内蒙古师范大学

计算机科学技术学院

本科毕业论文

|  |  |
| --- | --- |
| **题 目：** | 基于python的电影票房季节性分析与预测 |
| **学生姓名：** | 窦玉琢 |
| **学 号：** | 20161104542 |
| **所属学院：** | 计算机科学技术学院 |
| **专业班级：** | 16云分析 |
| **指导老师：** | 苗忠义 |

**完成日期 2019 年 4 月**

**目录**

**[摘要：](#_Toc1352017110_WPSOffice_Level1)** **[3](#_Toc1352017110_WPSOffice_Level1)**

**[ABSTRACT](#_Toc394993920_WPSOffice_Level1)** **[3](#_Toc394993920_WPSOffice_Level1)**

**[一、概述](#_Toc790860563_WPSOffice_Level1)** **[4](#_Toc790860563_WPSOffice_Level1)**

[1.1研究背景及意义](#_Toc394993920_WPSOffice_Level2) [4](#_Toc394993920_WPSOffice_Level2)

[1.2研究现状](#_Toc790860563_WPSOffice_Level2) [4](#_Toc790860563_WPSOffice_Level2)

**[二、相关技术简介](#_Toc1217191058_WPSOffice_Level1)** **[5](#_Toc1217191058_WPSOffice_Level1)**

[2.1 Python](#_Toc1217191058_WPSOffice_Level2) [5](#_Toc1217191058_WPSOffice_Level2)

[2.2 决策树模型](#_Toc400890484_WPSOffice_Level2) [5](#_Toc400890484_WPSOffice_Level2)

[2.3时间序列分析](#_Toc1110163949_WPSOffice_Level2) [6](#_Toc1110163949_WPSOffice_Level2)

[2.4 Numpy](#_Toc1187565707_WPSOffice_Level2) [6](#_Toc1187565707_WPSOffice_Level2)

[2.5 matplotlib](#_Toc703822331_WPSOffice_Level2) [6](#_Toc703822331_WPSOffice_Level2)

[2.6 Pandas](#_Toc801989441_WPSOffice_Level2) [7](#_Toc801989441_WPSOffice_Level2)

[2.7 Scikit—learn](#_Toc1429166315_WPSOffice_Level2) [7](#_Toc1429166315_WPSOffice_Level2)

**[三、课题概览](#_Toc400890484_WPSOffice_Level1)** **[7](#_Toc400890484_WPSOffice_Level1)**

[3.1课题思路](#_Toc393664510_WPSOffice_Level2) [7](#_Toc393664510_WPSOffice_Level2)

**[四、项目过程](#_Toc1110163949_WPSOffice_Level1)** **[8](#_Toc1110163949_WPSOffice_Level1)**

[4.1数据处理](#_Toc2069786810_WPSOffice_Level2) [8](#_Toc2069786810_WPSOffice_Level2)

[4.2数据可视化](#_Toc1966801564_WPSOffice_Level2) [8](#_Toc1966801564_WPSOffice_Level2)

[4.3任务描述](#_Toc1965591524_WPSOffice_Level2) [10](#_Toc1965591524_WPSOffice_Level2)

[4.4进行机器学习建模](#_Toc955802067_WPSOffice_Level2) [11](#_Toc955802067_WPSOffice_Level2)

[4.5决策树模型训练与评估](#_Toc987660509_WPSOffice_Level2) [12](#_Toc987660509_WPSOffice_Level2)

