



云顶数模

预测模型



主讲人：黄不南

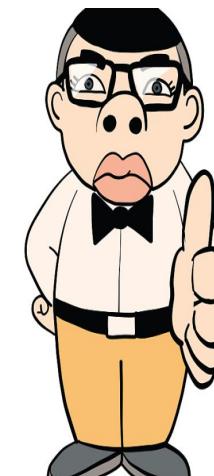


云顶数模



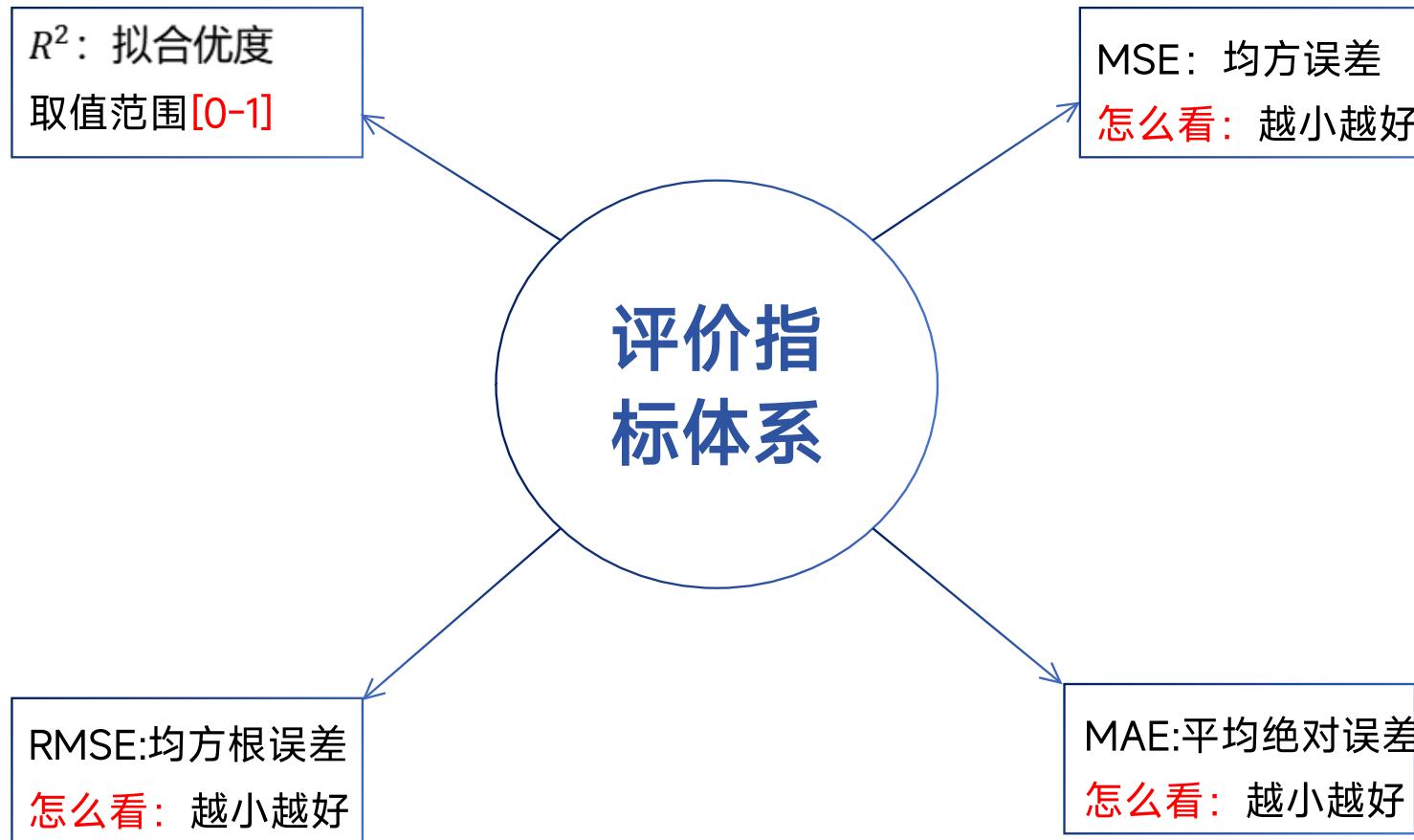
✓ 一句话概述机器学习模型

通过历史的数据来训练出一个模型，当有新的特征时，拿训练好的模型去预测



机器学习其实很简单！

01 机器学习回归问题入门必看



简单总结

越大越好: R^2

越小越好: RMSE
MSE
MAE



机器学习回归模型有哪些？

树模型

决策树

随机森林

梯度提升树

XGBoost

LightGBM

CatBoost

AdaBoost



线性模型

线性模型通过建立输入特征和目标值之间的线性关系进行预测。

线性回归

逻辑回归

岭回归

LASSO回归

一般我们了解这几种
就足够用了

其余模型

支持向量机

贝叶斯模型

神经网络

K近邻





不同模型应用场景划分--有监督模型

线性回归 (Linear Regression)

简单线性回归 (Simple Linear Regression) 一个自变量预测一个因变量

仅适用于线性关系，易受异常值影响

通过拟合一条直线，最小化预测值与实际值之间的误差平方和。

多元线性回归 (Multiple Linear Regression) 多个自变量预测一个因变量

多重共线性问题，模型解释性下降

通过线性组合多个自变量，预测因变量，最小化误差平方和。

非线性回归 (Nonlinear Regression)

