テーブルデータの解析

澤木陽人

April 11, 2023

Contents

1 概要

本問題を解くにあたって、問1では単純なパーセプトロンによる分類でテストした。また、基礎分析と問1の結果を鑑みて、単純な境界決定では精度があまり出ないことがわかったため、問2ではLightGBMによるアンサンブルを行った。さらに、問2では更なる精度向上を目指すべく、誤検出の多いクラス0,1のみを学習したモデルで部分的に再テストを行うというフローを採用することで、より良い精度を達成することができた。最終的な最高精度は、データセットの2割をテストデータとして、0.948であった。

ここでは、実際に手を動かした順番とは少し異なるが、わかりやすさのために、基礎分析・問1・問2のアプローチの順で分析をまとめる。gitにアップしたコードは実験の痕跡がわかりやすいよう、実際に実験した際の手順に沿ってそれぞれの推論結果を残しているため、考察等整理された実験結果については主にこちらのレポートを参照されたい。

2 基礎分析

基礎分析では、各特徴量の分散や順序統計量、散布図行列、相関行列などを俯瞰し、本分析の作戦を練るのに用いた。本レポートでは特に、散布図行列及び相関行列についてまとめる。

2.1 散布図行列

今回扱ったCovTypeデータセットは、植生タイプ7種類のカバレッジについて、54種類の各測定値や属性が与えられている。植生タイプは、元データセットでは1-7でラベリングされているが、本レポートでは一貫して0-6で表記する。なお、54種類の特徴量のうち、40種類は土壌のカテゴリカル属性をone-hotに直しているだけのため、実質的に特徴量は15種類程度となる。また、データセットには偏りがある。

CovType	count
Type 0	211840
Type 1	283301
Type 2	35754
Type 3	2747
Type 4	9493
Type 5	17367
Type 6	20510
Total	581012

Table 2.1: データセットに含まれる各植生タイプ

各データからランダムに1000件を抜き出し、植生タイプごとのペアプロットを

作成した。対角成分にはその特徴量分布のカーネル密度推定を描画している。後のLightGBMによる分析によって、特に重要と思われる特徴量がElevation/Horizontal Distance To Roadways/Horizontal Distance To Hydrolgy/Vertical Distance To Hydrology/ Horizontal Distance To Firepoints の5つであることが判明するため、ここではその5つを図示したものを示す。土壌タイプを除く全ての特徴量のプロットは巻末に示す。

