Maching Learning Technique Final Project

Team:LHS

R07944050 賴以尊 R07944033洪軒治 R07944030黃聖智

1. **Preprocessing**
    a. Data loading :首先將三個 .npz檔分別 load成 numpy array格式
    b. Cross Validation :利用 sklearn package將 training data以 7 : 3比例切 training set和 validation set
    c. Parameter selection: 先利用運算速度較高的lightgbm尋找最佳參數，接下來利用準確度高的xgboost做運算
    d. 一開始為了確認 ligthgbm或 xgboost等GBDT類的演算法可用於分類此資料，先以下面hyper parameter做 POC:
        i. data subsample rate = 0.1,
        ii. feature subsampling by tree = 0.1,
        iii. learning_rate = 0.1,
        iv. early_stopping_rounds=5,
    e. 接下來逐步增加參數值已提高準確度

2. **Evaluation Metrics**
    a. Track 1 : Weighted Mean Absolute Error(WMAE)

$$\mathrm{WMAE}(Y,\hat{Y}) = \frac{1}{n_{samples}} \sum_{i=1}^{n_{samples}} \sum_{j=1}^{3} w_j |y_{ij} - \hat{y}_{ij}|$$

Mean Absolute Error是一個通用、不考慮方向性的evaluation criterion，但是由於此次的目標為一multi-target regression problem，Mean Absolute Error會對large target較為敏感，所以在這個track中，我們將每個target分別乘上一個weight去平衡各target的影響，WMAE越小model表現越好。

    b. Track 2 : Normalized Absolute Error(NAE)

$$\mathrm{WMAE}(Y,\hat{Y}) = \frac{1}{n_{samples}} \sum_{i=1}^{n_{samples}} \sum_{j=1}^{3} w_j |y_{ij} - \hat{y}_{ij}|$$

track 2則是利用NAE去控制scale in multi-target regression task，NAE越小越好

3. **Model**
    a. **eXtreme Gradient Boosting(XGBoost)**
        1. 利用python xgboost package配合 Scikit-learn中的 API實作
        2. Tuning parameters :

data subsample rate = 0.7,
feature subsampling by tree = 0.7,
learning_rate = 0.1,
early_stopping_rounds=5,
objective = 'reg:logistic' in penetration_rate & alpha predicting,
objective = 'reg:squarederror' in mesh_size predicting

3. Track 1:

Public score:

Tree num./ Tree depth	500	1000
8	48.821233	48.446821
13	50.019956	47.124994

Private score:

Tree num./ Tree depth	500	1000
8	48.220073	47.840273
13	49.592956	47.610962

4. Track 2:

Public score:

Tree num./ Tree depth	500	1000
8	0.751542	0.695409
13	0.871322	0.781616

Private score:

Tree num./ Tree depth	500	1000
8	0.585985	0.555188
13	0.667613	0.555597

5. 在 Tree num和 Tree depth都為此表最大的情況下(10000, 13)表現有比較好，但由於(1)差距並不大，(2)如果再加大 forest的深度和廣度會train非常久，(3)以Track 2的 public score來說深度似乎過頭了，所以就沒有在更進一步下去

b. **LightGBM**

   i. 利用python lightgmb package配合Scikit-learn API

   ii. Tuning parameters :

      1. data subsample rate = 0.7,
      2. feature subsampling by tree = 0.7,
      3. learning_rate = 0.1,
      4. early_stopping_rounds=5,
      5. objective = 'xentropy' in penetration_rate & alpha predicting,
      6. objective = 'mae' in mesh_size predicting

   iii. Track 1

Public score:

Tree num./ Tree depth	500	1000
8	50.387761	50.353205
13	49.056597	49.226585

Private score:

