PETFINDER.MY · PAWPULARITY CONTEST

愛らしさスコアを予測する
SVR × Deep Learning アンサンブル解法

事前学習済み特徴量 + RAPIDS cuML SVR による “1st place Magic” の再現、及び BCE 損失による深層学習モデルとのアンサンブル

17.09524	17.94270	17.1306
PRIVATE SCORE	PUBLIC SCORE	

戦略設計: Google Gemini DeepResearch（Kaggle 上位解法分析）に基づく実装記録

EXECUTIVE SUMMARY

4つの柱で構成する解法

01

事前学習済み特徴量 × SVR

timm/CLIP など9モデルの画像埋め込みを RAPIDS cuML SVR（RBFカーネル）に投入する“1st place SVR Magic” を踏襲

02

Forward Feature Selection

11候補モデルから貪欲法でCV RMSEが改善する組み合わせだけを採用。17.516 → 17.141 まで改善

03

BCE損失によるDL学習

歪んだ分布に対する疑似最尤推定として二値交差エントロピー損失を採用。beit_large_224 を10-fold学習

04

SVR × DL アンサンブル

重み探索により SVR 0.93 : DL 0.07 のブレンドが最良。最終 CV RMSE 17.1306 を達成

1. DATASET

データセット概要

9,912	8→約6,800	1–100	12種
学習用画像数	テスト画像数（Code Competition）	Pawpularity スコア範囲	画像メタデータ列

目的変数: Pawpularity（平均 38.04 / 中央値 33.0 / 標準偏差 20.59）— 保護動物サイトの実際のクリック反応から算出された人気度スコア。テストセットは Code Competition のため提出時のみ実データ（無インターネット環境）に差し替わる。

メタデータ列（Subject Focus / Eyes / Face / Near / Action / Accessory / Group / Collage / Human / Occlusion / Info / Blur）は主催者が事前算出した0/1フラグ

評価指標: RMSE（Root Mean Squared Error）

データプレビュー：train.csv 先頭5行

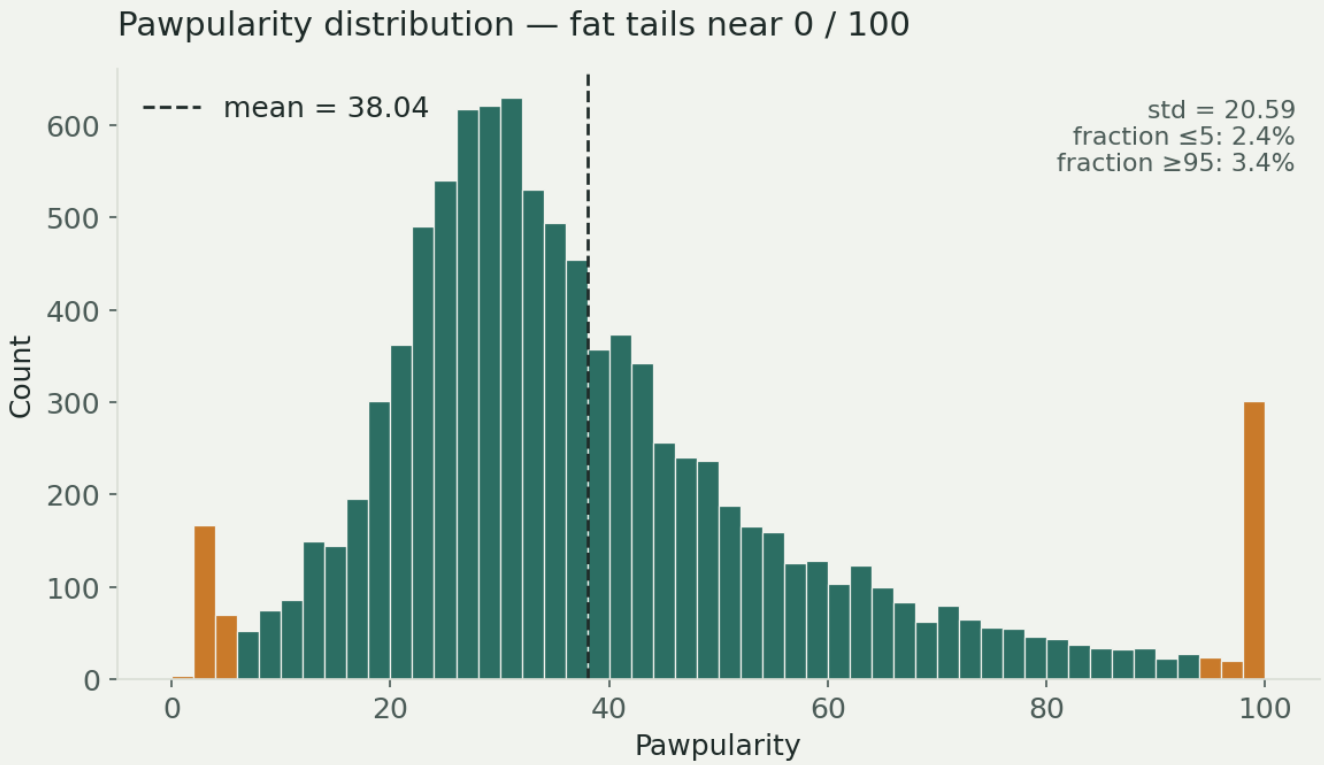
Id と目的変数 Pawpularity の間に、主催者が画像から事前算出した12種のメタデータフラグ（0/1）が並ぶ。列ごとに色分けして構造を示す。

Id	Subject Focus	Eyes	Face	Near	Action	Accessory	Group	Collage	Human	Occlusion	Info	Blur	Pawpularity
0007de1884…	0	1	1	1	0	0	1	0	0	0	0	0	63
0009c66b94…	0	1	1	0	0	0	0	0	0	0	0	0	42
0013fd999c…	0	1	1	1	0	0	0	0	1	1	0	0	28
0018df346a…	0	1	1	1	0	0	0	0	0	0	0	0	15
001dc955e1…	0	0	0	1	0	0	1	0	0	0	0	0	72