多项式回归 (Polynomial Regression) 通过多项式函数拟合数据，捕捉非线性关系

容易过拟合，高次多项式计算复杂

使用多项式函数拟合数据，通过增加多项式阶数捕捉数据的非线性趋势。

支持向量回归 (Support Vector Regression, SVR) 利用支持向量机进行回归，适用于高维数据

对参数选择敏感，计算复杂度高

通过核函数映射数据至高维空间，寻找最大化间隔的回归平面。

决策树回归 (Decision Tree Regression) 使用决策树结构进行预测，适用于非线性数据

容易过拟合，模型不稳定

通过树状结构分割数据空间，基于特征进行条件判断预测目标值。

k-近邻回归 (k-Nearest Neighbors Regression, k-NN) 基于邻近样本的平均值进行预测，适用于非参数问题

计算复杂度高，受噪声影响大

根据距离选择最近的k个邻居，取其目标变量的平均值作为预测结果。

高斯过程回归 (Gaussian Process Regression, GPR) 适用于不确定性评估和复杂函数拟合

计算资源需求高，扩展性差

基于高斯过程的贝叶斯方法，通过协方差函数建模数据的平滑性和相关性。

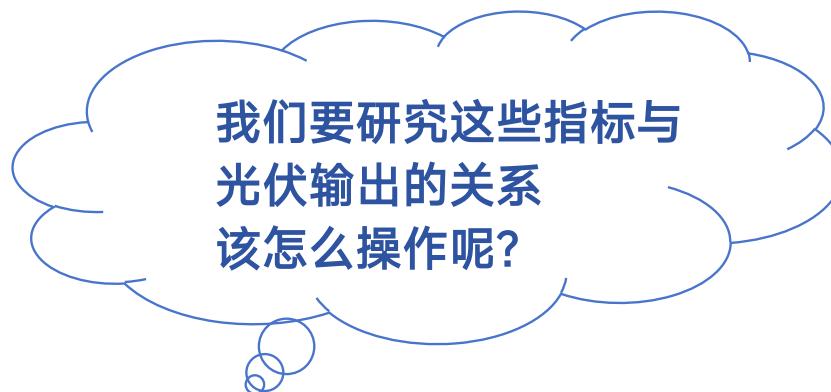


正则化回归 (Regularized Regression)	岭回归 (Ridge Regression)	通过L2正则化防止过拟合，适用于多重共线性问题	无法进行变量选择，仅缩小系数	在最小二乘法中加入L2正则化项，惩罚系数的平方和，减小模型复杂度。
	Lasso回归 (Lasso Regression)	通过L1正则化实现变量选择，适用于高维数据	当特征相关时，随机选择一个特征	在最小二乘法中加入L1正则化项，促使部分系数变为零，实现特征选择。
	弹性网络回归 (Elastic Net Regression)	结合L1和L2正则化，适用于高维数据和多重共线性	参数调节复杂，需要平衡L1与L2的权重	同时使用L1和L2正则化项，结合两者的优点，提升模型性能和稳定性。
集成回归方法 (Ensemble Regression Methods)	随机森林回归 (Random Forest Regression)	集成多个决策树进行预测，适用于高维数据和非线性关系	模型较大，训练时间长，解释性较差	通过构建多棵决策树并取平均值，减少过拟合，提高预测精度。
	梯度提升回归 (Gradient Boosting Regression)	通过逐步提升决策树性能，适用于复杂非线性关系	易过拟合，参数调节复杂	通过加法模型逐步构建树，每棵新树修正前一棵树的误差。
	AdaBoost回归 (AdaBoost Regression)	通过加权多个弱回归器，适用于提高模型精度	对噪声和异常值敏感，容易过拟合	迭代训练弱回归器，逐步增加错误样本的权重，提高整体模型性能。
深度学习 (Deep Learning)	XGBoost回归 (Extreme Gradient Boosting Regression)	高效的梯度提升实现，适用于大规模数据和高维特征	参数调节复杂，资源需求高	优化的梯度提升框架，采用正则化和并行计算提升性能和防止过拟合。
	LightGBM回归 (Light Gradient Boosting Machine Regression)	适用于大数据和高维特征，速度快，内存占用低	对小数据集可能效果不佳，参数调节需谨慎	基于梯度提升的高效框架，采用直方图算法和叶子优先策略加速训练。



小案例分享

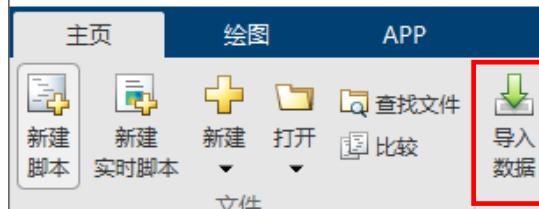
有点意思！说下去



B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
特征1	特征2	特征3	特征4	特征5	特征6	特征7	特征8	特征9	特征10	特征11	特征12	特征13	特征14	特征15	特征16	特征17	光伏输出
256.06	0.41	344.82	13.41	11.48	8.68	0.58	-9.89	54.33	162	-88.37	914.02	0.01	0	0.01	255.76	80.32	-0.20843
255.98	0.41	345.01	13.46	11.53	8.71	1.56	-9.85	54.48	162	-87.88	914.02	0	0	0	255.69	80.26	-0.20962
255.91	0.41	345.08	13.49	11.56	8.74	3.33	-9.83	54.52	163	-87.32	914	0.01	0	0.01	255.62	80.2	-0.20725
255.84	0.42	345.14	13.52	11.6	8.77	5.09	-9.8	54.57	164	-86.76	913.99	0.01	0	0.01	255.55	80.15	-0.20843
255.78	0.42	345.21	13.55	11.63	8.8	6.86	-9.78	54.61	165	-86.2	913.97	0.02	0	0.02	255.48	80.09	-0.20606
255.71	0.42	345.25	13.54	11.63	8.79	8.34	-9.73	54.39	164	-85.48	913.97	0.02	0	0.02	255.42	80.04	-0.20725
255.65	0.41	345.27	13.48	11.59	8.75	9.55	-9.65	53.9	164	-84.61	913.99	0.01	0	0.01	255.36	80	-0.2108
255.59	0.41	345.29	13.42	11.55	8.72	10.76	-9.57	53.42	162	-83.74	914.01	0.01	0	0.01	255.31	79.96	-0.20606
255.53	0.4	345.31	13.36	11.51	8.68	11.97	-9.49	52.93	161	-82.87	914.03	0	0	0	255.25	79.92	-0.20133
255.47	0.4	345.17	13.25	11.44	8.62	13.45	-9.37	52.18	162	-82.06	914.02	0.02	0	0.02	255.19	79.89	-0.20488
255.42	0.39	344.87	13.08	11.34	8.53	15.2	-9.2	51.17	163	-81.33	913.97	0.05	0	0.05	255.14	79.87	-0.20725
255.36	0.38	344.58	12.92	11.23	8.45	16.95	-9.04	50.15	164	-80.59	913.92	0.08	0	0.08	255.09	79.84	-0.2108
255.31	0.37	344.28	12.75	11.13	8.36	18.7	-8.87	49.14	165	-79.86	913.87	0.11	0	0.11	255.04	79.82	-0.20962
255.25	0.37	344.02	12.56	11.03	8.28	19.18	-8.68	48.19	165	-79.36	913.84	0.11	0	0.11	254.98	79.8	-0.20843
255.19	0.36	343.81	12.34	10.93	8.19	18.41	-8.46	47.3	164	-79.09	913.82	0.08	0	0.08	254.93	79.79	-0.20725
255.13	0.35	343.59	12.11	10.83	8.1	17.64	-8.23	46.41	163	-78.83	913.81	0.05	0	0.05	254.88	79.78	-0.2108
255.07	0.34	343.38	11.89	10.73	8.01	16.87	-8.01	45.52	162	-78.56	913.79	0.02	0	0.02	254.83	79.77	-0.20962
255.01	0.34	343.23	11.62	10.62	7.92	15.05	-7.75	44.67	16	-78.45	913.76	0	0	0	254.77	79.76	-0.20725
254.95	0.33	343.16	11.3	10.52	7.83	12.18	-7.44	43.86	158	-78.49	913.72	0	0	0	254.7	79.76	-0.20843
254.88	0.32	343.08	10.99	10.41	7.74	9.31	-7.13	43.06	157	-78.53	913.69	0.01	0	0.01	254.64	79.77	-0.20725
254.82	0.3	343.01	10.67	10.3	7.65	6.44	-6.82	42.24	155	-78.57	913.65	0.01	0	0.01	254.57	79.77	-0.20843
254.76	0.3	342.98	10.33	10.19	7.55	4.48	-6.49	41.6	153	-78.45	913.62	0.01	0	0.01	254.51	79.77	-0.20962



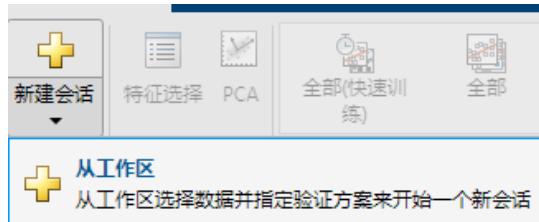
第一步：导入数据集



第二步：找到APP，打开回归学习器



第三步：加载数据集



第四步：选择你的因变量 (Y)

The dialog box shows the following settings:

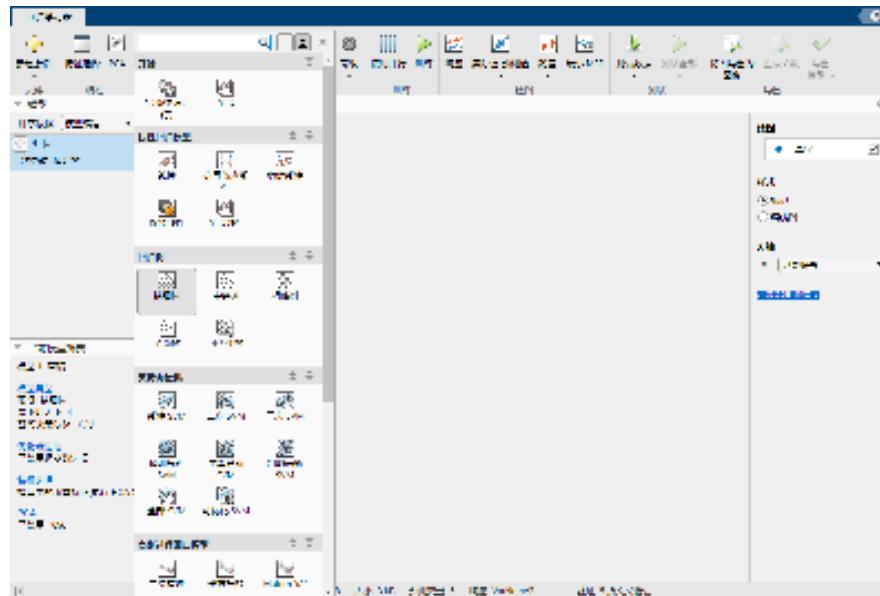
- Data Set**:
 - Data set variable: data (18x7 table)
 - Response:
 - From data set variable (VarName2)
 - From workspace (VarName2)
 - Predictor variables:
 - VarName1, VarName2, VarName3, VarName4, VarName5, VarName6, VarName7VarName5 is selected.
- Validation**:
 - Cross-validation (selected):
 - Divide data set into k folds (5 folds)
 - Leave-one-out validation (for large datasets):
 - 留出百分比: 25%
 - Re-substitution validation (for all data):
 - 不采用任何防止过拟合的措施。App 将所有数据用于训练和验证。
- How to Prepare Data**:
 - 如何准备数据
 - 进一步了解验证
- Buttons**:
 - 开始会话 (Start Session)
 - 取消 (Cancel)