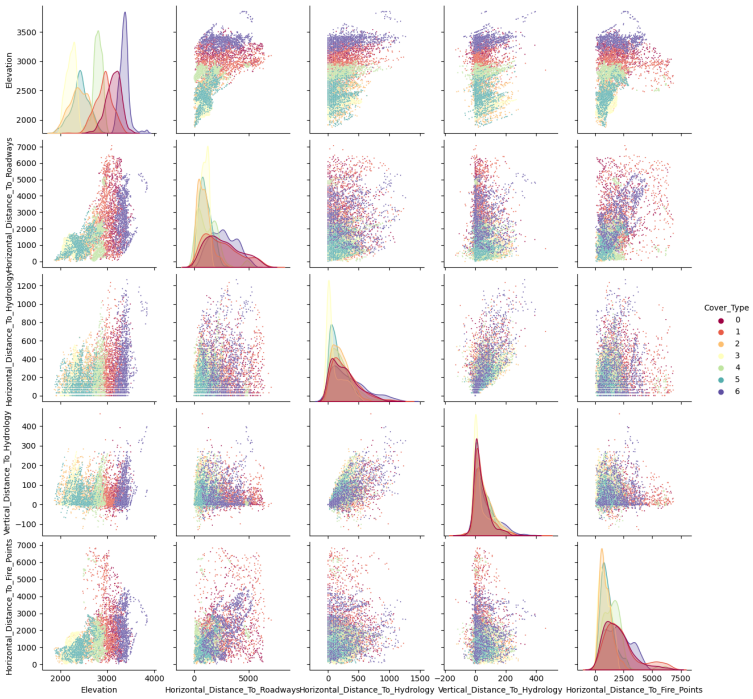

Figure 2.1: 重要特徴量の散布図行列

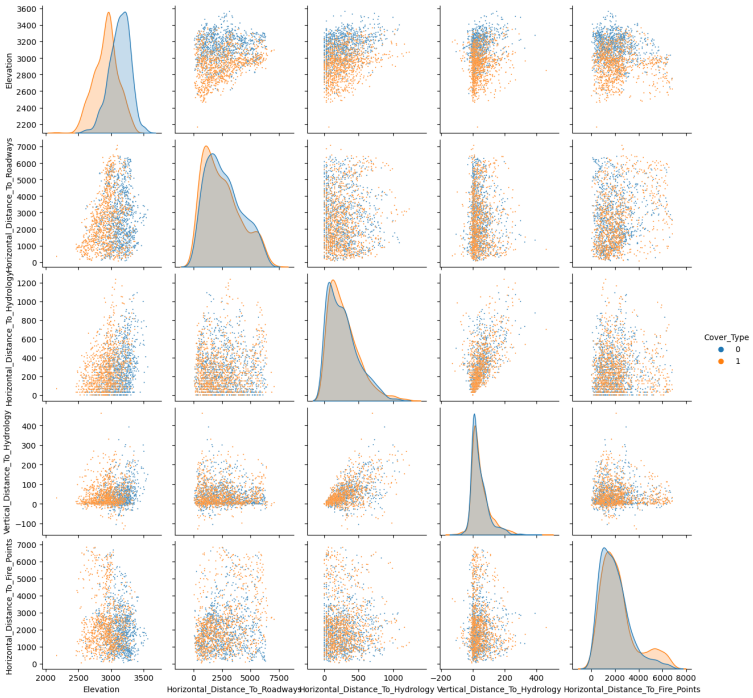

Figure 2.2: 重要特徴量の散布図行列（植生タイプ 0/1）

これを見るに、高度と道路との距離は植生ごとの大きな違いが認められる。しかし、植生タイプが0のものと1のものは高度を除いてどの特徴量に対しても比較的似ており、しかもそれがデータセットの8割以上を占めることが、ここからの分析に苦しむ要因となった。実際に、同じ図を、植生タイプ0・1に絞って表示したものがFig2.2である。激しくプロットが混ざり合っていることから分離が難しく、SVMでの境界決定ではソフトマージンのハイパラをかなり高く設定することになった。（余談：植生タイプ0と1はどちらもマツ科の植物で、難しいに決まっている。私は大の虫取り好きなので年中山に入っているが、例えば同じブナ科のアベマキ・ミズナラ・クヌギは同じような場所に群生していることが多く、いまだにクヌギの群生地を見つけるのが難しい。虫を取るならクヌギがベストだが、遠目ではどれも似ていてわかりにくく、毎年虫取りスポット選びに苦しめられている。まさかこのデータセットで同じようなもどかしさを感じることになるとは・・・）

2.2 相関行列

次に相関行列を調べた。土壌タイプは元はカテゴリカルデータであるため、相関は大して参考にならず、図示していない。以下の図は上から順に、全データに対する相関行列、植生タイプ0に対する相関行列、植生タイプ1に対する相関行列を示す。さて、数値で見てもやはり植生タイプ0と1の間に大きな違いを認めることは難しく、高い値を示している部分を見ても、あまり注目に値するところは少ない。時間ごとHillShade間で負の相関が出たり、Aspectと高い相関が出るのは太陽の移動があるから自然であり、水源までの水平距離と垂直距離の相関が大きいのも、そもそもの水源からの三次元的な距離に比例するから自然である。今回はSVMや決定技のブースティングを手法に選択しているため、あまり多重共線性に気を配らずに分析を行なってしまったが、DNNベースの手法だとこの辺りの明らかな相関にはもう少し繊細に特徴量選択をする必要がありそうだと感じた。