**[五、 结语](#_Toc1187565707_WPSOffice_Level1)** **[16](#_Toc1187565707_WPSOffice_Level1)**

[参考文献](#_Toc703822331_WPSOffice_Level1)**[17](#_Toc703822331_WPSOffice_Level1)**

**摘要：**

自我国自主拍摄了电影[《定军山》](https://www.baidu.com/s?wd=%E3%80%8A%E5%AE%9A%E5%86%9B%E5%B1%B1%E3%80%8B&tn=SE_PcZhidaonwhc_ngpagmjz&rsv_dl=gh_pc_zhidao" \t "/Users/doudou/Documents\\x/_blank)起，就标志着中国电影事业开始发展，经过了100多年的摸爬滚打，我国的影片创作质量不断提高，卓越的影片不断涌现，题材各异，融合了思想、潮流与艺术性，形成中国电影独特景象，出现了电影行业百家争鸣的美好景像，也因此得到了广泛网友观众们的好评，收视率、票房等硬性数据指标给了我们答案，这一切的一切都预示着，我国的电影事业正在进入内生发展的高速阶段，正处于由电影大国迈向电影强国的发展黄金时代。在如此形势下，本次课题，即票房的季节性分析与预测就显得尤为必要，论文是通过对1995年～2015年的票房月度分析，然后根据数据构建决策树模型、模型训练，并且引入时间序列分析模型进行了周期性分析，从而通过预测得出2016年预测数据，本课题是由python进行实现的，使用numpy、matplotlib和seaborn进行了数据的可视化。

**关键词**：电影票房、数据分析、季节性分析、python

**ABSTRACT**

Since the film Dingjunshan was made independently in China, it has marked the beginning of the development of Chinese film industry. After more than 100 years of hard work, the quality of film creation in China has been continuously improved, and outstanding films have been constantly emerging, with different themes, integrating ideas, trends and artistry, forming a unique picture of Chinese film, and creating a beautiful scene of hundreds of schools of thought in the film industry This has been well received by a wide range of netizens, audiences, box office and other hard data indicators give us the answer, all of which indicate that China's film industry is entering a high-speed stage of endogenous development, and is in the golden age of development from a big country of film to a powerful country of film. In such a situation, the seasonal analysis and prediction of box office is particularly necessary. The paper is based on the monthly analysis of box office from 1995 to 2015, then constructs the decision tree model and model training according to the data, and introduces the time series analysis model for periodic analysis, so as to get 2016 prediction data through prediction. This topic is carried out by python Finally, we use numpy, Matplotlib and Seaborn to visualize the data.

**Keywords**：box—office、data analysis、seasonal analysis、python

## 一、概述

### 1.1研究背景及意义

回顾2019年，中国电影交上了一份亮丽的成绩单：中国电影市场总票房达到642.66亿，较2018年同比增长5.4%。去年2月，在春节档的引领下，产出票房111.61 亿，占年度电影票房的18%。虽然本次新冠肺炎疫情对2020年电影市场是有一定的影响。受疫情影响，原计划在春节档上映的《夺冠》《唐人街探案3》《紧急救援》等影片暂缓上映，全国影院暂停营业，彰显了电影人的社会责任感。但是，阴霾掩盖不了月光白衣，疫情抵挡不了华夏共力，希望和春天一定会如约而至。

中国电影行业能够如此的蓬勃发展，能够屡破记录，不仅要归功于以年轻的创作者们为代表的创新力量，以及国家在关键时刻给出的政策支持，使得一批青年导演成长起来，为中国电影市场提供了类型丰富的产品，提升了国产电影的竞争力；而还要归功于这个大数据时代和各种新型技术产业的兴起，比如新兴的数据分析与预测技术。百度其实很早就已经开始着手研究并在内部测试票房预测了，据相关人士透露，基于百度大数据的票房预测已经能够做到80%以上的准确率。电影票房的影响因素受多方面影响，例如电影剧本质量、导演演员的知名度、影片类型等，但这次的课题我们主要对电影的季节性这一因素进行深入研究，也就是电影所谓的“档期”安排。当然，也并不是所有的电影全部遵循普遍的季节性变化，这种特例我们不予讨论，我们主要研究普适性的季节性规律。

### 1.2研究现状

电影史上最早关于票房预测的摸索，可以追溯到70年代，美国电影业的迅猛发展，市场风险激增。同时现代媒介的发展，居住环境的变化，出行方式的改变使得美国观众了解电影信息的渠道也发生了诸多改变，关于票房分析的研究急待新的理论支撑。在这样的背景下，诞生了一批全新的分析方法，其中以巴里•利特曼的票房预测模型最为著名。密歇根州立大学经济学家巴里•利特曼（Barry Litman）专注于娱乐领域的研究，在80年代提出了一个极具参考价值的电影票房分析模型，他把影响电影在商业上成功的变量划分为三大部分：创意、发行/上映时间以及电影营销。接着，利特曼将自变量与因变量（电影收入）进行层次回归的分析方法，得到的了一个回归方程式：