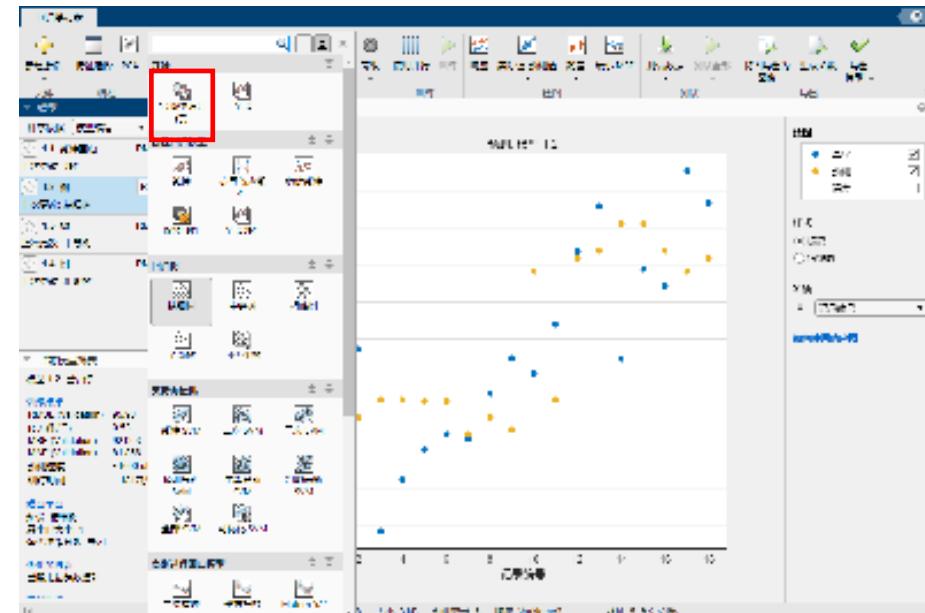


恭喜你！前面几步已经顺利完成 接下来准备进行多种机器学习模型的预测吧！

第一步：熟悉有哪些模型



第二步：选择预测所有模型



预测完成之后，点击导出模型，对话框内把表T改为自己的表名称即可进行预测



举个例子---论文写作的时候怎么写?

以线性回归模型举例

4.6.2 线性回归模型

步骤一：建立模型

模型 1——线性回归(Linear Regression)

本文采用的线性回归模型是假设目标值与特征变量线性相关的基础上，以误差均方函数为损失函数，并令损失函数最小以确定参数的回归方法。

模型 2——岭回归(Ridge Regression)

岭回归在损失函数中增加了二阶正则项的最小二乘，也叫 L2 范数，具有降维作用的同时，它还以限制模型参数对异常样本的匹配程度，处理相关性高的数据集，进而提高模型面对多数正常样本的拟合精度。我组通过 RidgeCV 针对正则化强度 alpha 进行调参，得出在 alpha 为 14 时可以达到较好的拟合效果。

模型 3——Lasso 回归(Lasso Regression)

Lasso 回归类似于上文提到的岭回归，也通过构造惩罚函数来处理特征变量的共线性问题。但是，相较于岭回归，Lasso 回归可以通过使惩罚项由 L2 范数变为 L1 范数将相对不显著的特征变量系数压缩为 0，达到剔除变量的目的；而岭回归仅仅只在一定程度上压缩特征变量系数且会保留回归模型的全部变量。

步骤二：模型评估

根据上文模型准备中所详细阐述的评测标准和调整决定系数 R^2 ，本文对线性回归模型、岭回归模型、Lasso 回归模型进行评估，其结果如下表所示：

模型	线性回归模型	岭回归模型	Lasso 回归模型
调整决定系数 R^2	0.76108	0.76121	0.75069
平均相对误差	0.52198	0.52298	0.53002
5%误差准确率	0.13317	0.13317	0.13317
评测标准	0.20214	0.20194	0.20214

由表可知，三种线性回归模型的调整决定系数 R^2 均小于 0.8，这意味着模型的拟合优度表现一般；平均相对误差均在 52% 左右，表现不佳；而 5% 误差准确率基本在 13% 左右，相对较高；除此之外，模型的测评标准相对较低，均在 0.2 左右。因为，本文暂不采用线性回归模型。

步骤

线性回归是个大框架，里面包含多种子模型，用到哪个就介绍哪个模型的原理+公式

随后展示不同的子模型的评估性能（交叉对比）选择出一个最好的子模型

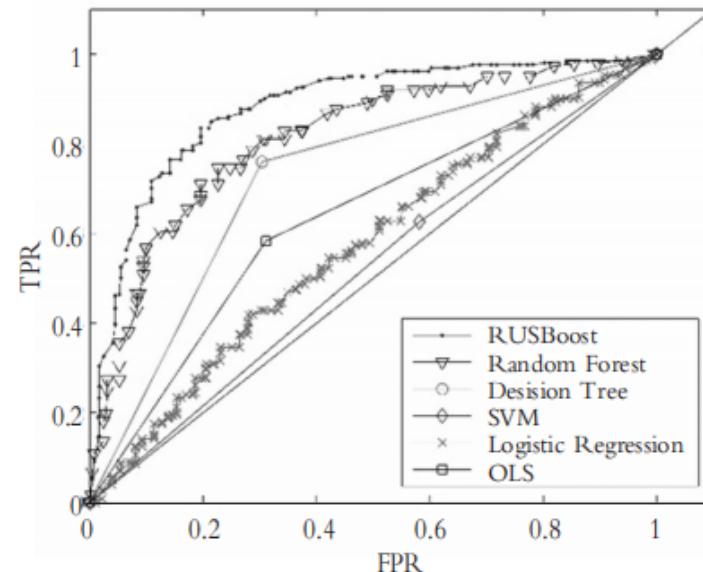
最后来上一段总结即可

02 机器学习分类问题入门必看



分类问题

和回归问题的操作步骤几乎一样，不同的是因变量从定量数据变成了定性数据



ROC曲线图

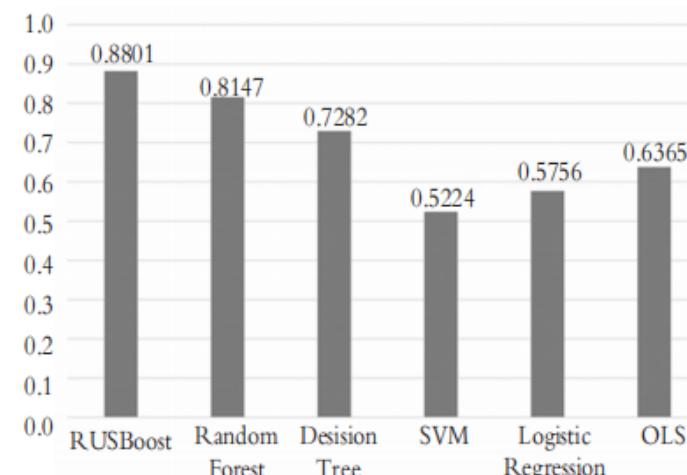


图 7 不同模型的AUC值

简单来说 分类问题我们最想得到的就是这两张图



评价指标介绍

- **分类报告。** 分类报告图示可以直观得到模型各项参数，包括每一类别的精确率（Precision），召回率（Recall），F1 分数值（F1-Score）。对于这三项值，其计算公式如下：

– 精确率

$$\text{Precision} = \frac{TP}{TP + FP} \quad (30)$$

– 召回率

$$\text{Recall} = \frac{TP}{TP + FN} \quad (31)$$

– F1 分数值

$$F1 = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} = \frac{TP}{TP + \frac{1}{2}(FP + FN)} \quad (32)$$

- **ROC/AUC 曲线。** 在分析特征曲线及曲线下面积（Receiver Operating Characteristic/Area Under the Curve, ROC/AUC）图之前，我们需要了解模型的相关参数，定义如下：