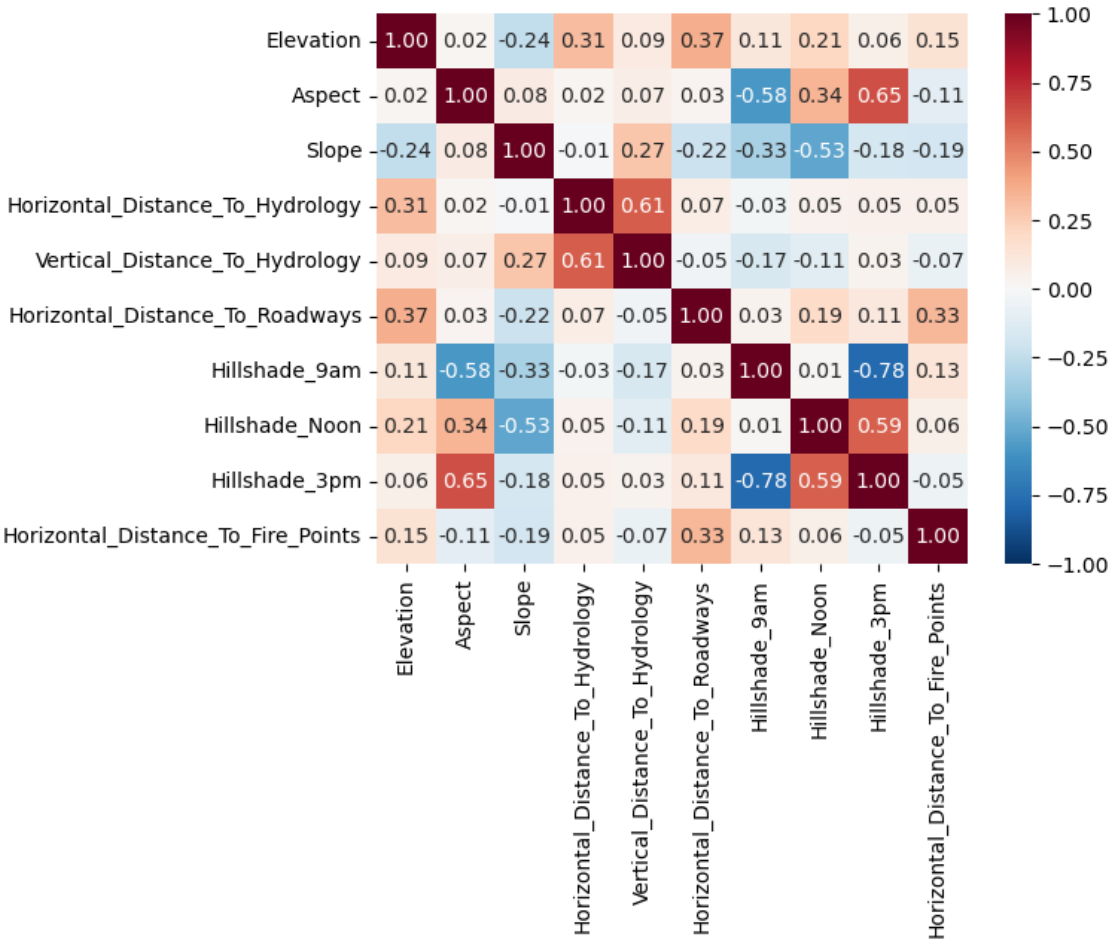

Figure 2.3: 特徴量の相関行列

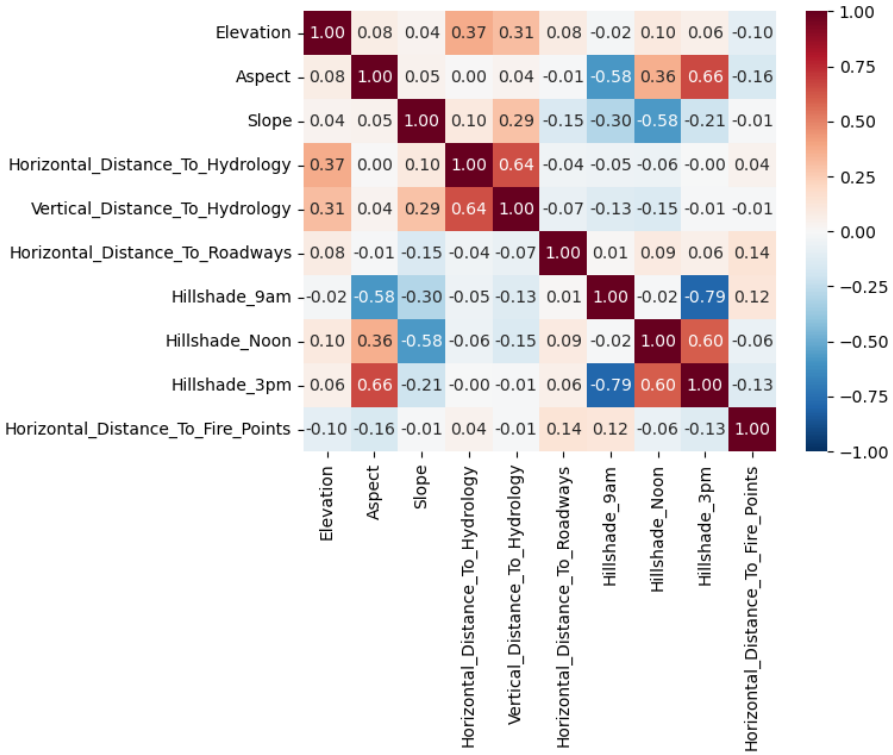

Figure 2.4: 特徴量の相関行列（植生タイプ0）

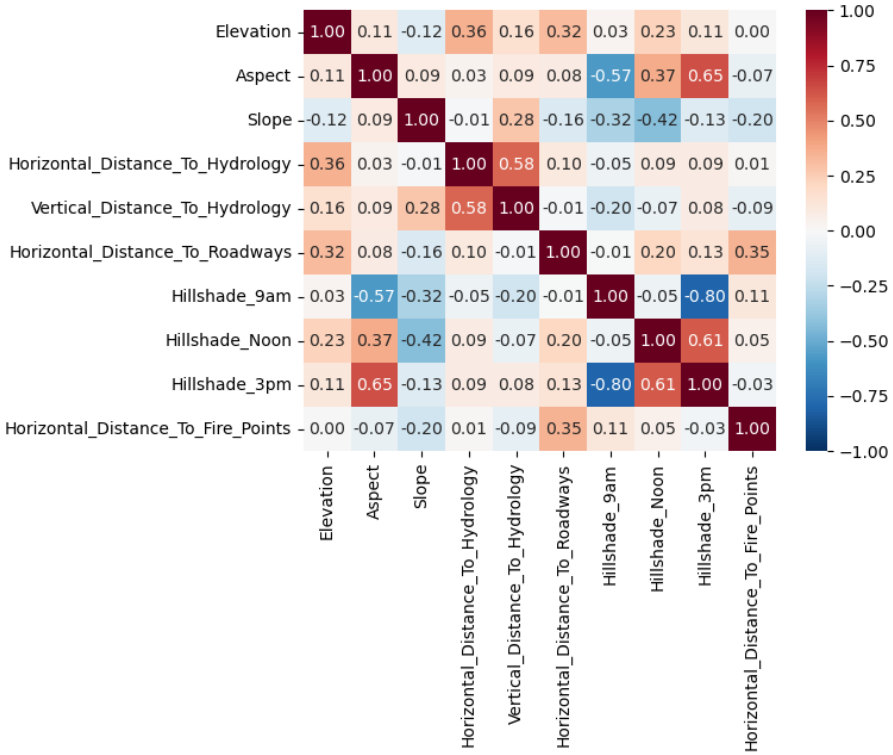

Figure 2.5: 特徴量の相関行列（植生タイプ1）

3 シンプルなNNによる分類（問1）

3.1 交差検証

まずはどの程度精度が出るデータセットかの実験の意で、非常にシンプルな多層パーセプトロンによる実験を行った。中間層2層、いずれのノードも30とし、150イテレーションで5分割の交差検証を行った。それぞれの交差検証による正答率は以下であった。

Folds	accracy
Fold 1	0.7695
Fold 2	0.7779
Fold 3	0.7729
Fold 4	0.7862
Fold 5	0.7803

Table 3.1: 各分割の正答率

3.2 誤分類データの内訳

予測したテストデータのうち間違っていたものが、何と何を分類し損ねたかを図に示したものが以下である。驚くべきことに、誤ったデータの大半は植生タイプ0と1を互い違いに分類していることがわかる。NNはある意味データさえ突っ込んでしまえば結果を待つのみであるので、分類結果にこれ以上の解釈を与えるのは難しいが、基礎分析の通り特徴量がかなり混ざりあっているクラスであるゆえに、表現力が足りない2層程度のモデルでは分離に無理があったと考えられる。