全9,912行×14列。Subject Focus〜Blur の12列は目的変数Pawpularityと単体では相関ゼロに近いことがEDA で判明（次項）

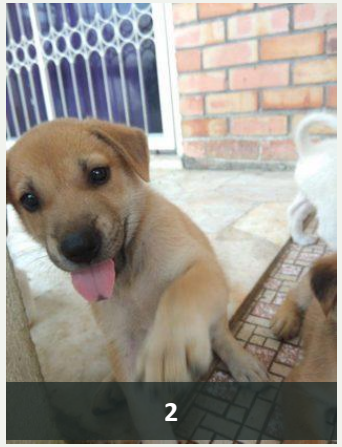
Pawpularity 分布の特徴

- 0付近・100付近に太い裾（fat tail）
- 分布は中心付近になだらかな山を持ちつつ両端に偏りが集中
- → 通常の回帰損失では境界付近の予測が発散しやすい
- → BCE損失（疑似最尤推定）採用の動機に直結

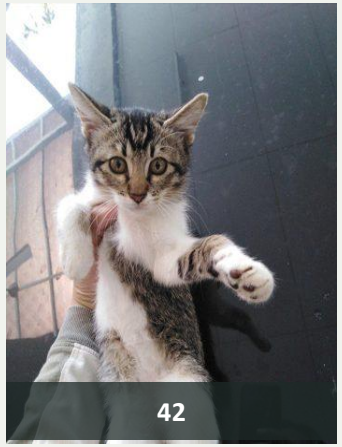


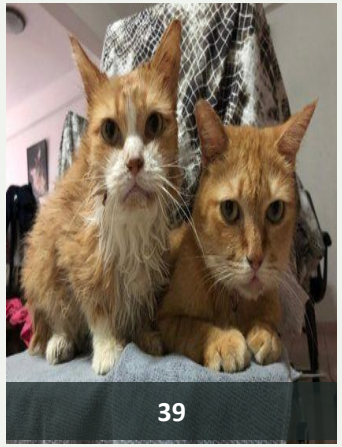
スコア帯ごとのサンプル画像

低スコア帯

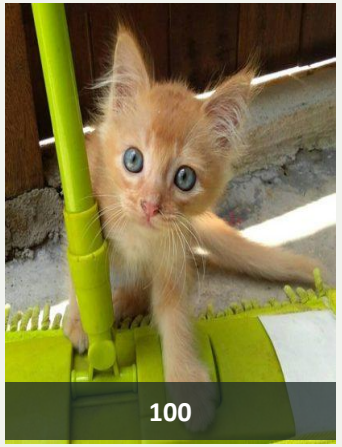


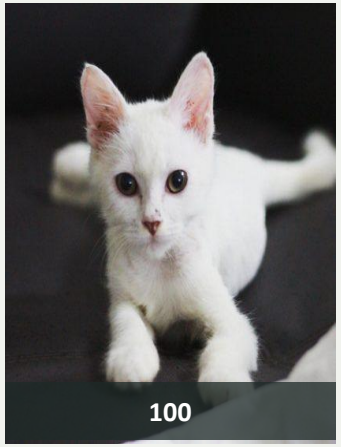
中スコア帯





高スコア帯

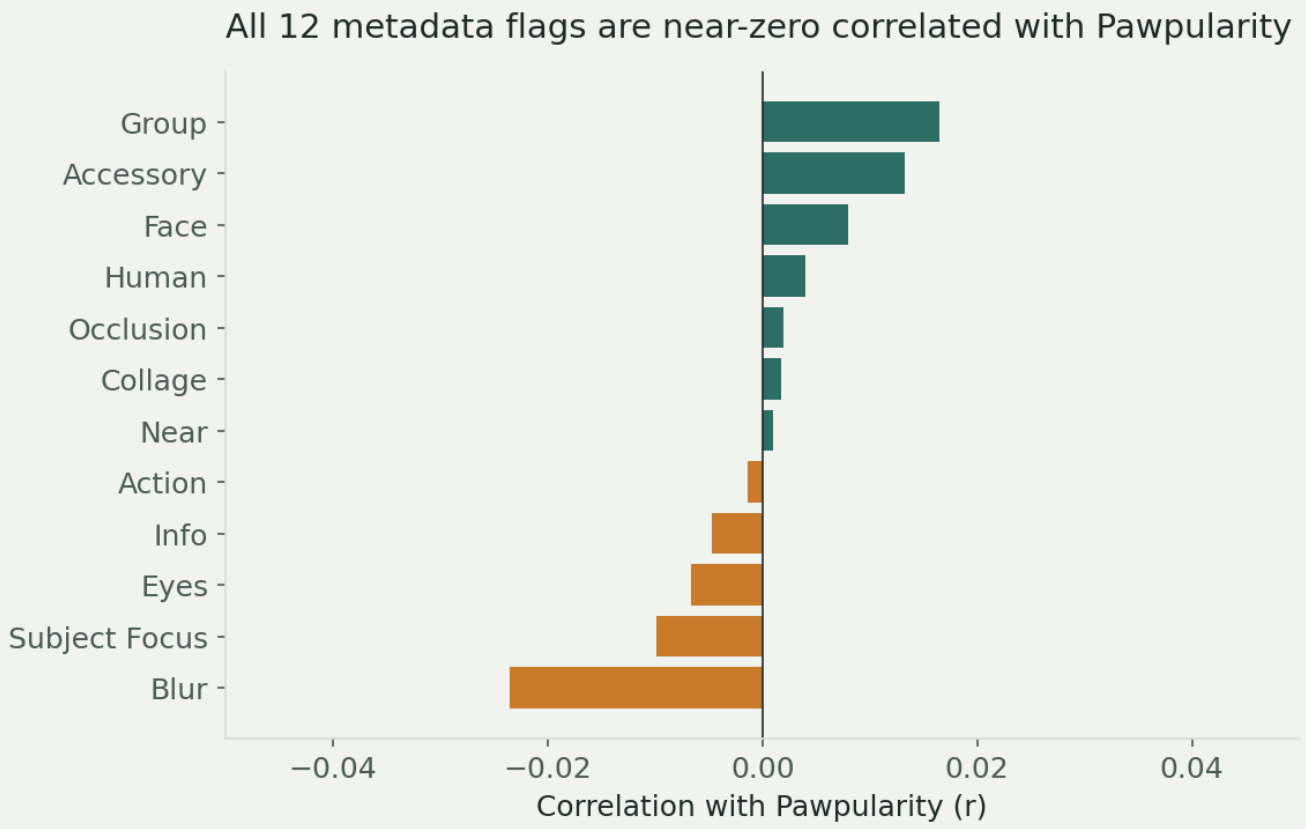




各スコア帯からランダムに抽出した実サンプル（左: Pawpularityが低い画像 / 中央: 中位 / 右: 高い画像）