– 灵敏度 (Sensitivity)。灵敏度又被称为真阳性率，即 TP 率，定义为：

$$\text{Sensitivity} = TPR = \frac{TP}{TP + FN} \quad (34)$$

– 特异性 (Specificity)。特异性又被称为真阴性率，即 TN 率，定义为：

$$\text{Specificity} = TNR = \frac{TN}{TN + FP} \quad (35)$$

– 1-Specificity。称为假阳性率 (False Positive Rate, FPR)，定义为：

$$FPR = 1 - \text{Specificity} = \frac{FP}{FP + TN} \quad (36)$$

– 1-Sensitivity。称为假阴性率 (False Negative Rate, FNR)，定义为：

$$FNR = 1 - \text{Sensitivity} = \frac{FN}{FN + TP} \quad (37)$$



不同分类模型应用场景

逻辑回归 (Logistic Regression) 二分类问题, 适用于线性可分数据 线性判别分析 寻找最佳线性分隔 (Linear Discriminant Analysis, LDA) 超平面, 适用于有类别标签的多分类问题	线性决策边界, 无法捕捉复杂非线性关系 假设特征服从正态分布, 类别协方差相同 朴素假设特征条件独立, 可能不符合实际	使用sigmoid函数将线性组合映射为概率, 用于二分类预测。 最大化类间方差与类内方差的比率, 找到最能区分类别的投影方向。 基于贝叶斯定理和特征条件独立假设, 计算各类别的后验概率。
线性分类模型 (Linear Classification Models) 朴素贝叶斯分类器 (Naive Bayes Classifier) 支持向量机 (Support Vector Machine, SVM)	文本分类、垃圾邮件检测等, 适用于高维特征 二分类和多分类问题, 适用于高维数据和非线性可分问题	对参数和核函数敏感, 计算复杂度高 通过核函数映射数据至高维空间, 寻找最大间隔的决策边界。
k-近邻分类 (k-Nearest Neighbors Classification, k-NN) 基于邻近样本进行分类, 适用于多分类和非参数问题	计算复杂度高, 受噪声影响大	根据距离选择最近的k个邻居, 采用多数投票决定类别。

支持向量机 (Support Vector Machine, SVM) 二分类和多分类问题, 适用于高维数据和非线性可分问题	对参数和核函数敏感, 计算复杂度高	通过核函数映射数据至高维空间, 寻找最大间隔的决策边界。
神经网络 (Neural Networks) 复杂的非线性关系, 适用于图像、语音、文本等高维数据	训练时间长, 需大量数据, 易过拟合	通过多层网络结构和激活函数, 学习数据的复杂非线性模式。
决策树分类 (Decision Tree Classification) 基于特征进行树状决策, 适用于多分类问题	容易过拟合, 模型不稳定	通过树状结构分割数据空间, 基于特征进行条件判断预测类别。
非线性分类模型 (Nonlinear Classification Models) 随机森林分类 (Random Forest Classification)	集成多个决策树进行预测, 适用于高维数据和多分类问题	模型较大, 训练时间长, 解释性较差
梯度提升分类 (Gradient Boosting Classification)	通过逐步提升决策树性能, 适用于复杂非线性关系和多分类问题	通过加法模型逐步构建树, 每棵新树修正前一棵树的误差。
k-近邻分类 (k-Nearest Neighbors Classification, k-NN) 基于邻近样本进行分类, 适用于多分类和非参数问题	基于邻近样本进行分类, 适用于多分类和非参数问题	计算复杂度高, 受噪声影响大
多层感知机 (Multilayer Perceptron, MLP) 复杂的非线性关系, 适用于多分类和回归问题	训练时间长, 需大量数据, 易过拟合	通过多层神经网络结构和反向传播算法, 学习数据的复杂非线性模式。
极限学习机 (Extreme Learning Machine, ELM) 快速训练的单隐层前馈神经网络, 适用于分类和回归问题	需要适当选择激活函数, 参数调节较少	随机设置隐藏层权重, 快速计算输出层权重, 实现高效学习。

我个人比较喜欢用的模型

决策树与集成方法 (Decision Trees and Ensemble Methods)	决策树分类 (Decision Tree Classification)	基于特征进行树状决策，适用于多分类问题	容易过拟合，模型不稳定	通过树状结构分割数据空间，基于特征进行条件判断预测类别。
	随机森林分类 (Random Forest Classification)	集成多个决策树进行预测，适用于高维数据和多分类问题	模型较大，训练时间长，解释性较差	通过构建多棵决策树并采用多数投票，减少过拟合，提高分类精度。
	梯度提升分类 (Gradient Boosting Classification)	通过逐步提升决策树性能，适用于复杂非线性关系和多分类问题	易过拟合，参数调节复杂	通过加法模型逐步构建树，每棵新树修正前一棵树的误差。
	AdaBoost分类 (AdaBoost Classification)	通过加权多个弱分类器，适用于提升模型精度	对噪声和异常值敏感，容易过拟合	迭代训练弱分类器，逐步增加错误样本的权重，提高整体模型性能。
	XGBoost分类 (Extreme Gradient Boosting Classification)	高效的梯度提升实现，适用于大规模数据和高维特征	参数调节复杂，资源需求高	优化的梯度提升框架，采用正则化和并行计算提升性能和防止过拟合。
	LightGBM分类 (Light Gradient Boosting Machine Classification)	适用于大数据和高维特征，速度快，内存占用低	对小数据集可能效果不佳，参数调节需谨慎	基于梯度提升的高效框架，采用直方图算法和叶子优先策略加速训练。
	Extra Trees分类 (Extremely Randomized Trees Classification)	类似随机森林，适用于高维数据和多分类问题	模型较大，解释性较差	构建多棵完全随机化的决策树，通过多数投票进行分类，减少过拟合。
	极限学习机分类 (Extreme Learning Machine Classification)	快速训练的单隐层前馈神经网络，适用于分类问题	需要适当选择激活函数，参数调节较少	随机设置隐藏层权重，快速计算输出层权重，实现



The screenshot shows the MATLAB desktop interface with several tabs open. The 'Classification' tab is active, displaying a table of 21 entries. Each entry includes a checkbox, a model name (e.g., 'RF', 'SVM', 'KNN'), its type ('Model' or 'Predictor'), its status ('已训练' or '未训练'), accuracy percentage, and the number of samples. The table is as follows:

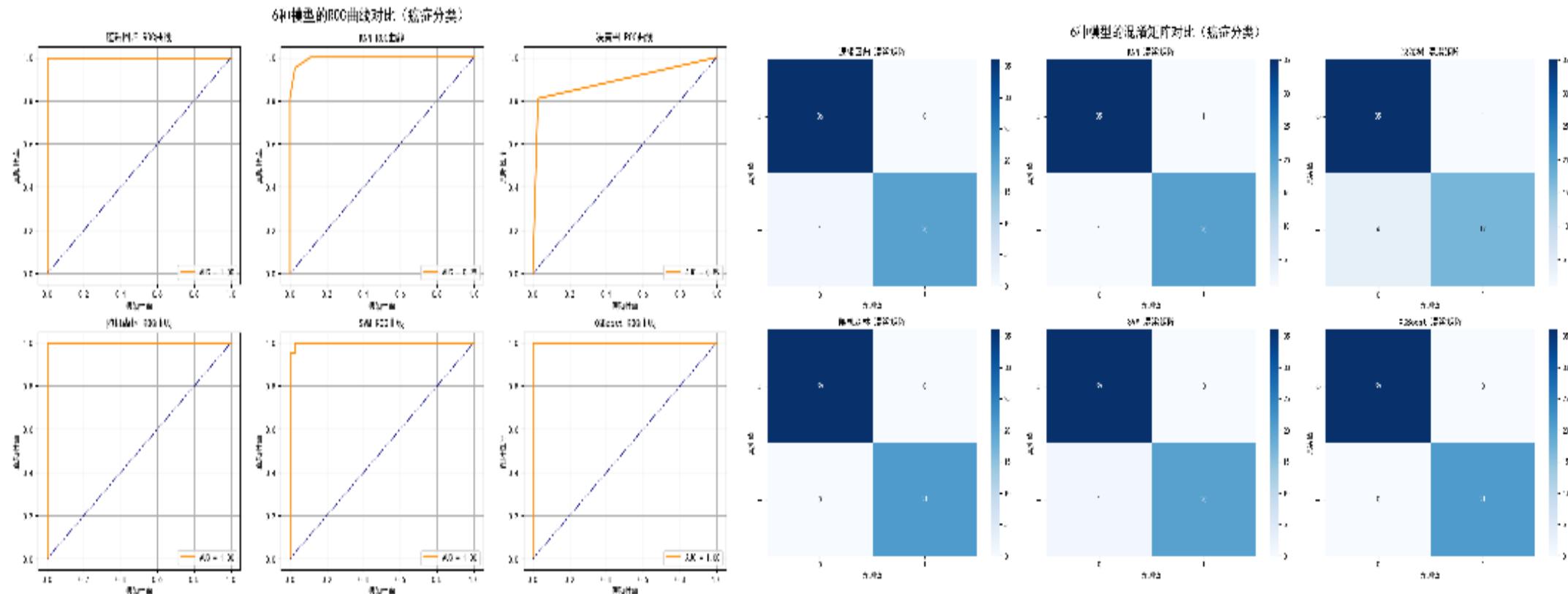
训练集	模型名称	模型类型	状态	准确率(%)	样例数
1	RF	Model	已训练	93.1%	30
2	RF	Model	已训练	93.1%	30
3	RF	Model	已训练	93.1%	30
4	RF	Model	已训练	93.1%	30
5	RF	Model	已训练	92.0%	30
6	RF	Model	已训练	93.1%	30
7	RF	Model	已训练	93.1%	30
8	RF	Model	已训练	93.1%	30
9	RF	Model	已训练	93.1%	30
10	RF	Model	已训练	93.1%	30
11	RF	Model	已训练	93.1%	30
12	RF	Model	已训练	93.1%	30
13	RF	Model	已训练	93.1%	30
14	RF	Model	已训练	93.1%	30
15	RF	Model	已训练	93.1%	30
16	SVM	Model	已训练	93.7%	30
17	SVM	Model	已训练	93.7%	30
18	SVM	Model	已训练	92.8%	30
19	SVM	Model	已训练	93.3%	30
20	KNN	Model	已训练	94.0%	30
21	SVM	Model	未训练	97.4%	30