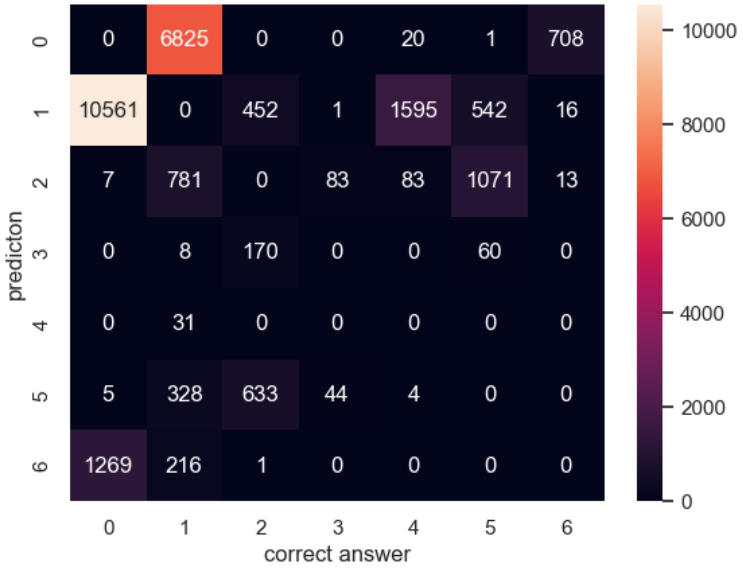

Figure 3.1: 誤分類結果

3.3 標準化して再チャレンジ

決定木ベースの手法ではあまり意味がないが、NNベースの場合は特徴量ごとの値の大きさはある程度スケーリングしておいた方が学習が進み精度が高くなることが多い。上での学習はそのままデータセットを入れていたため、精度が少しでも上がることを願って、特徴量の標準化を行った上で再検証した。前節までと同様に正答率と誤分類表は以下に示す。

Folds	accracy
Fold 1	0.8476
Fold 2	0.8586
Fold 3	0.8514
Fold 4	0.8527
Fold 5	0.8503

Table 3.2: 各分割の正答率（標準化後）

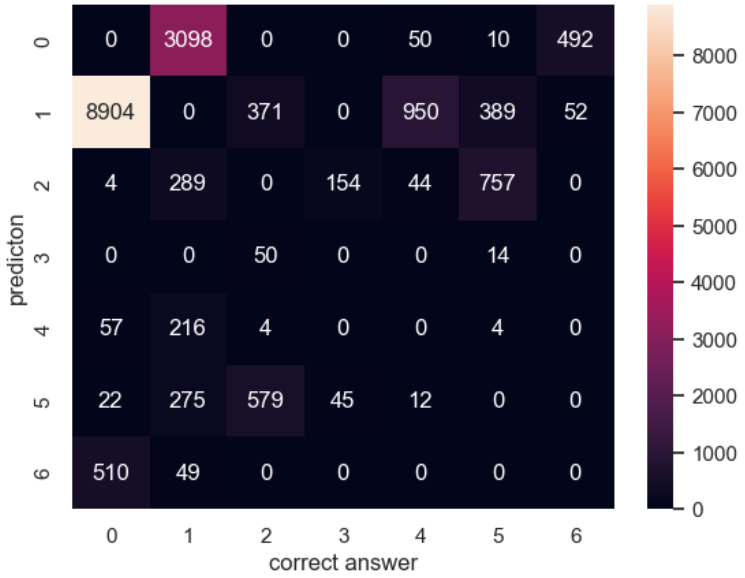

Figure 3.2: 誤分類結果（標準化後）

なんと全体的に10%近い精度向上が見られた（正直驚いた）。ほとんどのクラスの誤分類は6割程度に減ったが、やはり植生タイプ0の分類は難航している様子が見受けられる。さて、標準化によってここまでの精度向上が見られたが、これを「標準化はモデルの表現力が高くなる!」と結論づけるのはやや尚早で、おそらく元のデータでも十分なイテレーションで学習を回せば、もう少し精度があがるものだと考えられる。多層パーセプトロンはあくまでノードの値の重みを誤差逆伝播で修正しているだけであるので、Elevationのような大きな数値の特徴量は学習が進み十分に重みが小さく収束するまでは（重みと積を取る前から値が大きいので）圧倒的に大きな影響力を持つことになる。ここでの精度が10%と非常に有意に向上したのは、どの特徴量も”最初からほぼ公平な”影響力を持って学習を開始できるからだと思われる。何あれ、学習に時間がかかるため、学習イテレーションごとの分析は今回は割愛することにする。

4 LightGBMによる分類と分析（問2前半）

4.1 Light GBMによる交差検証

問2ではLightGBMを用いたアプローチの試行錯誤についてまとめる。LightGBMは決定木ベースの勾配ブースティングを用いたモデルであり、パラメータチューニングの手間が少ないことや、本データのようなテーブルデータの多値分類問題に有用であると考え採用した。

交差検証では分割数は5回とし、ハイパラは手元での実験の末、学習率0.06、 num leaves(木の最大葉数)=30, min data in leaf(葉の最小数)=15とし、各1200ラウンドで学習した。学習データとテストデータは原則8:2で分割しており、最終的なテストデータサイズは116203である。