Y＝-28.482×106+7.232×106顶级导演+14.846×106明星+11.818×106科幻+13.858×106续集+24.932×106奥斯卡提名－4.966×106剧情+6.972×106影评+3.814×106大发行公司。

第一个票房分析模型的出现，为今后的票房分析模型打下了坚实的基础，此后的模型被不断改良和发展，直到2013年，美国知名互联网公司谷歌公布了一项研究成果——电影票房预测模型，据介绍，该项模型能够提前一个月进行电影上映首周的票房预测，且准确率高达94%，发布之初便引起全球关注，如今距离2013已经过去7年，这项研究模型更加完善普及，被运用范围也越来越广，不断超越，如今已经较为成熟，精准度成熟度都更上了一个台阶。

## 二、相关技术简介

### 2.1 Python

有句老话讲得好“人生苦短，我用python！”，虽说是一句玩笑话，但也充分体现出python的优势。各行各业的领军人物也对python语言赞不绝口：日本野村证券副首席马修·汉普森在Quant Conference上明确表示：Python已经取代了Excel。现在穿过我们的交易大厅，交易员都在用Python写代码；在高盛的Analyst面试中，Python成为重点考察的技能；去年就有外媒报道，摩根大通花费108亿美元在技术培训和技术人员招聘上，并强制自家分析师参与Python课程学习。

Python的第一个优点，那就是简单易学。很多学过Java的朋友都知道，Python语法简单的多，代码十分容易被读写。我们在处理数据的时候，一般都希望数据能够转化成可运算的数字形式，这样，不管是没学过编程的人还是学过编程的人都能够看懂这个数据。

Python在数据分析和交互、探索性计算以及数据可视化等方面都显得比较活跃，这就是Python经常被用作数据分析的原因之一，python拥有numpy、matplotlib、scikit-learn、pandas、ipython等工具在科学计算方面十分有优势，尤其是pandas，在处理中型数据方面可以说有着无与伦比的优势，已经成为数据分析中流砥柱的分析工具。

Python也具有强大的编程能力，这种编程语言不同于R或者matlab，python有些非常强大的数据分析能力，并且还可以利用Python进行爬虫，写游戏，以及自动化运维，在这些领域中有着很广泛的应用，这些优点就使得一种技术去解决所有的业务服务问题，这就充分的体现的Python有利于各个业务之间的融合。如果使用Python，能够大大的提高数据分析的效率。

Python还提供了非常完善的基础代码库，覆盖了网络、文件、GUI、数据库、文本等大量内容，被形象地称作“内置电池(Batteries included)”。用Python开发，许多功能不必从零编写，直接使用现成的即可。

### 2.2 决策树模型

决策树是一种非参数的监督学习方法，它主要用于分类和回归问题。决策树模型通过一系列if then决策规则的集合，将特征空间划分成有限个不相交的子区域，对于落在相同子区域的样本，决策树模型给出相同的预测值。这些if then决策规则之间的层次关系形成一个树形结构，称之为决策树，这些不相交的子区域和树结构的叶子节点一一对应。决策树常见的算法有CHAID、CART、Quest和C5.0。对于每一个决策要求分成的组之间的“差异”最大。各种决策树算法之间的主要区别就是对这个“差异”衡量方式的区别。决策树很擅长处理非数值型数据，这与神经网络智能处理数值型数据比较而言，就免去了很多数据预处理工作。下面是决策树中几个重要概念：信息熵（Entropy）、信息增益和信息增益率（Information gain ratio）

### 2.3时间序列分析

因为本课题是预测关于年月对于电影票房的影响，所以我选择加入时间序列分析的算法。时间序列简单的说就是各时间点上形成的数值序列，时间序列分析就是通过观察历史数据预测未来的值。时间序列分析并不是关于时间的回归，它主要是研究自身的变化规律的。学术化的概念就是：时间序列是指按时间顺序排列的一组随机变量X1,X2,…,Xt表示一个随机事件的时间序列。时间序列分析的目的是给定一个已被观测了的时间序列，预测该序列的未来值。间序列是按照[时间](https://baike.baidu.com/item/%E6%97%B6%E9%97%B4/25651" \t "/Users/doudou/Documents\\x/_blank)排序的一组随机变量，它通常是在相等间隔的时间段内依照给定的采样率对某种潜在过程进行观测的结果。时间序列数据本质上反映的是某个或者某些随机变量随时间不断变化的趋势，而时间序列预测方法的核心就是从数据中挖掘出这种规律，并利用其对将来的数据做出估计。