我们还可以将结果导出为表格，放在我们论文的求解部分

点击导出模型后，我们就可以在对话框看到这个提示

我们只需要把T 改为我们自己的表名称即可完成预测

结果展示





举个例子---论文写作的时候怎么写?

第一步：构建评价指标体系

- 分类报告。分类报告图示可以直观得到模型各项参数，包括每一类别的精确率 (Precision)，召回率 (Recall)，F1 分数值 (F1-Score)。对于这三项值，其计算公式如下：

- 精确率

$$\text{Precision} = \frac{TP}{TP + FP} \quad (30)$$

- 召回率

$$\text{Recall} = \frac{TP}{TP + FN} \quad (31)$$

- F1 分数值

$$F1 = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} = \frac{TP}{TP + \frac{1}{2}(FP + FN)} \quad (32)$$

- 灵敏度 (Sensitivity)。灵敏度又被称为真阳性率，即 TP 率，定义为：

$$\text{Sensitivity} = TPR = \frac{TP}{TP + FN} \quad (34)$$

- 特异性 (Specificity)。特异性又被称为真阴性率，即 TN 率，定义为：

$$\text{Specificity} = TNR = \frac{TN}{TN + FP} \quad (35)$$

- 1-Specificity。称为假阳性率 (False Positive Rate, FPR)，定义为：

$$FPR = 1 - \text{Specificity} = \frac{FP}{FP + TN} \quad (36)$$

- 1-Sensitivity。称为假阴性率 (False Negative Rate, FNR)，定义为：

$$FNR = 1 - \text{Sensitivity} = \frac{FN}{FN + TP} \quad (37)$$

第二步：构建模型

5.3.2 多种多分类模型的建立

- 极端梯度提升 (eXtreme Gradient Boosting, XGBoost)。XGBoost 算法是一种基于树模型的优化模型，其将弱分类器组合，训练出一个较强的分类器。该算法通过多次迭代，生成一个新的树模型用于优化前一个树模型，随着迭代次数的增多，该模型的预测精度也会相应提高^[1]。

记通过数据处理后的数据集特征为 $R(x_{ij})_{m \times n}$ ，表示其包含 m 个用户， n 个特征，在训练中形成的 CART 树的集合记为 $F = \{f(x) = w_q(x), q: \mathbf{R}^n \rightarrow T, w \in \mathbf{R}^T\}$ ，其中 q 为树模型的叶节点决策规划， T 为某一树模型叶节点数量， w 为叶节点对应的得分^[2]。对于预测的 y 值，其计算公式为

$$\hat{y} = \varphi(x_i) = \sum_{k=1}^K f_k(x_i) \quad (17)$$

XGBoost 算法在每一次迭代过程中会保存前面所学习的模型，会将这些模型加入到新一轮迭代过程中，因此我们记第 i 个模型为预测结果为

$$\hat{y}_i^{(t)} = \hat{y}_i^{(t-1)} + f_t(x_i) \quad (18)$$

XGBoost 算法的目标函数计算公式如下

$$L^{(t)} = \sum_{i=1}^n l(y_i, \hat{y}_i^{(t-1)} + f_t(x_i)) + \gamma T + \frac{1}{2} \lambda \sum_{j=1}^T w_j^2 + \text{const} \quad (19)$$

上述公式中， l 为模型误差损失，描述在该模型下预测值与实际值之间的出差错损失， Ω 为模型叶节点的正则项惩罚系数， γ 与 λ 为模型的超参数^[3]。通常情况下，我们难以用枚举法得到在模型中所训练出来的树结构，因此这里采用贪婪算法，从单叶子节点开始，通过迭代方法，将其加入到树结构中，从而得到最优解，其计算公式^[4]如下

$$\mathcal{L}_{\text{split}} = \frac{1}{2} \left[\frac{(\sum_{i \in I_L} g_i)^2}{\sum_{i \in I_L} h_i + \lambda} + \frac{(\sum_{i \in I_R} g_i)^2}{\sum_{i \in I_R} h_i + \lambda} - \frac{(\sum_{i \in I} g_i)^2}{\sum_{i \in I} h_i + \lambda} \right] - \gamma \quad (20)$$

其中 $I_j = \{i | q(x_i) = j\}$ 为叶节点 j 上的样本集合^[5]，且有

$$g_i = \partial_{\hat{y}_i^{(t-1)}} l(y_i, \hat{y}_i^{(t-1)}) \quad h_i = \partial_{\hat{y}_i^{(t-1)}}^2 l(y_i, \hat{y}_i^{(t-1)}) \quad (21)$$



举个例子---论文写作的时候怎么写?

第三步：放模型性能评估表

评分项目	随机森林			XGBoost		
	准确率	平均绝对误差	均方误差	准确率	平均绝对误差	均方误差
语音通话整体满意度	0.5829	1.2910	6.5378	0.5949	1.2320	6.0331
网络覆盖与信号强度	0.5101	1.5304	7.6077	0.4936	1.4926	7.0562
语音通话清晰度	0.5424	1.2864	6.0691	0.5331	1.3039	6.0976
语音通话稳定性	0.5184	1.3996	6.5433	0.5184	1.4282	6.7928
手机上网整体满意度	0.4459	1.7692	8.6752	0.4359	1.7222	8.2265
网络覆盖与信号强度	0.3903	1.7322	7.5214	0.3775	1.7863	7.8775
手机上网速度	0.3775	1.7892	7.7664	0.3533	1.7222	6.9587
手机上网稳定性	0.3803	1.8348	8.0912	0.3889	1.7792	7.8846

表 7 KNN、SVM 各评分三项指标结果

评分项目	KNN			SVM		
	准确率	平均绝对误差	均方误差	准确率	平均绝对误差	均方误差
语音通话整体满意度	0.5893	1.2680	6.5055	0.5884	1.3582	7.3103
网络覆盖与信号强度	0.5009	1.5506	7.6924	0.4991	1.5948	8.0571
语音通话清晰度	0.5543	1.3002	6.2063	0.5543	1.3932	7.1133
语音通话稳定性	0.5239	1.4162	6.7385	0.5267	1.4678	7.2652
手机上网整体满意度	0.4217	1.8148	8.6268	0.4330	1.8575	9.3846
网络覆盖与信号强度	0.3946	1.7792	8.0271	0.3960	1.8504	8.6966
手机上网速度	0.3818	1.7806	7.7094	0.3732	1.8519	8.5071
手机上网稳定性	0.3832	1.7977	8.1054	0.3818	1.9259	9.0057