メタデータとPawpularityの相関

- 全12種のメタデータフラグは目的変数とほぼ無相関（|r| はいずれも0.1未満）
- 単純な表形式特徴量だけでは説明力が乏しい
- → 画像そのものの視覚的特徴（構図・表情・色彩など）を捉える事前学習済みモデルの埋め込みが不可欠、という設計判断を裏付け



重複除去とリーク防止

pHash（知覚ハッシュ、閾値≤6bit）と EfficientNet-B0 のコサイン類似度（≥0.9）を組み合わせ、9,912枚中 38グループ（重複含む）を検出。同一 group_id は必ず同じfoldに収める。

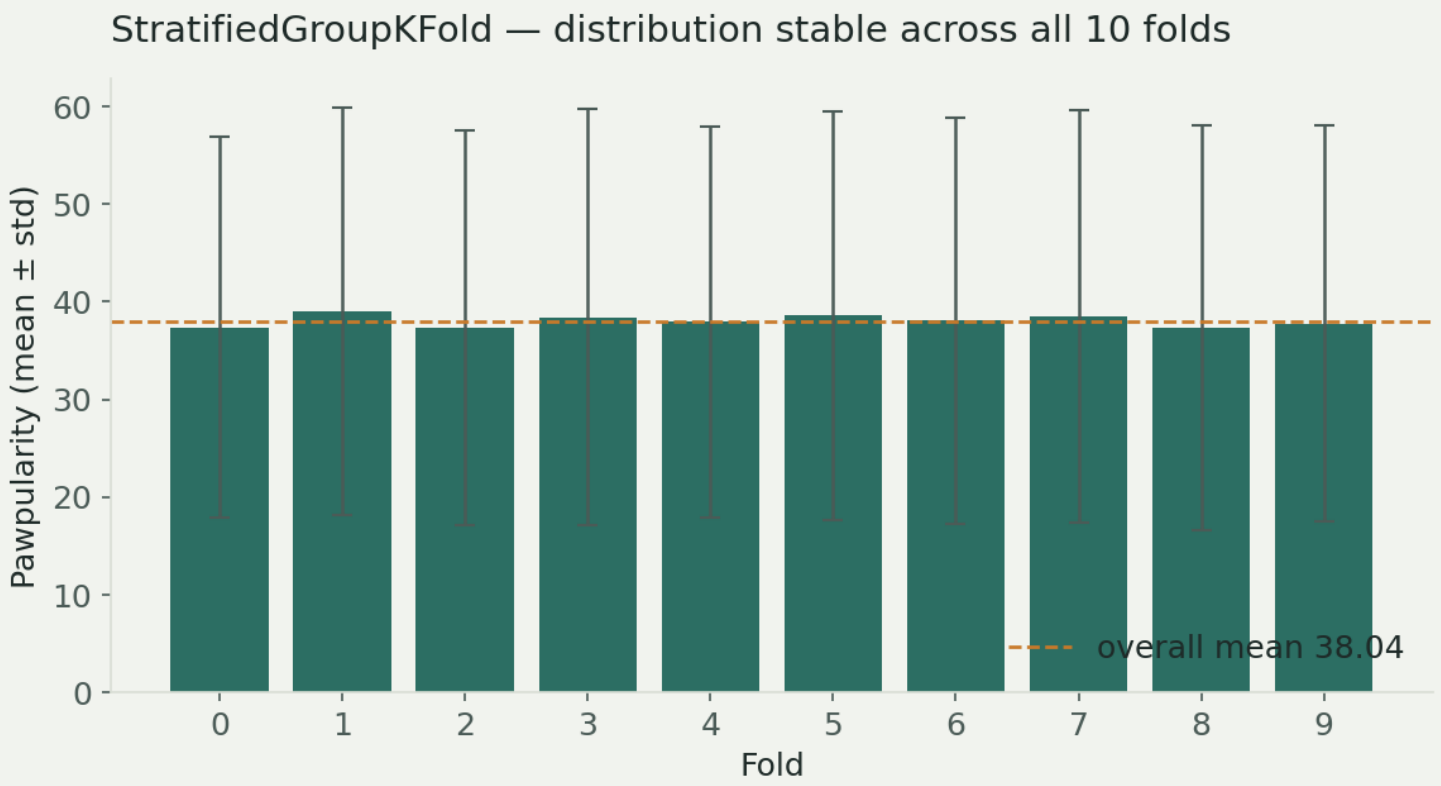
重複ペアの例

両者は近似的な同一画像 — 別foldに分かれると学習/検証間でリークが発生

StratifiedGroupKFold（10-fold）

- Sturges' rule（14 bins）でPawpularityを層化
- group = 重複検出のgroup_id
- seed = 42 で再現性を確保
- 10-fold: SVR CVと最終アンサンブル評価の両方で共通利用