### 2.4 Numpy

NumPy是Python中科学计算的基础软件包。它是一个提供多了维数组对象，多种派生对象（如：掩码数组、矩阵）以及用于快速操作数组的函数及API，它包括数学、逻辑、数组形状变换、排序、选择、I/O 、离散傅立叶变换、基本线性代数、基本统计运算、随机模拟等等。NumPy包的核心是ndarray对象。它封装了python原生的同数据类型的n维数组，为了保证其性能优良，其中有许多操作都是代码在本地进行编译后执行的。

### 2.5 Matplotlib

Matplotlib是一个Python 2D绘图库，可以生成各种硬拷贝格式和跨平台的交互式环境的出版物质量数据。Matplotlib可用于Python脚本，Python和IPython shell，Jupyter notebook，Web应用程序服务器和四个图形用户界面工具包。通过 Matplotlib，开发者可以仅需要几行代码，便可以生成绘图，直方图，功率谱，条形图，错误图，散点图等。

### 2.6 Pandas

Pandas 是基于[NumPy](https://baike.baidu.com/item/NumPy/5678437" \t "/Users/doudou/Documents\\x/_blank) 的一种工具，该工具是为了解决数据分析任务而创建的。Pandas 纳入了大量库和一些标准的[数据模型](https://baike.baidu.com/item/%E6%95%B0%E6%8D%AE%E6%A8%A1%E5%9E%8B/1305623" \t "/Users/doudou/Documents\\x/_blank)，提供了高效地操作大型数据集所需的工具。pandas提供了大量能使我们快速便捷地处理数据的函数和方法。你很快就会发现，它是使Python成为强大而高效的数据分析环境的重要因素之一。

### 2.7 Scikit—learn

Scikit-learn，又写作sklearn，是一个开源的基于python语言的机器学习工具包。它通过NumPy, SciPy和Matplotlib等python数值计算的库实现高效的算法应用，并且涵盖了几乎所有主流机器学习算法。

在工程应用中，用python手写代码来从头实现一个算法的可能性非常低，这样不仅耗时耗力，还不一定能够写出构架清晰，稳定性强的模型。更多情况下，是分析采集到的数据，根据数据特征选择适合的算法，在工具包中调用算法，调整算法的参数，获取需要的信息，从而实现算法效率和效果之间的平衡。而sklearn，正是这样一个可以帮助我们高效实现算法应用的工具包。其中sklearn中常用的模块有分类、回归、聚类、降维、模型选择和预处理模块。本次课题使用到了其中的回归模块，即GradientBoostingRegressor（梯度提升回归）。

## 三、课题概览

### 3.1课题思路

本文首先对课题研究的背景以及意义开始进行介绍，了解分析了国内外对课题的研究状况后，对于这次课题使用的相关技术的介绍进行了简单阐述，紧接着对课题的流程及代码进行详细的分析与介绍，包括数据处理、数据可视化、任务描述、机器学习建模和决策树的训练与评估所有模块，涵盖了本文的整个核心内容。本课题主要是通过Python语言对1995-2015年月度票房收入电影数据进行分析，收集其中电影名称、发行时间、发行地、票房等关键信息进行分析，通过构建决策树模型、模型训练，再加上时间序列分析技术，从而得出发行时间对电影票房的影响，然后使用numpy、seaborn、matplotlib等可视化工具进行可视化分析及展示。



## 四、项目过程

### 4.1数据处理

通过网络获得了1990——2016年度的电影票房数据，然后使用pandas等工具进行数据的读取和删改，获得了处理后的数据，具体代码如下：

movies\_df = pd.read\_csv('tmdb-movie-metadata/tmdb\_5000\_movies.csv')

movies\_df.dropna(how="any",subset=['release\_date'],inplace=True)

接下来进行数据探索，将有关数据进行整理：

movies\_df['release\_date'] = pd.to\_datetime(movies\_df['release\_date'])

