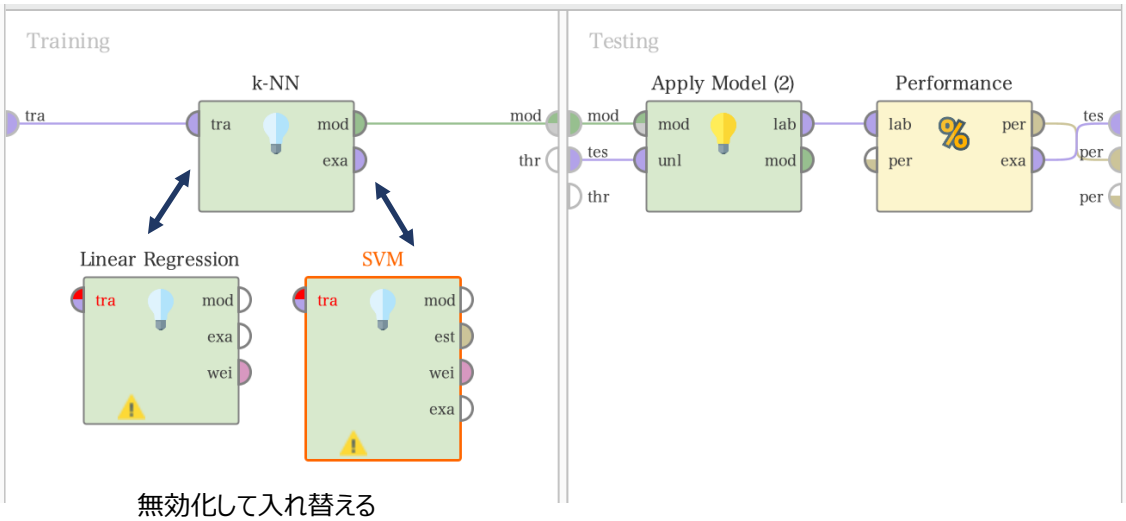
無効化して入れ替える

■2 層目



※今回は 5 folds Cross Validation で実施

■3 層目



無効化して入れ替える

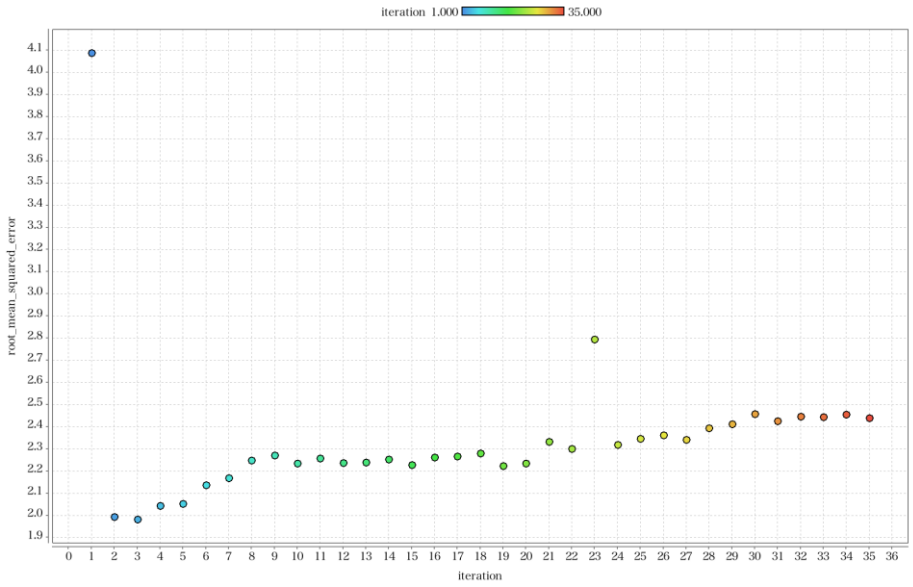
分析プロセスは、上記のように 3 層構造で作り上げます。具体的には、変数選択オペレーター（Forward Selection、Backward Elimination、Revolutionary Algorithm）の中に、Cross Validation を入れます。さらに、Cross Validation の中の左側（Training）に予測アルゴリズムを入れ、右側（Testing）でモデルのパフォーマンスを評価します。Performance オペレーターは複数ありますが、今回は“Performance（Regression）”を用い、同オペレーターのパラメータの”root mean squared error”にチェックを入れておきます。