表 8 LightGBM、多分类逻辑回归各评分三项指标结果

评分项目	LightGBM			多分类逻辑回归		
	准确率	平均绝对误差	均方误差	准确率	平均绝对误差	均方误差
语音通话整体满意度	0.5737	1.2366	5.8131	0.5829	1.2330	6.0230
网络覆盖与信号强度	0.4899	1.5037	7.1298	0.5000	1.4797	7.0101
语音通话清晰度	0.5396	1.2606	5.6860	0.5451	1.2845	5.9843
语音通话稳定性	0.5110	1.3517	6.1455	0.5285	1.3923	6.5727
手机上网整体满意度	0.4444	1.7123	8.1852	0.4302	1.7821	8.7365
网络覆盖与信号强度	0.3846	1.7393	7.4772	0.3718	1.8547	8.4530
手机上网速度	0.3704	1.7222	7.2464	0.3761	1.8219	8.0840
手机上网稳定性	0.3675	1.8433	8.1140	0.3818	1.8305	8.2550

第四步：画出ROC曲线图/混淆矩阵图

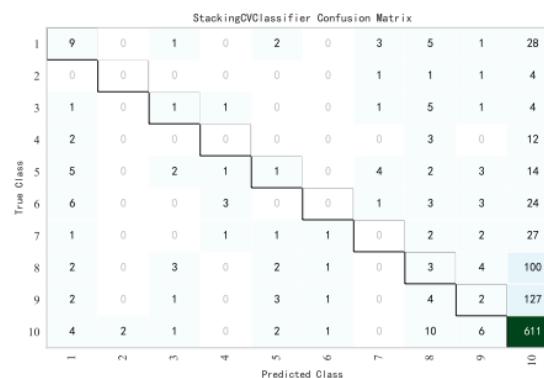
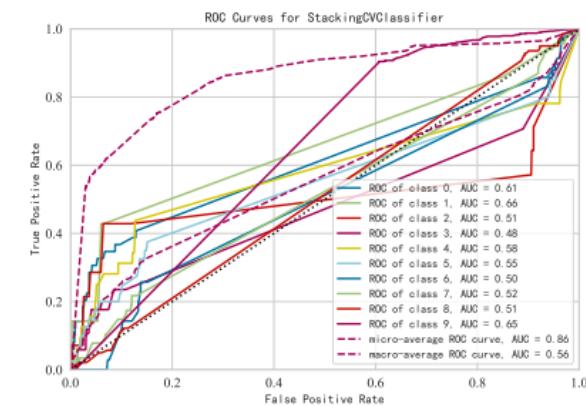


图 23 模型一混淆矩阵热力图 [语音业务-语音通话整体满意度]



这里还可以放上之前讲的：不同模型的评估指标柱状对比图

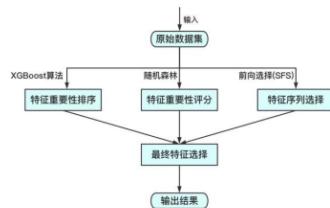


高阶优秀论文赏析

4.6 树模型

4.6.1 特征选择

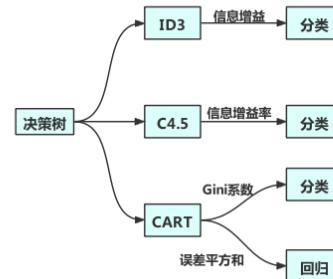
为增强模型的泛化能力并减少过拟合，本文将利用包裹式(Wrapper)特征选择法完成对各变量的遴选。相较于过滤式方法(Filter)，包裹式方法能够筛选出规模相对较小的优化特征子集。而机器学习本身具有的特征打分机制，能够被运用到特征选择任务上以达到筛选的目的。本文的特征选择流程如下图所示：



步骤二：特征重要性评分——随机森林

随机森林(Random Forest, RF)是一种将多棵互不关联的决策树(Decision Tree)加权集合构成的模型算法，是集成学习中 Bagging 方法的一种，具有随机属性且适用于较高维度的数据集。随机森林的核心在于样本采样和特征采样的随机性，其基本原理是输入待预测的 X 个原始数据，利用集成的 N 棵决策树执行判断和分类任务并得到 N 个分类结果，而随机森林通过集成汇总所有分类结果以最终输出投票次数最多的类别。

而决策树主要分为 ID3、C4.5 以及 CART 三种类型的算法，前两者分别以信息增益和信息增益率解决分类问题，但 CART 算法既能够以基尼系数(Gini Coefficient)为依据应用在分类问题上，也可以用于建立回归树模型。决策树各类型算法如右图所示：



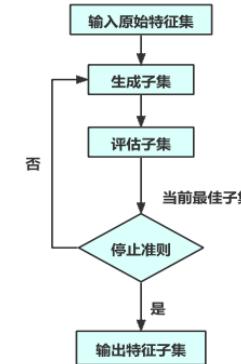
原理介绍 公式 流程图



步骤三：特征序列选择——前向选择(SFS)

由于高纬度特征往往存在特征冗余及特征无关的情况，因此当特征维度过高时，通常会导致分类器性能的下降。为了避免上述的“维灾难”，本文将利用特征序列选择(Sequential Feature Selection)来去除冗余特征及地相关性特征的影响，改善精度的同时减少模型训练所需时间。

特征序列选择的原理是从初始特征中筛选出高相关性的最优特征以达到降低数据集维度的目的，其基本步骤如右图所示：



4.6.2 线性回归模型

步骤一：建立模型

模型 1——线性回归(Linear Regression)

本文采用的线性回归模型是假设目标值与特征变量线性相关的基础上，以误差均方函数为损失函数，并令损失函数最小以确定参数的回归方法。

模型 2——岭回归(Ridge Regression)

岭回归在损失函数中增加了二阶正则项的最小二乘，也叫 L2 范数，具有降维作用的同时，它还以限制模型参数对异常样本的匹配程度，处理相关性高的数据集，进而提高模型面对多数正常样本的拟合精度。我组通过 RidgeCV 针对正则化强度 alpha 进行调参，得出在 alpha 为 14 时可以达到较好的拟合效果。

模型 3——Lasso 回归(Lasso Regression)

Lasso 回归类似于上文提到的岭回归，也通过构造惩罚函数来处理特征变量的共线性问题。但是，相较于岭回归，Lasso 回归可以通过使惩罚项由 L2 范数变为 L1 范数将相对不显著的特征变量系数压缩为 0，达到剔除变量的目的；而岭回归仅仅只在一定程度上压缩特征变量系数且会保留回归模型的全部变量。

03 时间序列预测模型入门必看



时间序列模型？还是得用深度学习！



相信大家都听说过深度学习，但是却没有使用过

怎么理解时间序列数据？怎么选择好的模型？

大家都说数模是边打边学，那么我们最起码要知道拿到数据集后，应该使用哪种模型，这才是我们的根本



接下来跟我一起学习吧！



1: 什么叫时间序列?

答：就是字面意思，有时间的数据集叫时间序列数据

2: 我该怎么判断选用哪些模型?