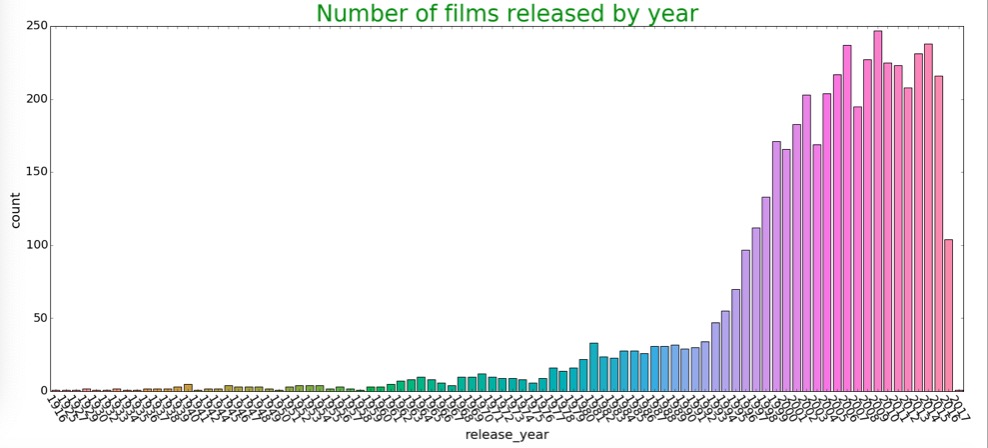
movies\_df['release\_year'] = movies\_df['release\_date'].dt.year

movies\_df['release\_month'] = movies\_df['release\_date'].dt.month

movies\_df['release\_year\_month'] = movies\_df['release\_year'] \* 100 + movies\_df['release\_month']

### 4.2数据可视化

将整理好的数据通过matplotlib进行可视化，方便读者阅读。将电影数量按照年统计显示：



然后将数据按照年月统计进行转换和可视化：

year\_month\_revenue=movies\_df.groupby('release\_year\_month')['revenue'].sum().reset\_index()

plt.figure(figsize=(25, 10))

plt.plot(year\_month\_revenue['revenue'])

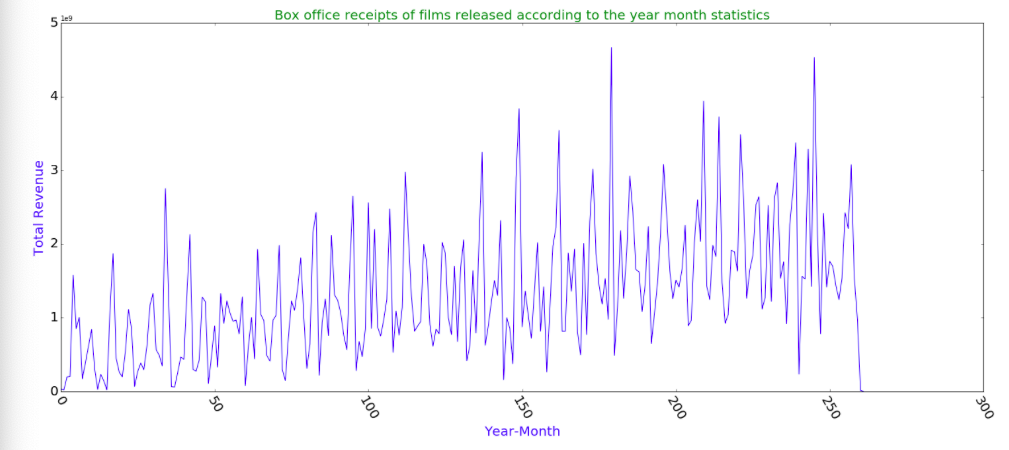
plt.title('Box office receipts of films released according to the year month statistics',fontsize=20,color='green')

plt.ylabel('Total Revenue',fontsize=20,color='blue')

plt.xlabel('Year-Month',fontsize=20,color='blue')

plt.xticks(rotation=300,fontsize=20)

plt.yticks(rotation=0,fontsize=20)

plt.show()

### 4.3任务描述

然后利用1995年至2015年的所有月度票房收入数据，模型分析其中的季节性规律，自回归的方式预测未来 2016 年月度票房收入情况并进行可视化：

rain = year\_month\_revenue[year\_month\_revenue['release\_year\_month'] <= 201600]['revenue']

test = year\_month\_revenue[year\_month\_revenue['release\_year\_month'] > 201600]['revenue']

plt.figure(figsize=(18, 6))