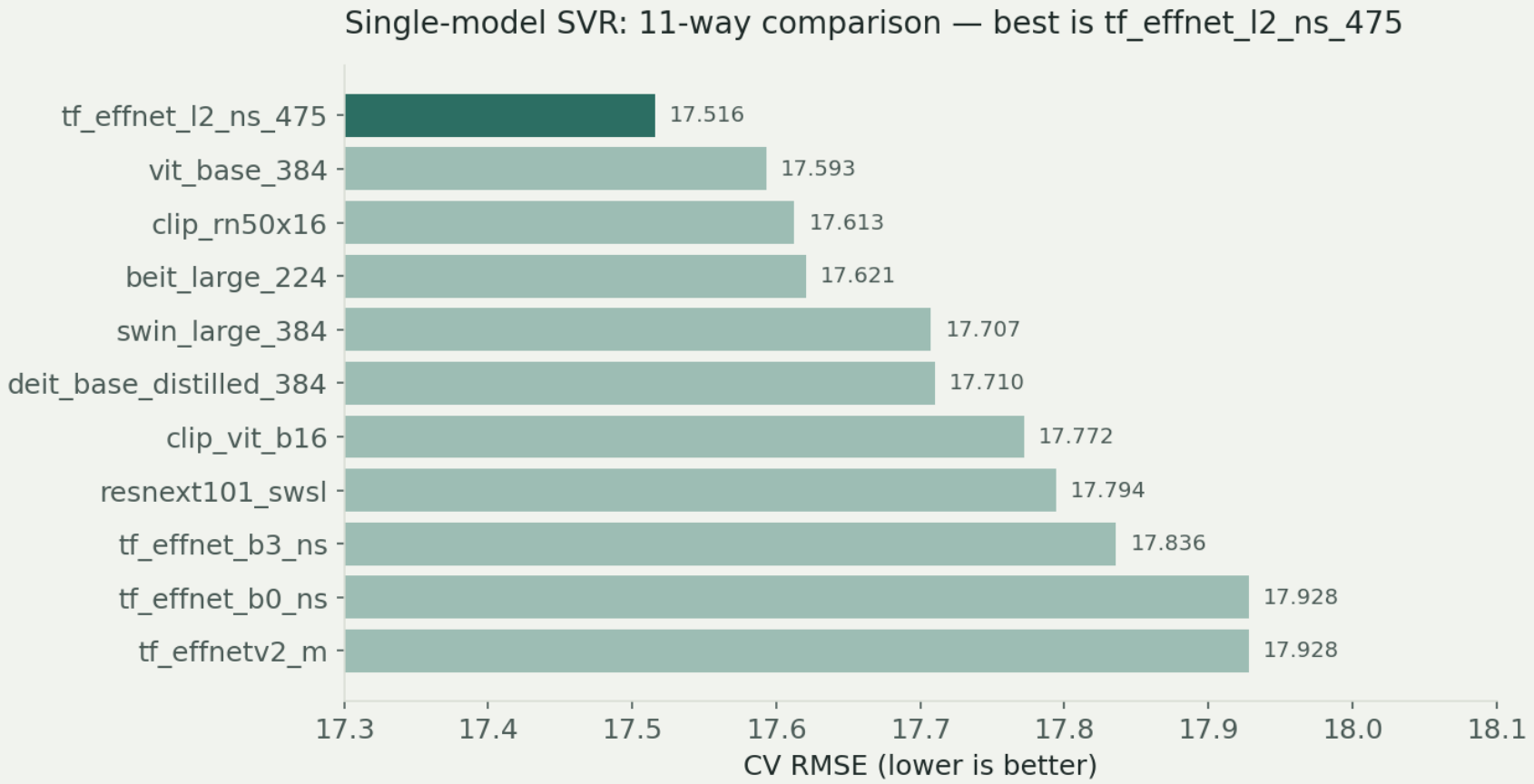
Fold間のスコア分布バランス



10 fold すべてでPawpularity分布がほぼ一致 — 層化が機能していることを確認

特徴抽出モデル単体のSVR CV比較

timm 7モデル + CLIP 2モデル（画像埋め込み）を抽出し、それぞれ単体で RAPIDS cuML SVR（RBFカーネル）10-fold CVを実施。単体最良は tf_efficientnet_l2_ns_475（RMSE 17.516）



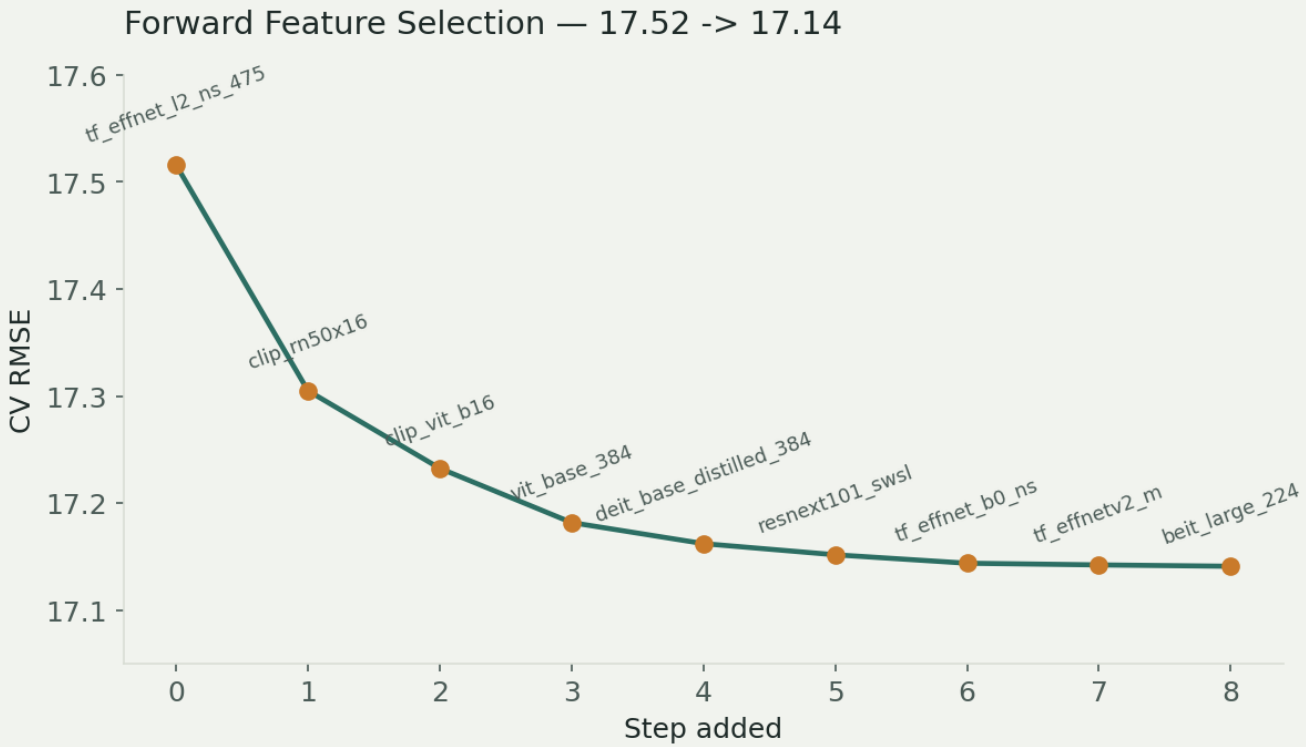
Forward Feature Selection（Hill Climbing）

