**Thinking1 XGBoost与GBDT的区别是什么？**

* GBDT是一种算法，而XGBoost则是GBDT的系统性工程实现。
* XGBoost优化了GBDT的求解过程：

1. 直接用XGBoost打分函数（包括了损失与正则两部分）去指导树的生成，避免了line search。而GBDT则是用平方（对数）损失指导树的生成，再做line search。
2. GBDT只用了一阶导数信息，而XGBoost则利用泰勒公式进行二阶展开。求解更快，还可以自定义损失函数。
3. XGBoost打分函数综合了loss和正则，避免了决策树容易过拟合的问题。

* XGBoost优化了分裂点搜索算法。不是简单地利用样本个数进行分位，而是以二阶导数值作为权重。
* 传统的GBDT每轮使用全部数据，XGBoost支持对数据进行采样。
* 传统的GBDT没有设计对缺失值处理，XGBoost能够学习出默认的节点分裂方向来处理缺失值。

**Thinking2 举一个你之前做过的预测例子（用的什么模型，解决什么问题，比如我用LR模型，对员工离职进行了预测，效果如何... 请分享到课程微信群中）**

员工离职率预测项目：（题目给了两个数据集：train.csv - 训练集和test.csv - 测试集）

* train数据集加载
* train数据预处理
* train dataset：分离特征train\_Attrition 和Target
* drop对模型没有作用的特征’user\_id’, 'EmployeeCount', 'EmployeeNumber', 'StandardHours', 'Over18'
* apply one-hot encoder to the train dataset for train\_Attrition.dtypes == 'object'
* 发现Target Variable: Attrition has imbalanced class problem：the percentage of Current Employees is 84.0% and the percentage of Ex-employees is: 16.0%.This needs to be addressed before implementing the Machine Learning algorithms.
* train数据集切分 - train\_test\_split

Since there is class imbalance (i.e. more employees with turnover=0 than turnover=1), I used stratify=y to maintain the same ratio as in the training dataset when splitting the dataset.

* train模型训练和预测

1. 尝试了baseline algorithms：Logistic Regression, Random Forest, SVM, KNN, Decision Tree Classifier, Gaussian NB.
2. Based on the ROC AUC comparison analysis, Logistic Regression and Random Forest show the highest mean AUC scores. I used these two algorithms for further analysis.
3. The fine-tuned Logistic Regression model(AUC score: 0.809) has a higher AUC score compared to the Random Forest Classifier(AUC score: 0.800).So I used Logistic Regression model on the test dataset.

* 把训练好的模型应用于测试集

1.加载测试集数据

1. 数据预处理
   * drop对模型没有作用的特征’user\_id’, 'EmployeeCount', 'EmployeeNumber', 'StandardHours', 'Over18'
   * apply trained one-hot encoder to the test dataset for object features

3.用predict\_proba预测员工离职的概率

**Thinking3 请你思考，在你的工作中，需要构建哪些特征（比如用户画像，item特征...），这些特征都包括哪些维度（鼓励分享到微信群中，进行交流）**

在做预测房屋价格SalePrice的项目中，我看了各个特征之间的相关关系（corr），得出以下结论：

* 'OverallQual', 'GrLivArea' and 'TotalBsmtSF' are strongly correlated with 'SalePrice'. So they will be useful in the prediction model.
* 'GarageCars' and 'GarageArea' are also some of the most strongly correlated variables. However, the number of cars that fit into the garage is a consequence of the garage area. 'GarageCars' and 'GarageArea' are like twin brothers and cannot be distinguished. Therefore, we just need one of these variables in our analysis (we can keep 'GarageCars' since its correlation with 'SalePrice' is higher).
* 'TotalBsmtSF' and '1stFloor' also seem to be twin brothers. We can keep 'TotalBsmtSF'.
* More than 15% of the data is missing in 'PoolQC', 'MiscFeature', 'Alley', and none of these features seem to be very important, since most of them are not aspects in which we think about when buying a house. So these features will be dropped.