各Foldにおける損失曲線は以下のようになった。損失にはmulti-logLossを用いた。そこそこしっかり学習されている様子が見られる。平均してテストデータの正解率で0.93〜0.94を記録しており、これだけでも悪くない精度が出た。

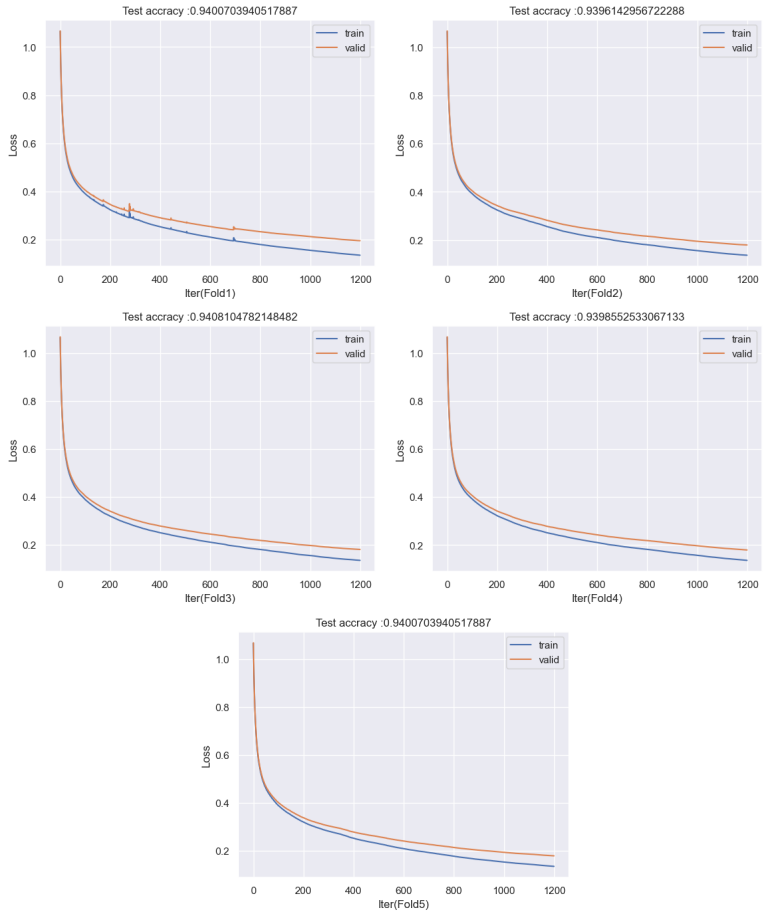

Figure 4.1: 損失曲線

Folds	accracy
Fold 1	0.9401
Fold 2	0.9401
Fold 3	0.9395
Fold 4	0.9402
Fold 5	0.9397

Table 4.1: 各分割の正答率（LightGBM）

また、特徴量ごとの重要度を表すと以下のようになった。やはり、高度や水源までの距離、道路や発火地点までの距離が重要な要素であり、大きな比重が置かれている。土壌タイプいずれも大きな影響力は持っていないように思われるが、これは後に考察す