11候補から、CV RMSEが改善する限りモデルを貪欲に追加する探索を実施。最終的に9モデルの組み合わせで単体最良から更に0.375改善。

17.516 → 17.141

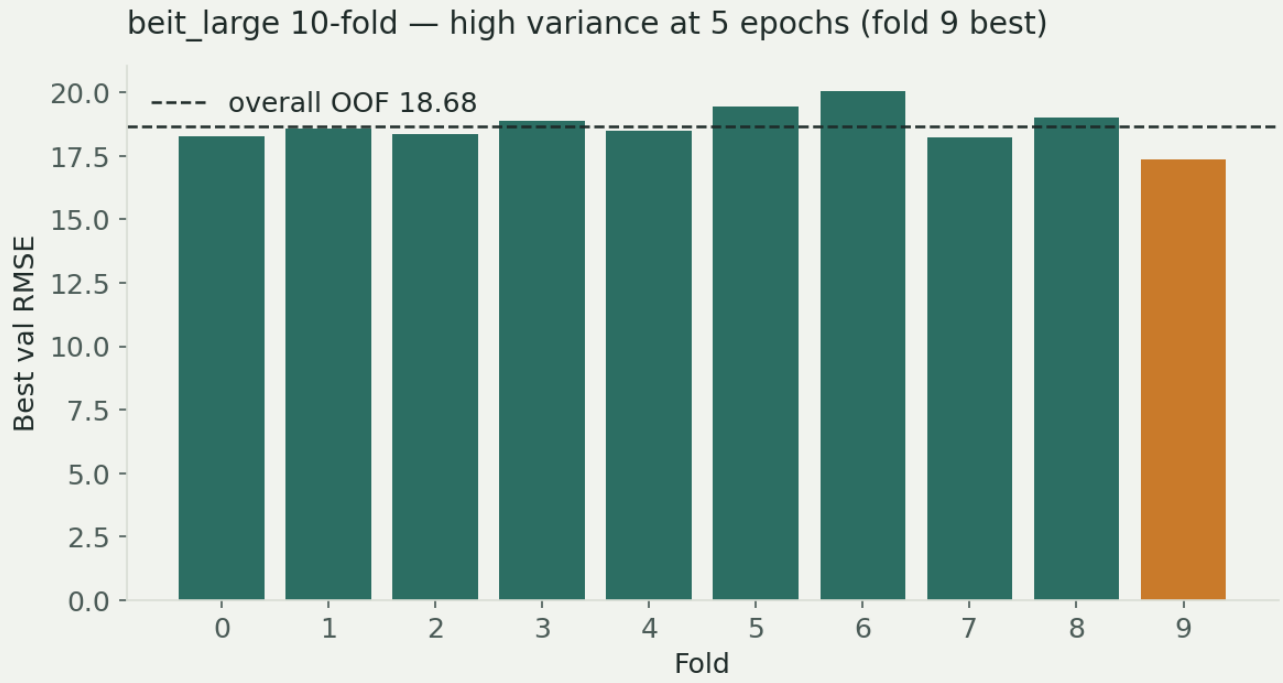
CV RMSE（単体最良 → 9モデル合成）

最終9モデル: tf_effnet_l2_ns_475 / clip_rn50x16 / clip_vit_b16 / vit_base_384 / deit_base_distilled_384 / resnext101_swsl / tf_effnet_b0_ns / tf_effnetv2_m / beit_large_224



beit_large_224 の BCE損失学習

- BCE損失を歪んだ回帰目的変数への疑似最尤推定として採用
- Bug 1: 目的変数=100付近でBCE境界発散 → 損失計算時のみ ±1e-3 でクリップして解消
- Bug 2: ヘッド初期化で予測が極端値に飽和 → 出力重みを小さく初期化し bias=logit(平均) から学習開始
- AMP + 勾配チェックポイント/累積で RTX3060 12GBに収める
- クラッシュ耐性: atomic checkpoint（一時ファイル→os.replace）で電源断/再起動から自動再開