3 つの予測モデル、3 つの変数選択アルゴリズムを入れ替えて、実行した結果が下記の表です。予測アルゴリズムとしては、k-NN（最近傍法）が今回のデータセットに対して、他の予測アルゴリズムと比べ、当てはまりがいいことが分かります。変数選択の手法に注目すると、Forward Selection（FS）、Backward Elimination（BE）よりも Revolutionary Algorithm（RA）の変数選択アルゴリズムを用いた方が、RMSE が小さくなっていることが確認できます。特筆すべき点として、電気回生ブレーキを備えていること、エンジン排気量は全ての手法で特徴量として選択されており、RMSE を小さくするという観点で重要な特徴量であったと言えるでしょう。

アルゴリズム	Linear Regression			SVM			k-NN		
変数選択の手法	FS	BE	RA	FS	BE	RA	FS	BE	RA
RMSE	2.661	2.653	2.673	2.821	2.829	2.945	2.142	2.148	2.059
Number of Attribute	13	25	19	14	21	11	9	25	17
transmission = AMS	0	1	0	0	1	0	0	1	0
transmission = SA	0	1	0	1	1	0	0	1	1
transmission = M	0	1	1	0	1	0	0	1	0
transmission = A	1	1	1	1	1	0	1	1	0
transmission = CVT	1	1	1	1	1	1	0	1	1
transmission = SCV	1	1	1	0	0	1	0	1	1
air_aspired_method = Turbocharged	1	1	1	1	0	1	0	1	1
air_aspired_method = Supercharged	0	1	0	0	1	0	0	1	0
air_aspired_method = Naturally Aspirated	1	1	1	0	1	0	1	1	1
air_aspired_method = Turbocharged+Supercharged	0	1	1	0	1	1	0	1	1
regen_brake = Electrical Regen Brake	0	1	0	1	1	0	0	1	0
regen_brake = No	1	1	1	1	1	1	1	1	1
Drive Desc = All Wheel Drive	0	1	1	1	1	1	1	1	0
Drive Desc = 2-Wheel Drive, Rear	1	1	1	0	0	0	0	1	0
Drive Desc = 2-Wheel Drive, Front	1	1	1	1	1	1	0	1	1
Drive Desc = 4-Wheel Drive	0	1	0	0	0	0	1	1	1
Cyl Deact = N	0	1	1	0	1	0	1	1	1
variable_valve = Y	1	1	0	0	1	1	0	1	1
fuel_type = Gasoline (Premium Unleaded Required)	0	1	1	1	1	1	1	1	0
fuel_type = Gasoline (Premium Unleaded Recommended)	1	1	0	0	0	0	0	1	1
fuel_type = Gasoline (Regular Unleaded Recommended)	0	1	1	1	1	1	0	1	1
fuel_type = Diesel, ultra low sulfur (15 ppm, maximum)	1	1	1	1	1	0	0	1	1
eng_disp	1	1	1	1	1	1	1	1	1
num_cyl	0	1	1	0	1	0	0	1	1
num_gears	0	1	1	1	1	0	1	1	1
batt_capacity_ah	1	0	1	1	1	0	0	0	0

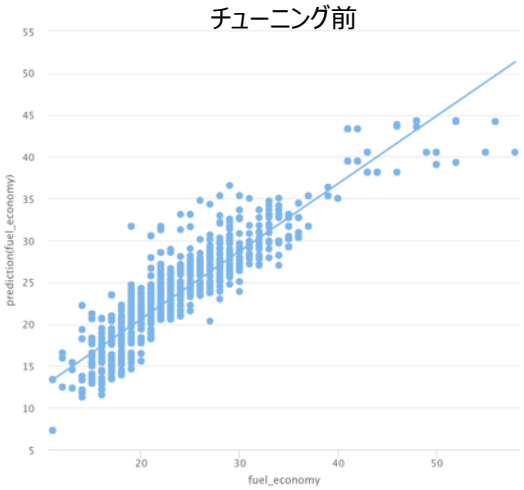
※1 変数選択の手法：FS＝Forward Selection、BE＝Backward Elimination、RA＝Revolutionary Algorithm
※2 テーブル内の0は使用していない変数を表し、1は使用した変数を表す。
※3 グレースケールは全ての手法で選択された変数。