Tree num./ Tree depth	500	1000
8	51.273889	51.235551
13	50.012292	50.164468

   iv. Track 2

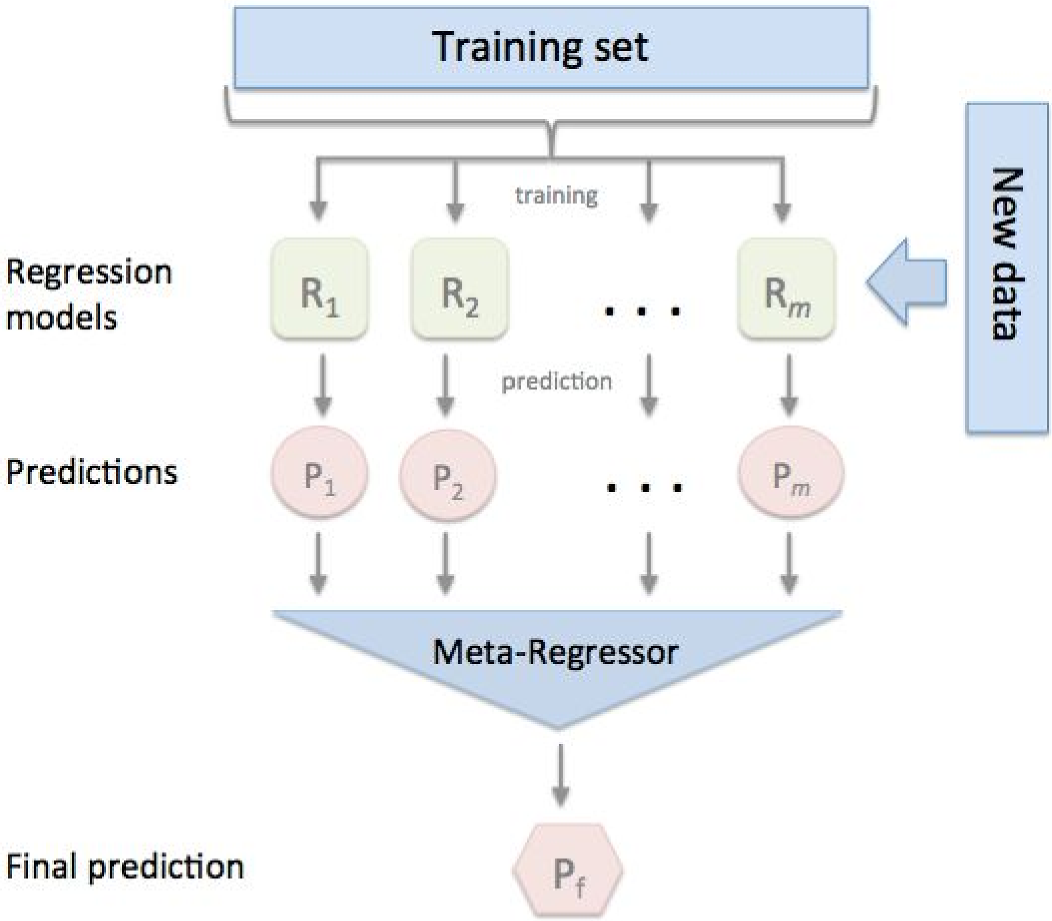
Public score:

Tree num./ Tree depth	500	1000
8	1.093496	1.098682
13	1.280168	1.116997

Private score:

Tree num./ Tree depth	500	1000
8	0.698276	0.701246
13	0.805989	0.775782

c. Stacking



i. 利用stacking 結合 {lightGBM, xgboost}
ii. Tuning parameters : 沿用前面參數
iii. meta regressor使用ridge regressor
iv. Track 1

Public score:

Tree num./ Tree depth	500	1000
8	45.902131	44.700456

Private score:

Tree num./ Tree depth	500	1000
8	47.512355	45.091522

v. Track 2

Public score:

Tree num./ Tree depth	500	1000
8	0.929634	0.789599

Private score:

Tree num./ Tree depth	500	1000
8	0.561642	0.526920

4. **Final Recommendation & Conclusion**
   a. **Model : Stacking with Xgboost and lightGBM**
   b. Parameter:
      1. data subsampling ratio : 0.7
      2. feature subsampling by tree : 0.7
      3. learning rate : 0.1
   c. **Score :**

Track num./Score	Track 1	Track 2
Public	45.902131	0.695409
Private	44.700456	0.526920

   d. **Pros :** Xgboost 的預測結果表現相當優秀，而 maching learning在調參上比起deep learning也更容易理解。此外，lightGBM則在運算效能上相當優異，以及較佳的GPU運算支援。最後為了提升準確度，將上述提到的兩個Model做Stacking，取得了最佳成績。
   e. **Cons :** 兩組最佳model的共通點為tree num都到達了1000棵，而Xgboost因為其GBDT的設計等因素，本身training就很久了，在size加大到1000之下，training time要花好幾個小時且model本身對gpu加速的支持並不好，沒有比較有顯著效果的加速方法。而Stacking則須將兩邊運算完後才能得出結果，因此為此三種方法中運算最久的方式。

5. **Work Loads:**
   a. 洪軒治 : lightGBM training, Stacking model training, 研究主要Learning架構, 報告製作
   b. 賴以尊： Xgboost training, Stacking model training, 報告製作
   c. 黃聖智 : Xgboost training