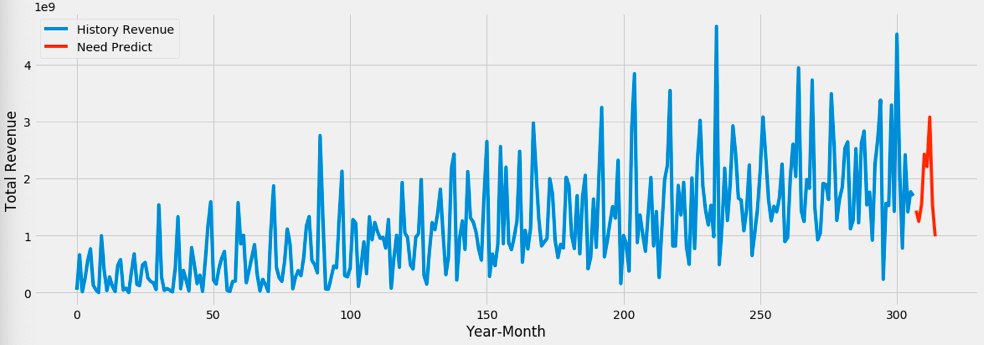
plt.plot(train, label='History Revenue')

plt.plot(test, label='Need Predict')

plt.ylabel('Total Revenue')

plt.xlabel('Year-Month')

plt.legend()

plt.show()

### 4.4进行机器学习建模

print('训练集个数：', train.shape[0])

print('测试集个数：', test.shape[0])

接下来的代码进行构造决策树模型的相关特征，即进行特征工程。

all\_revenue = year\_month\_revenue['revenue']

window = 30

all\_features = []

all\_labels = []

for tr\_i in range(window, all\_revenue.shape[0]):

feature = []

window\_temps = []

for wi in range(window):

window\_temps.append(all\_revenue[tr\_i-wi-1])

feature.append(all\_revenue[tr\_i-wi-1])

temp\_stat = [np.mean(window\_temps), np.max(window\_temps),

np.std(window\_temps), np.min(window\_temps),

np.median(window\_temps)]

feature.extend(temp\_stat)

diff\_feat = [window\_temps[-1] - pre for pre in window\_temps[:-1]]

feature.extend(diff\_feat)

temp\_diff\_stat = [np.mean(diff\_feat), np.max(diff\_feat),

np.std(diff\_feat), np.min(diff\_feat),

np.median(diff\_feat)]

feature.extend(temp\_diff\_stat)

all\_features.append(feature)

all\_labels.append(all\_revenue[tr\_i])

all\_features = np.array(all\_features)

all\_labels = np.array(all\_labels)

train\_x = all\_features[:-test.shape[0]]

train\_y = all\_labels[:-test.shape[0]]

test\_x = all\_features[-test.shape[0]:]

test\_y = all\_labels[-test.shape[0]:]

print('训练集样本数：', train\_x.shape)

print('测试集样本数：', test\_x.shape)

### 4.5决策树模型训练与评估

使用scikit—learn库中的GradientBoostingRegressor（梯度提升回归）进行算法拟合。

from sklearn.ensemble import GradientBoostingRegressor

from sklearn.metrics import mean\_squared\_error

train\_y = np.log1p(train\_y)

test\_y = np.log1p(test\_y)

regr = GradientBoostingRegressor()

regr.fit(train\_x, train\_y)

predict\_revenue = regr.predict(test\_x)

mse\_error = mean\_squared\_error(test\_y, predict\_revenue)

print('Test RMSE: %.3f' % mse\_error)

然后进行数据可视化，可以看出数据拟合结果基本差不多。

plt.figure(figsize=(18, 6))