る。

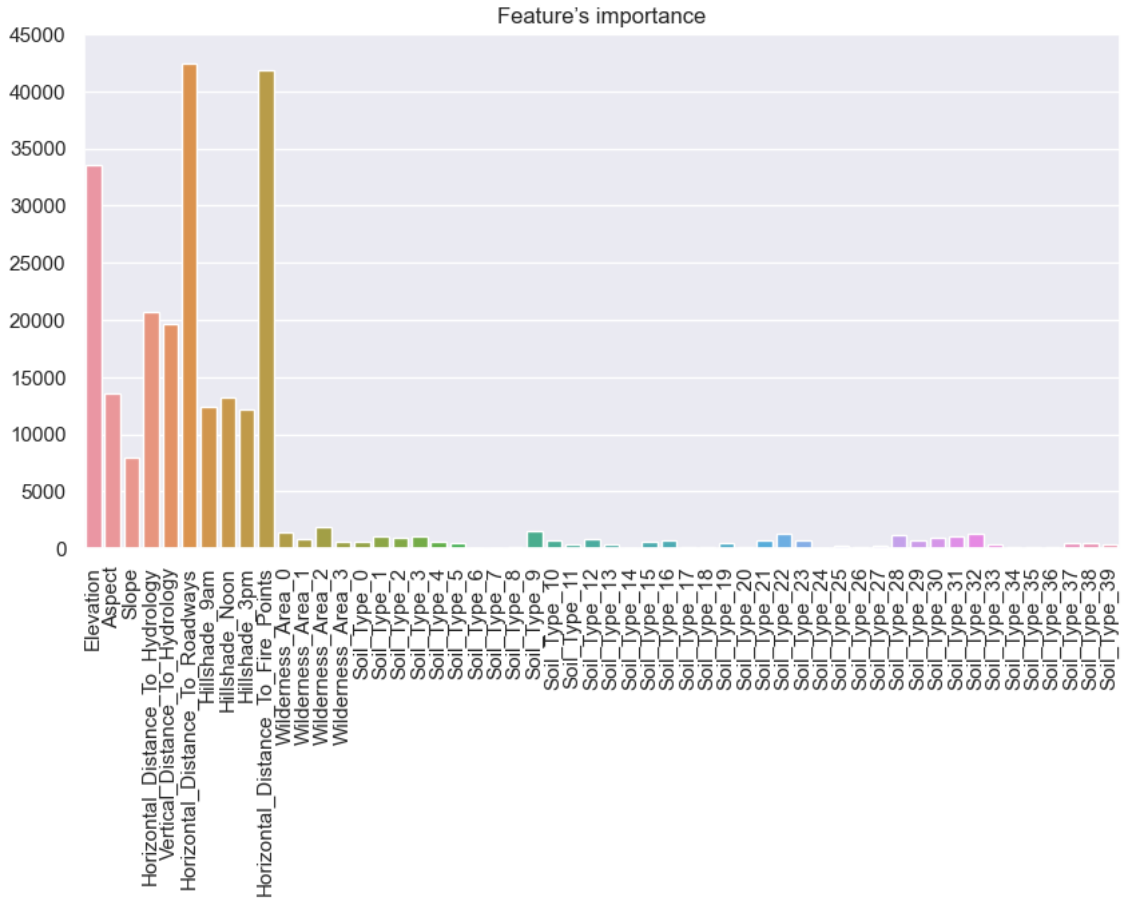

Figure 4.2: 特徴量ごとの重要度

4.2 誤分類データの内訳

間違えた6%はどのようなデータであろうか。前章と同様に、誤分類データの可視化を行った。やはり、植生タイプ0と1がかなりの鬼門のようで、誤り全体に占める8割が相変わらずこの2クラスで生じている。

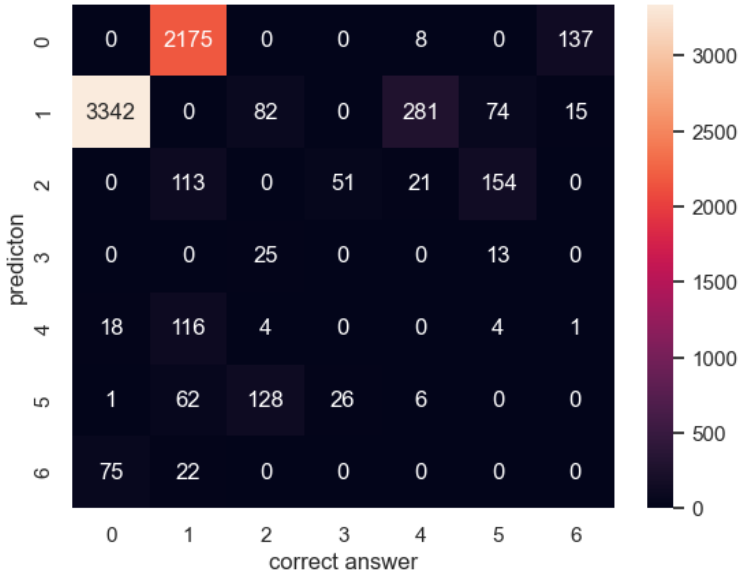

Figure 4.3: 誤分類結果

ここまでは、他の植生タイプ間の重要な特徴量が主に距離や高度に関するものであることについて触れてきたが、前述の通り、高度以外の距離特徴量は植生タイプ0と1の間で大きな違いが見られない。そのため、誤分類した植生タイプ0と1について、土壌タイプに何か差異があるかをさらに分析することにした。土壌タイプは全部で40種類あり、それぞれのデータについて各一つが割り振られている。植生タイプ0・1について、誤分類されたデータと、データセット全体で土壌タイプの分布が大きく異なるようであれば、そこに注目して精度向上を目指せるのではと考えた。先ほどの特徴量重要度では土壌タイプはほとんどが1500以下と低かったが、これはカテゴリカルな土壌タイプを40項目に分けて立項しているために、データ一つ一つについて、ほとんどの他の土壌の選択肢は重要でないとみなされてしまい低くなったものと考えられる。言い換えれば、土壌タイプ全体が持ちうる重要度は、一つ一つに表れている重要度と比較して十分に大きく、重要なデータであると考えたのである。そこで以下は、データセット全体中の植生タイプ0・1と、テストデータで誤分類した植生タイプ0・1の、それぞれの土壌タイプの

ヒストグラムである。残念なことに、全体としての分布は似通っており、特に誤分類したデータの分布の違いはなおさら小さかった。植生タイプ0には土壌タイプ34以降のデータが、植生タイプ1には土壌タイプ12番以前のデータが多く含まれているという顕著な違いはあるが、これは正しく分類されているものが多いことがわかる。よって誤分類の多くは共通の土壌タイプを持つ部分で発生しており、土壌タイプから精度向上を目指すアプローチはあまりうまく行かなそうだと結論された。