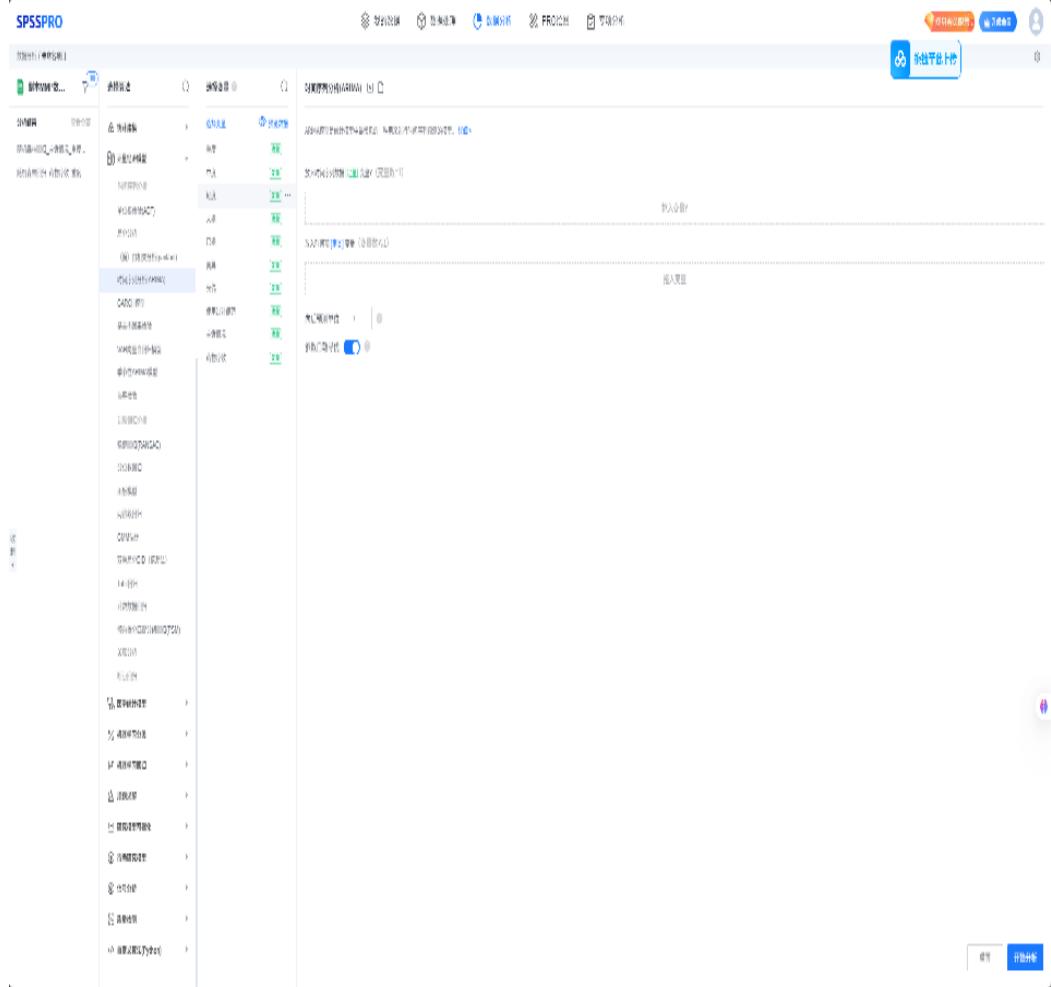
答：分三种情况

- 1: 数据集很少（只有5-10行）用灰色预测
- 2: 数据集中等（几百条）用ARIMA等普通模型
- 3: 数据集大（几千几万）用LSTM等深度学习模型

3: 有没有无脑软件可以操作这些的？

答：还真有

https://www.spsspro.com/?utm_source=biying&msclkid=b703b185da3219803316d9d0c4328f1f





不同时间序列模型应用场景

经典时间序列模型

自回归模型 (AR)	当前值与过去若干期值线性关系, 仅捕捉线性关系, 忽视外部影响因素 适用于平稳数据	利用过去值的线性组合预测当前值 通过最小化误差拟合模型。
移动平均模型 (MA)	当前值与过去若干期误差项的线性关系, 适用于平稳数据	仅考虑误差, 忽视数据自身的依赖关系 通过过去误差的线性组合预测当前值, 捕捉随机波动。
ARMA模型	结合AR和MA, 适用于平稳时间序列	仅适用于平稳数据, 参数选择复杂 结合自回归和移动平均, 通过线性组合捕捉数据的依赖和随机波动。
ARIMA模型	非平稳时间序列, 通过差分实现平稳化	对季节性和结构性变化处理有限 在ARMA基础上加入差分操作, 捕捉趋势和周期性变化。
季节性ARIMA (SARIMA) 模型	季节性时间序列, 具有周期性波动的数据	模型复杂, 参数众多, 易过拟合 在ARIMA基础上加入季节性差分和季节性ARMA成分, 处理周期性变化。
指数平滑模型 (Exponential Smoothing)	平滑时间序列数据, 适用于有趋势和季节性的情况	对突变和异常值敏感, 参数选择影响效果 通过加权平均过去数据, 权重随时间递减, 实现平滑预测。
Holt线性趋势模型 (Holt's Linear Trend Model)	具有线性趋势的时间序列	无法捕捉季节性, 模型假设简单 扩展指数平滑, 包含趋势成分, 通过双指数平滑捕捉趋势变化。
Holt-Winters季节性模型 (Holt-Winters Seasonal Model)	具有趋势和季节性的时序	对复杂季节性和非线性趋势处理有限 三指数平滑, 包含水平、趋势和季节性成分, 适应趋势和季节性变化。
向量自回归模型 (VAR)	多变量时间序列, 捕捉变量间相互依赖关系	参数众多, 需保证数据平稳, 难以解释 扩展AR模型, 考虑多变量之间的相互影响, 通过多方程联立预测。
GARCH模型 (Generalized Autoregressive Conditional Heteroskedasticity)	金融时间序列, 捕捉波动性变化和聚集效应	模型复杂, 参数估计困难 捕捉时间序列的波动性, 通过自回归和移动平均建模条件异方差。
状态空间模型 (State Space Models)	动态系统建模, 适用于复杂结构和非线性关系	模型构建和参数估计复杂, 需先验知识 使用隐藏状态和观测方程描述系统动态, 通过卡尔曼滤波等方法估计状态。
TBATS模型 (Exponential Smoothing State Space Model with Box-Cox Transformation, ARMA Errors, Trend and Seasonal Components)	复杂季节性和多重季节性时间序列	模型复杂, 参数众多, 计算资源需求高 结合Box-Cox变换、ARMA误差、趋势和季节性成分, 适应复杂季节性。

基于机器学习时间序列模型

支持向量回归 (SVR)	高维数据、非线性关系时间序列预测	对参数和核函数选择敏感, 计算复杂度高	通过核函数映射数据到高维空间, 寻找最大化间隔的回归平面。
长短期记忆网络 (LSTM)	处理长期依赖关系的时间序列, 如金融数据、语音数据	训练时间长, 需大量数据, 模型复杂	通过门控机制控制信息流, 捕捉长期依赖和复杂的时间序列模式。
Prophet模型	具有趋势和季节性的时间序列, 适用于商业预测	对异常值敏感, 参数调整有限	基于加性模型, 结合趋势、季节性和假日效应, 通过可解释组件进行预测。
随机森林回归	高维数据、非线性关系时间序列预测	模型较大, 训练时间长, 解释性较差	通过构建多个决策树并取平均值, 减少拟合, 提高预测精度。
梯度提升回归	复杂非线性关系时间序列预测, 高预测性能	易过拟合, 参数调节复杂	通过逐步构建树, 每棵新树修正前一棵树的误差, 提升整体模型性能。
Transformer	长期依赖关系、复杂模式的时间序列预测	需大量数据和计算资源, 训练复杂	利用自注意力机制捕捉序列中不同位置的信息, 适应长期依赖关系。
Temporal Fusion Transformers (TFT)	多变量时间序列预测, 结合注意力机制和门控机制	模型复杂, 训练时间长	结合自注意力和门控机制, 处理多变量和复杂时间依赖, 提升预测性能。
k-近邻回归 (k-Nearest Neighbors Regression, k-NN)	非参数时间序列预测, 适用于简单关系	计算复杂度高, 受噪声影响大	根据距离选择最近的k个邻居, 取其目标变量的平均值作为预测结果。
高斯过程回归 (Gaussian Process Regression, GPR)	不确定性评估和复杂函数拟合, 适用于小到中等规模数据	计算资源需求高, 扩展性差	基于高斯过程的贝叶斯方法, 通过协方差函数建模数据的平滑性和相关性。
极限学习机 (Extreme Learning Machine, ELM)	快速训练单隐层前馈神经网络, 适用于回归预测	需要适当选择激活函数, 参数调节较少	随机设置隐藏层权重, 快速计算输出层权重, 实现高效学习。

1问：深度学习怎么使用？

答：分两种

- 1: 可以询问AI，给你代码
- 2: 可以使用工具箱（推荐清风数学建模的工具箱）并且配套了教程

2问：这些模型应该怎么展示在论文中？

答：和之前一样的套路：模型原理介绍+公式+流程图
最后再放上你的预测结果表+图

4.2 ARIMA-LSTM 发电量预测模型的建立

对于中国发电量，ARIMA 与 LSTM 模型的融合能够对长期数据进行精准捕捉，提高发电量预测的精准度。预测模块使用长短期记忆神经网络，将融合后的特征 TSF 作为网络输入。选择 LSTM 作为预测模型有以下两个原因：LSTM 是适合于逆行时间拓展的模型，具有长期记忆功能，意味着 LSTM 可以有效处理时间序列数据，对时间上的长期依赖性有很好的处理能力；CNN 在提取局部特征方面具有优势，而利用 LSTM 的长期记忆性可以解决 CNN 局部处理的问题。LSTM 是为了解决长期记忆问题而提出的一种网络结构^[6]。LSTM 结合了记忆细胞状态 (cell state) 和三个门控结构：遗忘门(forget gate)、输入门(input gate)、输出门 (output gate)，其模型结构如图所示：