今回のデータセットと相性が良かったとされる k-NN ですが、右の 3 パターンは全て k＝5 で試した結果です。しかしながら、k=5 が最も RMSE を小さくする値なのでしょう？grid search オペレーターを使用することでハイパーパラーメータを最適化することができます（次章参照）。K の値を 1～35 で試した時、RMSE を最も小さくできる値を探索した結果が下記の図です。k=2 または k=3 を選択した時、RMSE は 2.0 を下回り、誤差を比較的小さくすることができました。K の値を大きくするにつれて、RMSE は上昇する傾向にあることが確認できます。

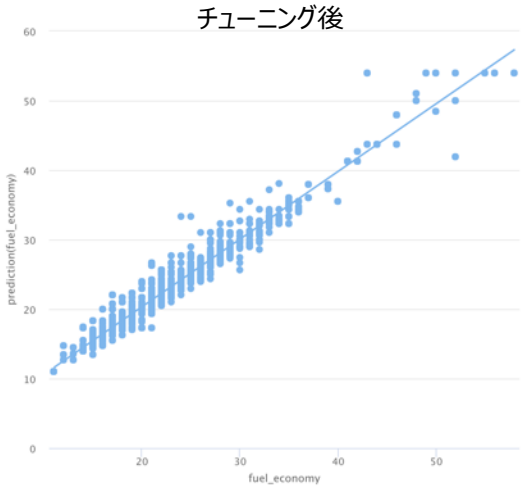


最後に、縦軸に予測値、横軸に実測値を取り、どの程度上手く予測できているのかプロットしてみました。左側が単純に linear regression を当てはめた場合、右側が変数選択を行い k-NN(k=3)で当てはめた場合です。チューニング後の方がチューニング前より上手く当てはまっている様子が見て取れます。

チューニング前



チューニング後



まとめ

機械学習における特徴量選択に関して、全ての属性の組み合わせ 2^n-1（n は属性の数）に対して変数選択を実施し精度を確認することは、データセットが大きくなればなるほど計算量が多くなるため難しくなります。その解決策と提示されたのが、Forward Selection や Backward Elimination という変数選択手法でした。しかし、それら変数選択手法も万能ではなく、局所最適となっている可能性が高いという問題点がありました。全体最適となる変数選択手法として、提案されたのが Evolutionary Algorithm であり、自然進化を模倣した変数選択アルゴリズムです。

MPG（燃費）を予測する回帰問題を例にとり、変数選択手法の評価を行いました。予測アルゴリズムに関しては、k-NN（最近傍法）との相性がよく、変数選択手法については、Evolutionary Algorithm を用いた場合、RMSE が最も小さくなることを確認できました。回帰係数ならびに各手法により選択された変数からもエンジン排気量、電気回生ブレーキの手法を備えているかどうかが MPG（燃費）に比較的強く影響を与えている要因であることが明らかになりました。

今回は、特徴選択に焦点を当て、モデルの精度向上を試みましたが、特徴量設計自体の見直し（特徴量生成や次元圧縮など）やハイパーパラメーターの最適化などの方法を用いることで、モデルの精度向上の余地があるものと考えられます。

Reference

Ingo Mierswa（2018）「Better Machine Learning Models with Multi-Objective Optimization」.

A Sequential Process Monitoring Approach using Hidden Markov Model for Unobservable Process Drift - Scientific Figure on ResearchGate. Available from: https://www.researchgate.net/figure/11-Illustration-of-local-optimum-and-global-optimum_fig16_306558608 [accessed 16 Aug, 2019]

Data

The raw data is located on the EPA government site.Modified data set is here.