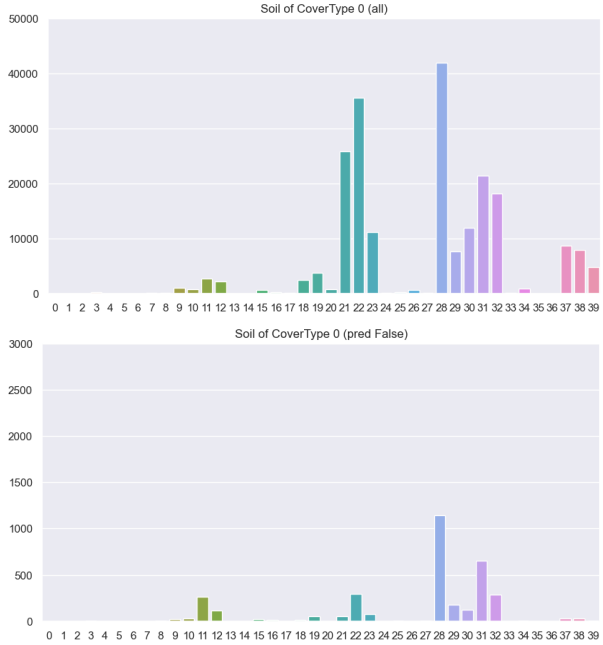

Figure 4.4: 植生タイプ0の全データと誤分類データの土壌タイプ度数分布

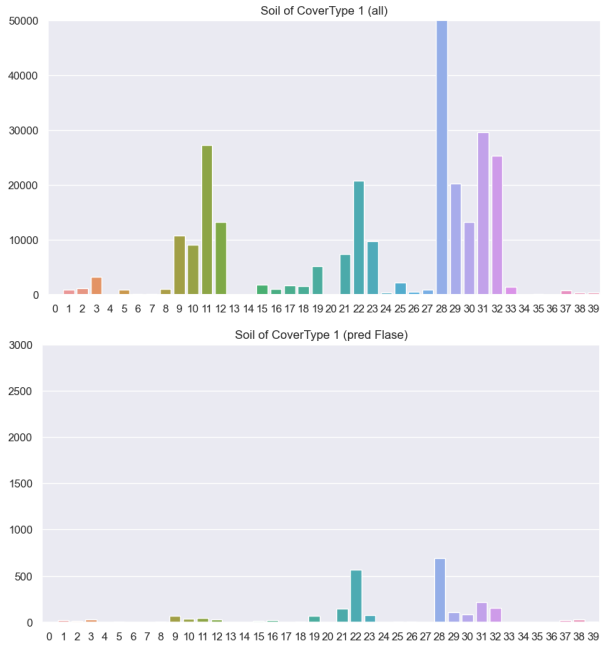

Figure 4.5: 植生タイプ1の全データと誤分類データの土壌タイプ度数分布

5 アンサンブルと再テストフロー（問2後半）

5.1 推論のモデル

ここまでの内容を振りかえると、LightGBMのハイパラチューニングと交差検証だけでもそこそこの性能が出るが、植生タイプ0・1の分類に課題があることがわかった。そのため、あらかじめ学習データのうちタイプ0と1のみを学習させた別のモデル（以下2値モデルという）を用意し、全体で学習させたモデルが0と1の判別に迷った場合には、そのデータを2値モデルで再判別するという手順で推論を行う。また交差検証によって各モデルは5つに分割されるため、推論はさらにこの5つのモデルのアンサンブルとして行う。以下は学習から推論までの流れを図にしたものである。

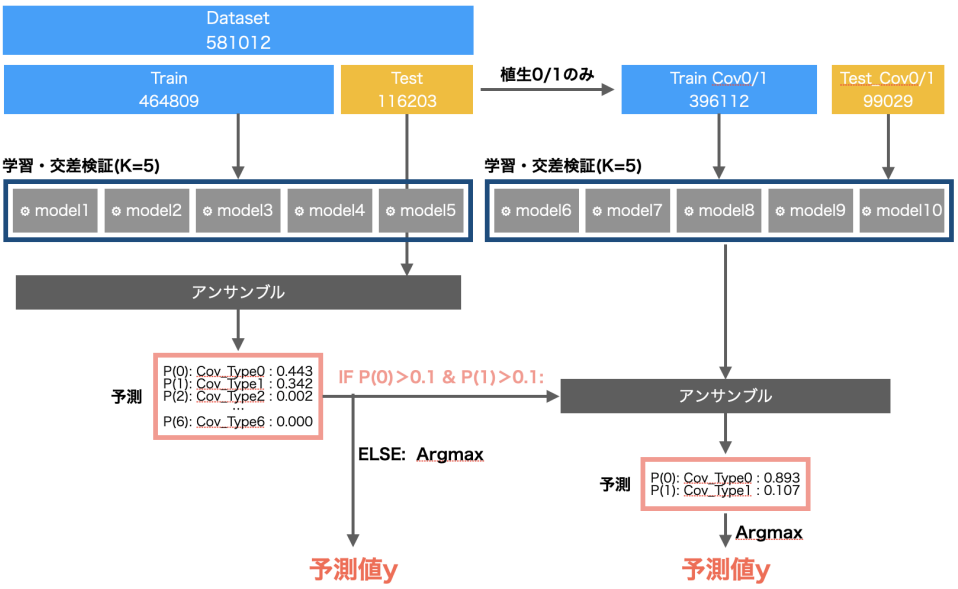

Figure 5.1: 推測モデルの構成

5.2 学習と推論結果

全体で学習したモデルは前章の通りであるが、植生タイプ01のみで学習した2値モデルの学習は平均して0.949程度と、わずかに高くなった（むしろ2値に特化しても5%も分類できていない）。学習の損失曲線及び正解率は以下である。