10-fold中、Fold 9が最良（推論に採用）— fold間分散が大きく、単体ではSVRに劣る（CV RMSE 18.68）

SVR × DL ブレンド

重み w を 0〜1 で探索し、10-fold CV RMSEを最小化する組み合わせを採用

SVR 単体（9モデル）	17.1410
DL 単体（beit_large）	18.6791
ブレンド　w(SVR)=0.93	**17.1306**

最終 CV RMSE

17.1306

SVRが支配的、DLは補助的な役割 — 単体では劣るDLも異なる誤差傾向を持つため僅かにRMSEを押し下げる

UMAP と t-SNE、そして埋め込み手法

UMAP

局所構造を保持しつつ大域構造もある程度維持。計算が高速で、クラスタの形状や距離感が比較的解釈しやすい

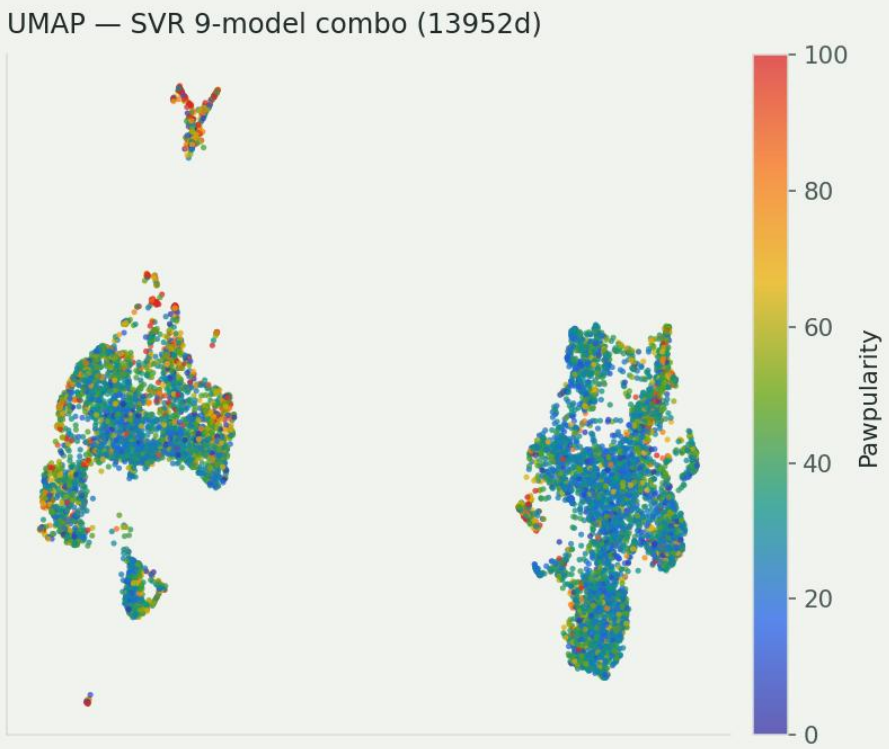
t-SNE

局所的な近傍関係の可視化に強いが、大域的な距離やクラスタ間の相対位置は歪みやすい。計算コストは相対的に高い

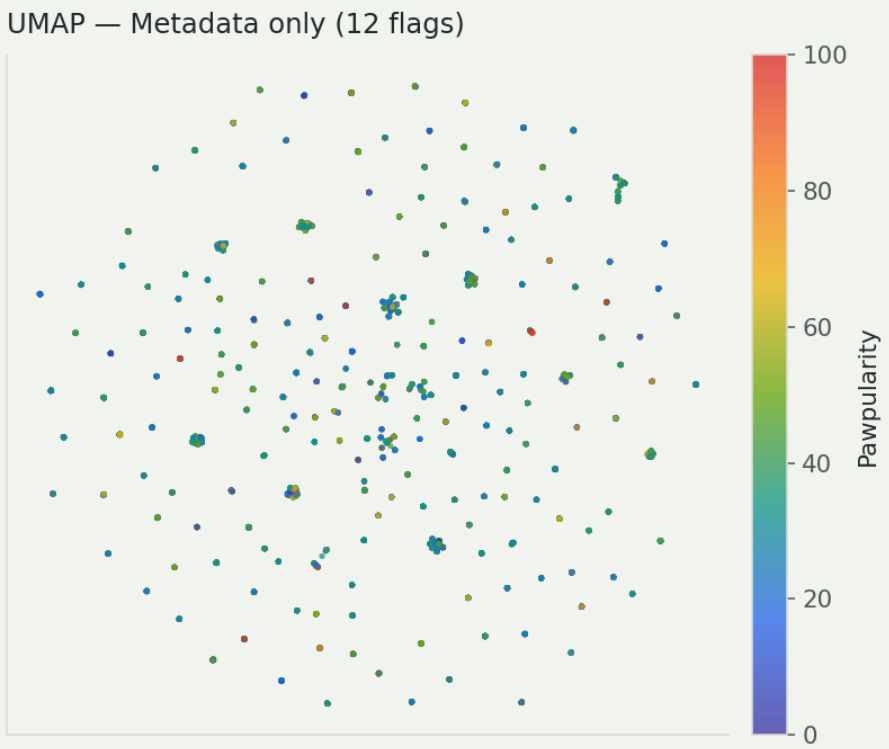
可視化に用いた特徴量（すべて学習パイプラインで抽出済みの埋め込み）

- **clip** CLIP ViT-B/16 画像埋め込み（512次元）
- **l2ns475** tf_efficientnet_l2_ns_475 特徴量（5,504次元）
- **svr9combo** SVR採用9モデル連結特徴量（13,952次元）
- **metadata** 主催者提供メタデータ12列