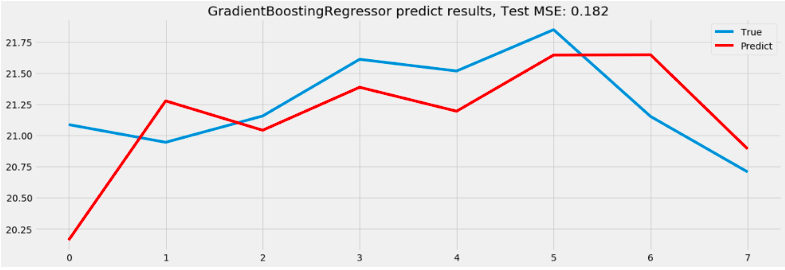
plt.plot(test\_y, label='True')

plt.plot(predict\_revenue, color='red', label='Predict')

plt.legend()

plt.title('GradientBoostingRegressor predict results, Test MSE: %.3f' % mse\_error)

plt.show()



然后将上述代码进行封装，封装成函数，用于评估不同窗口下，模型预测的 rmse 的影响。

def check\_diff\_window\_rmse(window):

window = 30

all\_features = []

all\_labels = []

for tr\_i in range(window, all\_revenue.shape[0]):

feature = []

window\_temps = []

for wi in range(window):

window\_temps.append(all\_revenue[tr\_i-wi-1])

feature.append(all\_revenue[tr\_i-wi-1])

temp\_stat = [np.mean(window\_temps), np.max(window\_temps),

np.std(window\_temps), np.min(window\_temps),

np.median(window\_temps)]

feature.extend(temp\_stat)

diff\_feat = [window\_temps[-1] - pre for pre in window\_temps[:-1]]

feature.extend(diff\_feat)

temp\_diff\_stat = [np.mean(diff\_feat), np.max(diff\_feat),

np.std(diff\_feat), np.min(diff\_feat),

np.median(diff\_feat)]

feature.extend(temp\_diff\_stat)

all\_features.append(feature)

all\_labels.append(all\_revenue[tr\_i])

all\_features = np.array(all\_features)

all\_labels = np.array(all\_labels)

train\_x = all\_features[:-test.shape[0]]

train\_y = all\_labels[:-test.shape[0]]

test\_x = all\_features[-test.shape[0]:]

test\_y = all\_labels[-test.shape[0]:]

train\_y = np.log1p(train\_y)

test\_y = np.log1p(test\_y)

regr = GradientBoostingRegressor()

regr.fit(train\_x, train\_y)

predict\_revenue = regr.predict(test\_x)

mse\_error = mean\_squared\_error(test\_y, predict\_revenue)

return mse\_error, predict\_revenue

再次调用函数

windows = range(30)rmse\_errors = []

for window in windows:

rmse, \_ = check\_diff\_window\_rmse(window)

rmse\_errors.append(rmse)

print('process window = {}, rmse = {}'.format(window, rmse))

然后进行可视化，查看在不同窗口下均方根误差的大小，以下为绘图代码：

plt.figure(figsize=(14, 5))

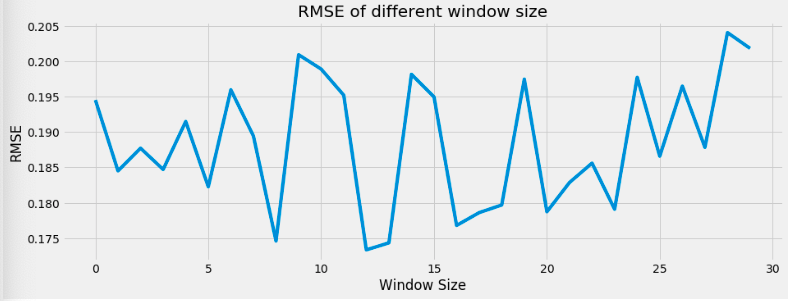
plt.plot(windows, rmse\_errors)

plt.xlabel('Window Size')

plt.ylabel('RMSE')

plt.title('RMSE of different window size')

plt.show()



由图可知：在window=12的时候，均方根误差（rmse）最小。

接下来，让模型对未来几个月的票房进行预测，根据模型学习的季节效应选择最佳时间。

rmse, predicts = check\_diff\_window\_rmse(window=12)

test\_y = np.expm1(test\_y)

predicts = np.expm1(predicts)

plt.figure(figsize=(18, 6))