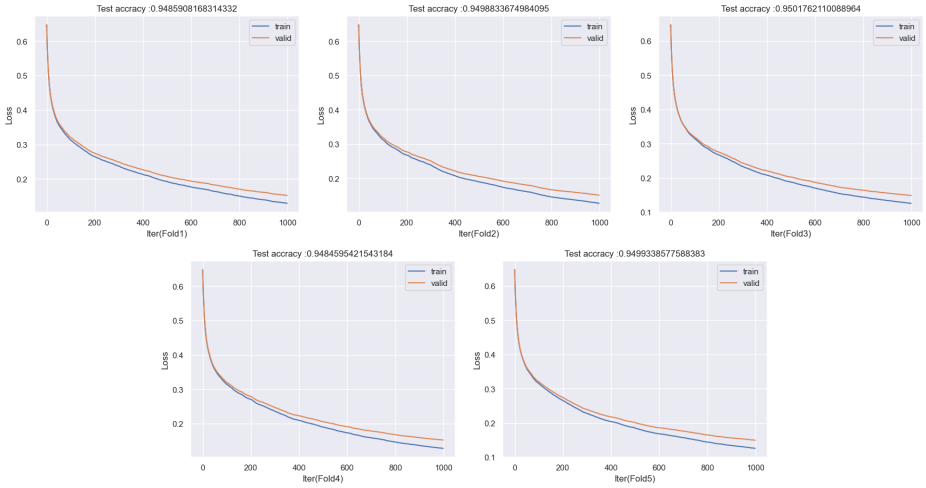

Figure 5.2: 損失曲線

Folds	accracy
Fold 1	0.9486
Fold 2	0.9499
Fold 3	0.9502
Fold 4	0.9485
Fold 5	0.9499

Table 5.1: 各分割の正答率（2値モデル）

さて、この手法では植生タイプ0/1の分類確率がともに0.1以上になったデータを、2値モデルで再テストしているが、そもそも再テストになったデータの正解ラベルが2-6である可能性もある。そのようなデータは確実に誤分類となってしまう欠点があるため、しっかりそこもチェックしておきたい。各正解数や再テストデータに含まれるそういったデータの数について以下図及び表にまとめる。

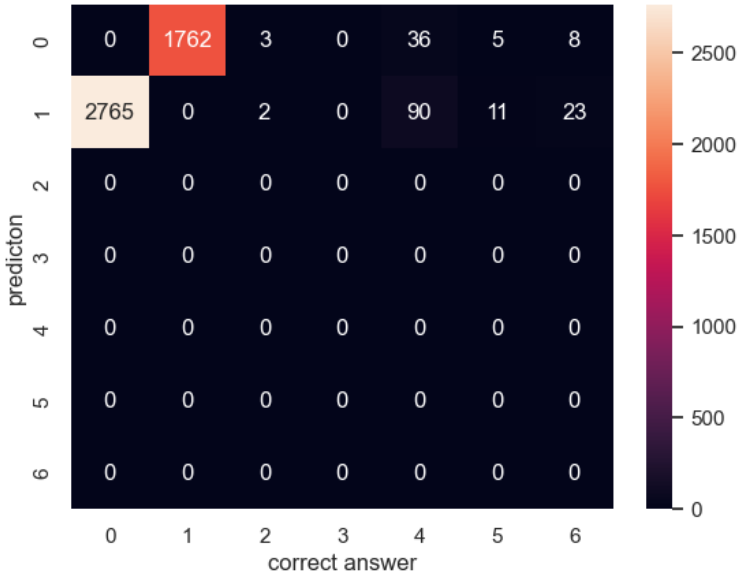

Figure 5.3: 誤分類結果（再テスト）

項目	データ数	正解数	精度
全データ	116203	110113	0.9478
再テスト以外	78337	76972	0.9826
再テスト	37866	33161	0.8757
再テスト中のCovType0	18147	15382	0.8476
再テスト中のCovType1	19541	17779	0.9098
再テスト中のCovType2	5	0	0.0000
再テスト中のCovType3	0	0	0.0000
再テスト中のCovType4	126	0	0.0000
再テスト中のCovType5	16	0	0.0000
再テスト中のCovType6	5	0	0.0000

Table 5.2: 各正答率

6 失敗したこと・やりたかったこと

6.1 SVMによる分類

問1では、最初はテーブルデータによる解法としてSVMがシンプルだろうと考えて手を動かしていた。しかし、SVMの計算量は$O(n^2)$であり、ハイパラのチューニングまでは行ったものの、50万件近い学習データはいつまで経っても学習が終わらず断念した。チューニングでは学習データとテストデータをランダムに10000件、2000件選択し、それらを用いてサーチを行った。正則化項C、カーネル（rbf・sigmoid)、非線形カーネルのgamma値、決定関数の形状（ovo・ovr）についてグリッドサーチを行った。

Hyper Params	value
C	0.01
kernel	rbf
γ	0.1
decision function shape	ovr

Table 6.1: グリッドサーチで決定したハイパーパラメータ

6.2 tabnetによる分類

問1ではNNモデルで簡単な実験を行ったため、問2では同じくNNベースのモデルで比較検討を行いたいと考えていた。長らくKaggleではXGboostや今回扱ったLightGBMなどの決定木ベースのモデルがテーブルデータ解析の定石となっていたが、最近はtabnetと呼ばれるNNベースのモデルが注目を集めつつある。今回の提出課題においては、問2では「複数のアプローチの比較検討」が求められていたため、問1におけるNNと問2にお

けるLightGBMの2パターンを比較した今回の内容に加えて、前述のSVMと、tabnetの考察まで含め、精度向上に取り組みたかった。しかし、どうしても時間が足りず今回は提出期限までの実装を断念した。

7 参考

コーディングで参考にしたページやリファレンスなど。

- sklearn.neuralnetwork.MLPClassifier
  - https://scikit-learn.org/stable/modules/generated/sklearn.neural_network.MLPClas
- GridSearchCV(scikit-learn)によるチューニング
  - https://starpentagon.net/analytics/scikit_learn_grid_search_cv/
- 【Python覚書】LightGBMで交差検証を実装してみる
  - https://potesara-tips.com/lightgbm-k-fold-cross-validation/
- microsoft LightGBM
  - https://github.com/microsoft/LightGBM
- 【Python覚書】LightGBMで交差検証を実装してみる
  - https://potesara-tips.com/lightgbm-k-fold-cross-validation/