特徴空間の可視化：画像特徴量 vs メタデータ



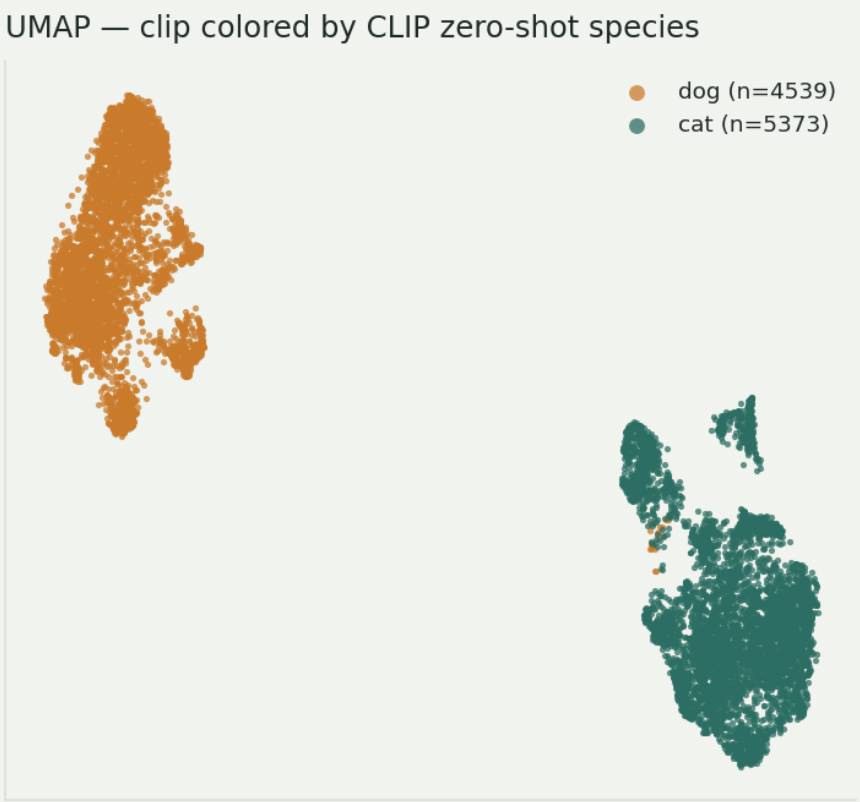
SVR採用9モデル特徴量（UMAP）



メタデータ12列（UMAP）

画像特徴量は明瞭な島状クラスタ構造を形成する一方、メタデータは構造に乏しい塊状 — Pawpularity予測には画像埋め込みが本質的に重要であることを可視化でも裏付け（色 = Pawpularityスコア）