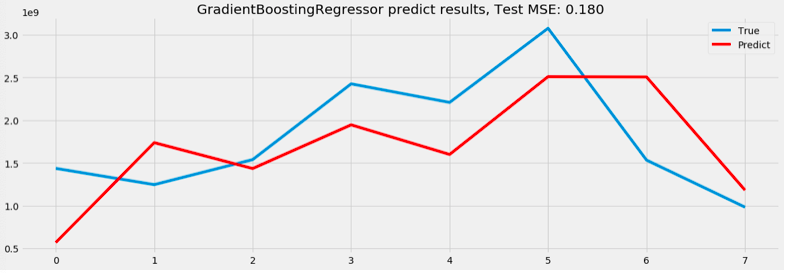
plt.plot(test\_y, label='True')

plt.plot(predicts, color='red', label='Predict')

plt.legend()

plt.title('GradientBoostingRegressor predict results, Test MSE: %.3f' % rmse)

plt.show()



可以看出，模型预测虽然存在一定的误差，但总体趋势基本一致，可以判断在未来的第 6 - 7 月，票房收入达到最高。

## 结语

本文首先从研究现状、意义及背景三个方面说明了电影票房季节性分析与预测的研究有很大的可研究价值，然后通过对课题所用到的技术如Python和决策树等进行了概述。接下来经过简单的文字性描述理清了课题的思路，最后进行对代码的分析与解读。经过本次实验的研究我学习到了很多：通过不断实践和探索，终于实现了基于Python语言的的电影票房季节性分析与预测，通过本次课题的研究与实现，我最大的收获就是，让我充分了解了要想完成一个完整的数据分析项目，是有庞大的工程量和操作难度，想制成一个完美的数据分析结果需要非常细致的工作，其中最重要的还是数据的获取、清洗和特征提取等一系列的数据处理的工作，数据好了分析预测才能够更加精准，而这又需要极大的耐心和丰富的经验。而且我发现理论知识和真正的实践有很大差距。要真正做到理论联系实际，还是需要多实践多练习，实践才能出真知。还有由于在此之前对于Python等知识了解的不够深入，各种语句运用起来不够灵活，思维比较刻板闭塞，所以在开始的时候遇到了很多的困难，屡屡想要放弃，不过通过在网上不断搜寻有关资料和积极寻找导师商议，最终通过苗老师不辞辛苦循循善诱的帮助下最后都得到了解决，完成了此次课题任务，在此过程中，我不仅学到了很多书面的、理论性的知识，同时也提高了自己解决实际问题的能力，尤其是学会如何从大量的信息中筛选出所需有用的信息，同时我更加深刻的体会到了，虽然书本上的大部分知识都是有价值，正确的，但实际上每个人编程的思路和对数据处理的方法、思想都是不同的，这就要求我们一定要通过实践才能找到解决问题的方案。在此次毕业设计活动中，我不断的提高了自己，也得到了宝贵的经验，我相信这些对我以后的发展都会有很大帮助。最后，再次对于我的导师表示衷心的感谢，苗老师严谨治学的态度让我深受鼓舞，老师治学态度的影响不仅仅限于这次课题论文设计研究，更影响了我今后的人生观价值观，衷心祝愿苗老师桃李遍天下。

### 参考文献：

[1][平珊珊](http://www.wanfangdata.com.cn/details/detail.do?_type=perio&id=shanddz201912064). 我国电影票房影响因素分析及预测[J]. 社会科学前沿,2019,8(04):489-495.doi:10.12677/ASS.2019.84069.

[2][倪珅淼](http://www.wanfangdata.com.cn/details/detail.do?_type=perio&id=shanddz201912064). 电影票房影响因素分析[J]. [改革与开放](http://www.wanfangdata.com.cn/perio/detail.do?perio_id=ggykf" \t "/Users/doudou/Documents\\x/blank),2016,(4):119-120,122.

[3][吴发翔](http://www.wanfangdata.com.cn/details/detail.do?_type=perio&id=txsj201903207),[钱佳威](http://www.wanfangdata.com.cn/details/detail.do?_type=perio&id=txsj201903207),[刘江帆](http://www.wanfangdata.com.cn/details/detail.do?_type=perio&id=txsj201903207). 一种基于C5.0决策树算法的票房预测研究[J]. [科技广场](http://www.wanfangdata.com.cn/perio/detail.do?perio_id=kjgc" \t "/Users/doudou/Documents\\x/blank),2016,(4):186-192.