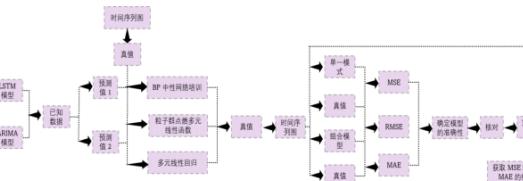


图 9ARIMA-LSTM 混合预测结构图

$$f_t = \sigma(W_f [h_{t-1}, x_t] + b_f) \quad (4)$$

其中 σ 为 sigmoid 激活函数， W_f 与 b_f 是系数矩阵。下一步是输入门决定要将多少新的信息添加到细胞状态中。输入门由两个组成部分构成，必须同时考虑它们的影响。一个使用 sigmoid 激活函数决定哪些信息需要更新，另一个 tanh 激活函数创建一个新的向量，最后将两部分进行联合对细胞状态进行更新。

评价准则

为了评估预测性能，使用均方根误差 (RMSE) 和平均绝对百分比误差 (MAPE) 这两个指标来评估预测的准确性。采用 Dstat 指标评价其方向精度。上述三个指标

$$RMSE = \sqrt{\frac{1}{N} \sum_{i=1}^N (\hat{x}_i - x_i)^2} \quad (5)$$

$$MAPE = \frac{100\%}{N} \sum_{i=1}^N \left| \frac{\hat{x}_i - x_i}{x_i} \right| \quad (6)$$

$$D_{stat} = \frac{1}{N} \sum_i a_i \times 100\% \quad (7)$$

提供更加准确、可靠的帮助。

表 8 单模型预测效果表										
	ARIMA 模型	支持向量机	LSTM	SARIMA						
时间	实际值	预测值	误差%	预测值	误差%	预测值	误差%	预测值	误差%	单位
2024.02	5264	6798.31	28.15	4839.79	8.06	6556.18	24.55	7188.33	36.56	亿千瓦时
2024.03	6631	6113.17	7.81	4880.36	26.4	6727.94	1.45	6800.61	2.56	亿千瓦时
2024.04	6361	7443.47	17.02	5105.24	19.74	8050.02	26.36	6894.9	8.39	亿千瓦时
2024.05	6724	6670.52	0.8	4947.05	26.43	804.128	9.88	6964.9	3.58	亿千瓦时
2024.06	7033	6492.74	7.68	4824.76	31.4	6837.73	2.71	7046.87	0.2	亿千瓦时
2024.07	7754	6542.57	15.67	4950.22	36.19	7242.08	4.25	6751.64	12.97	亿千瓦时
2024.08	7607	7212.3	4.93	5109.9	29.66	7243.9	4.71	6731.67	11.09	亿千瓦时
2024.09	6947	7217.93	3.9	5065.83	27.08	6828.3	1.71	7180.5	3.36	亿千瓦时
2024.10	6603	7126.91	7.93	5064.97	23.29	6976.25	5.65	7220.67	9.35	亿千瓦时

15

表 9 电力需求预测模型与组合模型预测性能比较										
	ARIMA-LSTM	CNN-LSTM	ARIMA-GM(1,1)	ARIMA						
时间	实际值	预测值	误差%	预测值	误差%	预测值	误差%	预测值	误差%	单位
2024.02	5264	6889.02	6.35	4354.28	20.72	6699.32	27.27	5737.53	11.58	亿千瓦时
2024.03	6631	6183.3	6.75	6274.55	5.38	6119.27	7.72	6487.41	2.17	亿千瓦时
2024.04	6361	7205.19	14.53	6161.04	31.4	7226.19	14.39	6412.46	0.81	亿千瓦时
2024.05	6724	6804.75	1.2	5835.44	13.21	6727.44	0.05	6614.11	1.63	亿千瓦时
2024.06	7033	6294.33	10.5	5766.47	18.01	6234.92	11.49	6897.58	1.93	亿千瓦时
2024.07	7754	6501.09	16.2	5956.61	27.86	6266.38	19.23	7200.8	8.47	亿千瓦时
2024.08	7607	7333.92	3.59	5882.49	26.61	7279.37	4.3	7004.82	7.63	亿千瓦时
2024.09	6947	7209.39	5.07	5390.18	22.41	7145.86	2.85	6982.18	0.51	亿千瓦时
2024.10	6603	7246.8	9.75	5181.25	21.53	7053.3	6.82	6444.98	2.39	亿千瓦时
2024.11	6718	7185.52	6.9	5040.34	24.97	7201.3	7.19	6879.01	2.4	亿千瓦时
2024.12	8156	7096.07	13	5076.06	37.76	7239.53	11.24	7264.26	10.93	亿千瓦时
2025.01	7232	7439.83	2.87	4832.95	33.17	7673.63	6.11	7373.34	6.84	亿千瓦时
2025.02	6235	6832.12	9.58	5083.53	18.47	7620.74	22.23	6957.76	5.82	亿千瓦时
2025.03	6944	6577.11	5.28	4980.34	28.22	7225.5	4.05	6699.38	3.6	亿千瓦时
2025.04	6362	7352.71	15.57	5083.72	20.8	8147.11	28.06	6782.42	6.61	亿千瓦时
2025.05	6716	6913.55	2.8	5016.18	25.01	7625.97	13.55	7101.73	5.74	亿千瓦时
2025.06	7451	6964.67	7.17	4764.68	36.05	7273.13	2.39	7021.39	5.77	亿千瓦时
2025.07	8324	7065.64	15.12	4857.44	41.65	7447.12	10.53	7332.39	1.91	亿千瓦时
2025.08	8520	7701.31	9.61	4361.16	48.81	7700.28	9.62	7052.43	1.723	亿千瓦时
2025.09	7092	7639.09	7.71	4939.03	38.07	7830.36	10.69	7072.28	0.28	亿千瓦时
2025.10	6834	7207.38	6.63	4084.25	40.24	7290.51	6.68	6836.67	0.04	亿千瓦时

16

表 9 电力需求预测模型与组合模型预测性能比较

表 9 电力需求预测模型与组合模型预测性能比较



学习的难度+时间

机器学习+时间序列预测模型是考题的热点，那么对于小白来说怎么学习，难度和如何安排时间成为了大家最关心的问题

- 1：学习规划安排：首先将这门课的视频吃透，并且一定要去自己跑一遍，而不是边看边运行，这样根本没有用
- 2：自学安排：网上的大多数教程都是偏向原理，我们其实更需要的是知道怎么用它即可，那么自己在家里找数据，跑代码才是最合适的自学方法
- 3：怎么找数据？---请看下一页



小白怎么最高效的练习题库？

推荐：[Kaggle: Your Home for Data Science](#)

这里面有大量的数据集，可以找到然后下载，都是免费的

第二步：运用AI

在这个时代变换下，有AI我们不应该抵触，更应该合理使用它，尤其是对小白来说，更要学会使用它

第三步：打开matlab/python 开始练习吧！

The screenshot shows the Kaggle homepage with a search bar at the top. Below it, there's a section titled '数据' (Data) which displays a list of datasets. The first dataset listed is '外卖单内向行为数据' (Food delivery single-sided behavior data), followed by '自然灾害遥感卫星遥感' (Natural disaster remote sensing satellite remote sensing), '中国空气质量 120231 - 6 个月' (China air quality 120231 - 6 months), 'Superhero_whichHero_dataset' (Superhero_whichHero_dataset), 'Facebook 粉丝最喜欢的歌手数据集' (Facebook fans' favorite singer dataset), and '餐饮业餐券优惠券' (Catering industry meal券 discount coupons). Each dataset entry includes a thumbnail, a title, a brief description, and a '查看' (View) button.



最后总结：我们重点分为两步

第一步：会用模型

需要学会最基本的实战
操作

需要学会如何合理使用
AI

需要多加练习

争取练够100道小案例分
享题

第二步：会写论文

- 1: 模型流程图的绘制
- 2: 多看往届赛题的
优秀论文
- 3: 只看不练 等于白
看！