2つの大きな島 — 犬と猫



犬・猫の2クラスタ（CLIP埋め込み・UMAP）

CLIP埋め込み空間（UMAP）で観察された2大クラスタについて、CLIPのゼロショット分類（テキストプロンプト vs 画像埋め込みのコサイン類似度）で検証

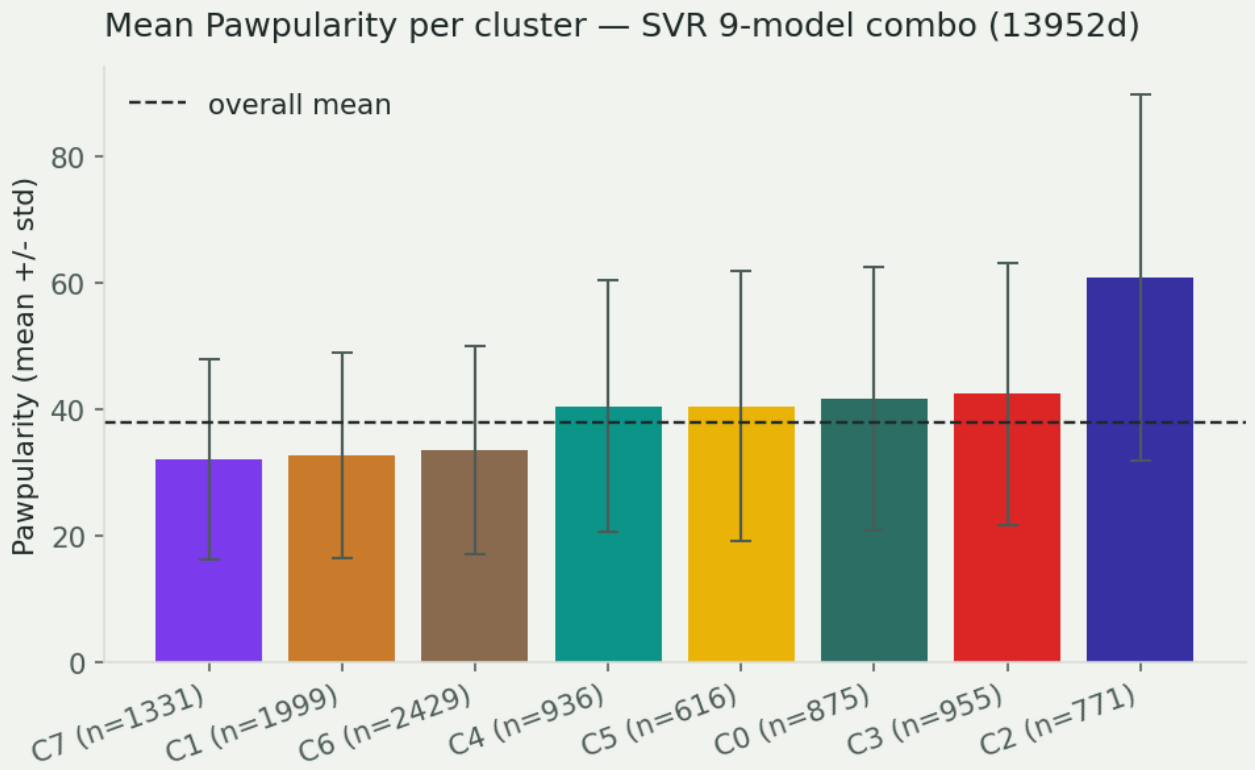
99.7%

UMAPの2クラスタ分割とCLIPゼロショット犬猫判定の一致率（9,885 / 9,912）

犬 4,539枚　　猫 5,373枚

学習済み画像埋め込みは、明示的なラベルなしに種の違いという意味的構造を自然に捉えていたことが分かる

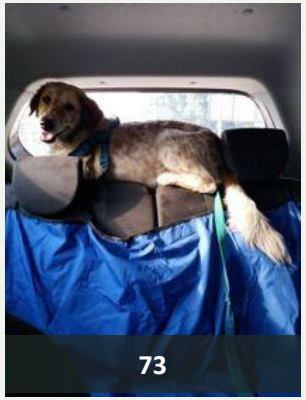
クラスタ別スコア傾向

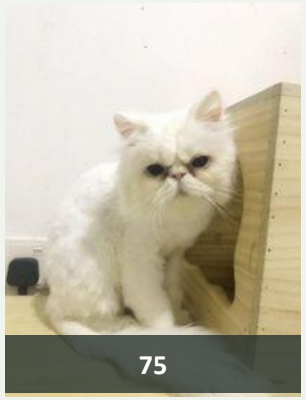


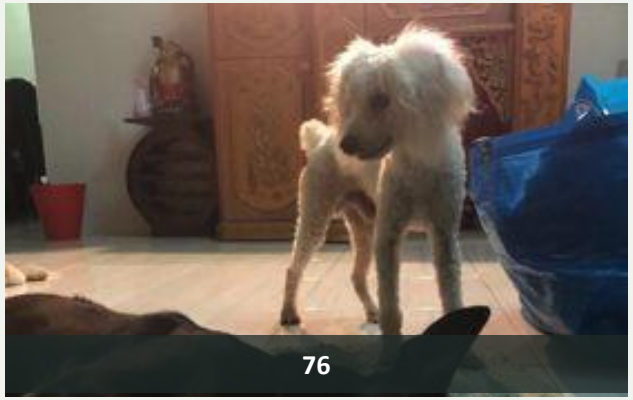
- 8クラスタの平均Pawpularityは32.2〜60.9と幅がある
- 多数派クラスタ（クラスタ1・6・7、計5,759枚）は平均32〜34で分布のボリュームゾーンと一致
- 少数派クラスタほど平均スコアが高くなる傾向 — 次ページでクラスタ2（最高スコア群）を詳細分析
- → これらの傾向は特徴量として既にSVR/DLモデルに暗黙的に織り込み済み

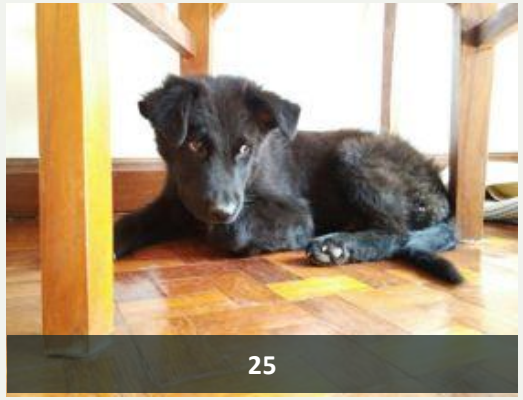
クラスタ2の特徴 — 高スコアのポイント

クラスタ2（n=771）のサンプル画像（実際のPawpularityスコア付き）









高スコアのポイント

8クラスタ中、最も平均スコアが高いクラスタ（mean≈61、他は32–43）が明確に分離した。サンプル画像を見ると単体被写体・中央構図・シンプルな背景に偏る傾向 — メタデータのNear/Face/Eyesフラグ単体では相関ゼロだったが、埋め込み空間ではこうした構図パターンが暗黙的に捉えられていた。

ローカル学習 → オフライン推論の全体像

1

ローカル学習（約5時間）

RTX3060 12GB / WSL2 cuML。特徴抽出 → SVR fit → beit_large 10-fold DL 学習

→

2

成果物パッケージング

重み・SVR joblib・オフラインwheelをKaggle Datasetとしてアップロード（計3本）

→

3

Kaggle Notebook（オフライン）

インターネット遮断環境で特徴再抽出 → SVR推論 → DL推論 → ブレンド

→

4

submission.csv 提出

採点用の非公開テストデータに対してSave & Run All（Commit）で自動実行・採点

鍵となる工夫: オフライン環境向けの pip wheel 事前ダウンロード / HF_HUB_OFFLINE 設定 / cuML のGPU カーネル非互換に備えたnumpyによる手動RBF推論実装

Kaggle提出で直面した4つの試練

1 P100 GPUの非互換

Kaggle標準GPU（Pascal, compute capability 6.0）でPyTorchのカーネルが“no kernel image” エラー。fp32化＋GPU→CPUフォールバックで対処

2 cuMLのサイレント失敗

同根本原因でcuML SVRのpredict()が例外を出さず全ゼロを返す。support_vectors_等から手動でnumpyのRBF決定関数を実装し回避

3 T4x2選択が反映されない

Session optionsでの選択がSave & Run Allに反映されない。T4x2の対話セッションを実際に起動・確認した上でそこからSave Versionして解決

4 オフライン依存の解決

Code Competitionはネット遮断。重み/wheel/SVRを3本のKaggle Datasetとして事前アップロードし、HF_HUB_OFFLINE等を設定して対応

FINAL RESULTS

最終提出結果

17.09524	17.94270	17.1306	0.035
PRIVATE SCORE	PUBLIC SCORE	LOCAL CV RMSE	PRIVATE とCVの差

Private ScoreとLocal CVの差はわずか0.035 — StratifiedGroupKFold + 重複除去による堅牢なCV設計が、未知データへの汎化性能を正確に予測できていたことを示す。

開発時間: DL学習だけで約5.5時間（10-fold本番実行）／ 試行錯誤・デバッグ含む累計約12.8時間

FURTHER IMPROVEMENTS

さらなる改良の余地

1. クラスタ別・種別（犬/猫）モデルの分離学習
2. 高スコア外れクラスタ（クラスタ2）に特化した特徴量設計
3. DLモデルの多様化（複数バックボーンでのアンサンブル強化）
4. 疑似ラベリングによるテストデータ活用
5. 画像前処理・拡張（クロップ/構図正規化）の強化

いずれのアイデアも、今回のEDA で確認された分布・クラスタ構造は既に画像特徴量経由でモデルに暗黙的に織り込まれているため、直接的な改善効果は限定的と推定される