

Mario Filho

## Dia 1

Hoje criei os primeiros notebooks para a série de aulas sobre ensemble.

Decidi focar em stacking, já que a parte de média simples e ponderada já é mostrada no curso original.

Comecei por separar os dados entre treino de nível 0, treino de nível 1 e validação da stack.

Pensei em fazer com block-CV, mas acho que vai overfitar demais, talvez eu teste no final.

Esses dados são extremamente instáveis, então eu acho que terei problemas para fazer a stack funcionar melhor no teste, mas ainda assim é válido fazer a stack para estabilizar as previsões.

Criei os dois modelos que eu mais gosto de juntar: gbm e redes neurais.

Simples, sem muito tuning, só pra ver o que acontece.

Salvei todas as previsões e testei uma stack com previsões de todos os pontos explorados pela Bayes optimization durante o tuning.

Fiz isso para tentar reduzir o bias de ter escolhido os melhores modelos no mesmo dataset que estou usando para treinar.

Melhorou a validação com um Ridge (LinReg não funcionou bem). Lasso não convergiu.

Random Forest demora demais. ExtraTrees parece estável mas não chega ao score de validação da Ridge, talvez a Ridge esteja overfitando?

Na validação a Ridge bate os modelos originais, mas a ExtraTrees não.

Espiei o teste e a Ridge tá overfitando bravamente, Extratrees está razoavelmente estável.

Na prática não tem como saber, o teste é só mais uma amostra, eu ainda iria preferir os ensembles. Se tratando de dados extremamente ruidosos, a Extratrees parece boa.

Mario Filho

## Dia 2

Gerei mais alguns modelos para ensemble.

Geralmente GBM e NN são o núcleo do negócio, mas como KNN e modelos lineares ajudam em alguns casos, resolvi testar.

Não consegui rodar o KNN mas o modelo linear não melhorou.

Vou focar em GBM e outras NNs

Gerei modelos de LGBM em combos de grupos de features. Fiz uma seleção manual sequential forward na stack.

Parece ter funcionado, melhorou validação e teste (espiei novamente).

Fiz no esquema de melhorar acima da distribuição de adicionar coluna aleatória e também com epsilon para tentar deixar mais robusta.

Agora quero testar GBMs em subsets diferentes, por enquanto por grupos determinados previamente, mas em algum momento vou tentar gerar aleatoriamente.

Mesma coisa para features e parâmetros.

Preciso testar outras arquiteturas de NN também e gerar os modelos de grupo para features e subsets.

Mario Filho

## Dia 3

Decidi tunar uma rede neural com 3 camadas e dropout.

Não é tão fácil encontrar um equilíbrio entre fazer um modelo específico ficar bom e gerar modelos para o ensemble.

Aumentei para 300 epochs de treino e joguei para a nuvem.

Redes maiores tendem a precisar de mais tempo para convergir, os resultados individuais estão ruins, mas acho que o ruído dos dados colabora.

Estou acompanhando o tuning e parece que redes com mais unidades e quase nada de dropout funcionam melhor. Se este for o caso mesmo, vou rodar o tuning novamente com unidades acima de 500.

Nesse ponto seria interessante ver se a rede está under/overfitting. Em tese a decisão de arquitetura deve ser feita olhando se a rede consegue, pelo menos, overfitar.

Aí depois você tenta regularizar.

Tem gente que diz para fazer a maior rede possível e depois regularizar, mas na prática eu sempre precisei mudar a arquitetura além de regularizar.

Pelo menos as dezenas (talvez centenas) de previsões vão ficar disponíveis para testar no ensemble, então não é gasto computacional à toa.

Joguei um XGB pra treinar na GPU (ver se ele é mais rápido mesmo). Parece ser, e me deu resultados melhores que o LGBM nos feature groups!!! Resolvi fazer um geralzão também.

Nem tunei, estou usando os params padrões do numerai. Talvez seja o número de árvores? Isso nunca aconteceu comigo.

XGB bateu meu melhor LGBM de longe, devem ser os parâmetros, a diferença não deveria ser tão grande. LGBM tá com 100 árvores, XGB com 2k.

Vou abrir uma instância spot para poder pegar uma GPU mais potente, sagemaker tá muito caro :(

0.92 USD spot vs 4 USD no sagemaker, por hora.

Mario Filho

Dia 4

Esqueci de escrever, escrevo no dia 5.

Mario Filho

## Dia 5

Implementei um número absurdo de ideias desde a última vez que escrevi.

XGB com feature groups

Redes neurais com 3 hidden layers e dropout

XGB com row groups

XGB com Hypers manuais e Random

Tudo isso para gerar diversidade no modelo.

Também decidi testar umas formas mais malucas de treinar esse ensembles:

Treinei um XGBClassifier que prevê cada nível do target como uma classe e multiplica a probabilidade prevista pelo valor original. Saiu melhor do que eu pensava, por volta de 0.037 corr.

XGBRanker: em vez de tratar como regressão, e se pegarmos cada era e tratarmos como um problema de ranking? Não pós-processei as previsões, usei os próprios valores do ranking já que eles estão entre 0 e 1 também, mas é algo que pode ser melhorado.

Também deu por volta de 0.037 corr.

Testei a Random Forest do XGB, mas deu um resultado horrível tanto ajustando os params do booster quanto usando o objeto próprio da API.

Boosted Random Forest: usei uma dessas com Adaboost quando peguei o 6º lugar na comp do Facebook, então decidi testar essa versão do XGBoost.

É beeeem legal e bateu o melhor modelo individual (que também era XGBoost).

Em tese não é bem uma RF, já que a subsample não é feita com bootstrapping e o meu melhor resultado foi sem subsample.

Mas o legal é que, em vez de usar apenas uma árvore, usar várias em cada passo do boosting ajuda a reduzir a variância, que é absurda nesses dados.

Agora é testar tudo na stack amanhã :)