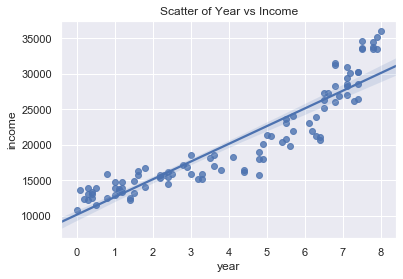
1、给定一个“IT行业收入表”（income.xlsx）,其中包含了不同工龄的薪水情况，请编写一个python程序，构建一个用于预测薪水的一元线性回归模型，绘制散点图进行模型可视化，并对模型进行评估。

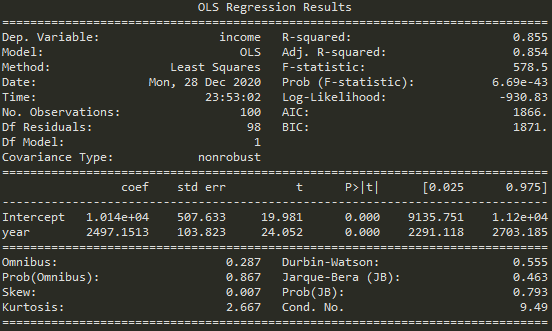
使用python的statsmodel库执行一元回归拟合，并使用seaborn库进行散点图绘制，代码如下：

1. # 数据导入
2. **import** pandas as pd
3. data = pd.read\_excel(r"C:\Users\Mac\Desktop\income.xlsx")
4. data.rename(columns={'工龄':'year', '薪水':'income'}, inplace=True)
6. # 一元线性回归
7. **import** statsmodels.formula.api as smf
8. result = smf.ols('income~year', data=data).fit()
10. # 绘制散点图
11. **import** matplotlib.pyplot as plt
12. **import** seaborn as sns
13. sns.set()
14. sns.regplot('year', 'income', data=data)
16. # 输出回归结果概述
17. **print**(result.summary())

输出如下散点图：



并有如下回归结果分析：



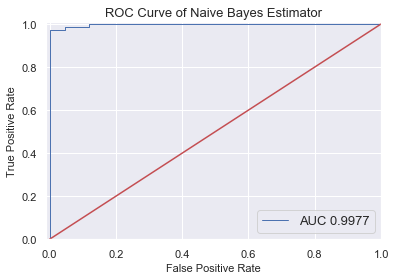
可以看到回归使用的是最小二乘法，并且没有使用正则项。回归效果方面，R值达到0.854，F检验的p-value小至6.69e-43，均表明了工龄与薪水的正相关性较强，说明了用工龄来预测薪水是可行的，散点图中的散点与拟合线较为贴合也印证了这一点。

2、 给定一个肿瘤数据表, 其中包含肿瘤的6个特征变量，分别是最大周长，最大凹陷度，平均凹陷度，最大面积，最大半径，平均灰度值。目标变量为肿瘤性质，0代表为恶性，1代表为良性。请编写一个python程序，按照8：2的比例来划分训练集和测试集，运用朴素贝叶斯模型来预测肿瘤为良性肿瘤还是恶性肿瘤，并绘制ROC图，对模型进行评估。

首先使用sklearn库中的StratifiedShuffleSplit类，把数据按肿瘤性质分层分割成训练集和测试集。然后使用GaussianNB类（对数据有正态性假设）对训练集进行拟合，得到一个朴素贝叶斯分类器。使用得到的分类器对测试集进行预测，并计算真阳性率和假阳性率绘制ROC曲线，代码如下：

1. # 数据导入
2. data = pd.read\_excel(r"tumour.xlsx")
4. # 分层抽样
5. **from** sklearn.model\_selection **import** StratifiedShuffleSplit
6. split = StratifiedShuffleSplit(test\_size=0.2)
7. **for** ind1,ind2 **in** split.split(data, data['肿瘤性质']):
8. X\_train, y\_train = data.iloc[:,:6].loc[ind1], data['肿瘤性质'].loc[ind1]
9. X\_test, y\_test = data.iloc[:,:6].loc[ind2], data['肿瘤性质'].loc[ind2]
11. # 朴素贝叶斯分类
12. **from** sklearn.naive\_bayes **import** GaussianNB
13. bayes = GaussianNB()
14. bayes.fit(X\_train, y\_train)
16. # 绘制ROC曲线
17. **from** sklearn.metrics **import** roc\_curve, roc\_auc\_score
18. **import** matplotlib.pyplot as plt
19. **def** plot\_roc\_curve(clf\_name, X, y):
20. clf = globals()[clf\_name]
21. **try**:
22. scores = clf.decision\_function(X)
23. **except**:
24. scores = clf.predict\_proba(X)[:, 1]
25. fpr, tpr, thresholds = roc\_curve(y, scores)
26. auc = roc\_auc\_score(y, scores)
27. label = "AUC " + str(round(auc,4))
28. plt.plot(fpr, tpr, linewidth=1, label=label)
29. plt.plot([0, 1], [0, 1], 'r')
30. plt.axis([-0.01, 1, 0, 1.01])
31. plt.xlabel('False Positive Rate', fontsize=11)
32. plt.ylabel('True Positive Rate', fontsize=11)
33. plt.legend(loc='best', fontsize=13)
34. plot\_roc\_curve('bayes', X\_test, y\_test)
35. plt.title("ROC Curve of Naive Bayes Estimator", fontsize=13)

得到如下ROC曲线图：



可以看到，曲线整体十分接近左上角，auc值几乎接近1，这就说明朴素贝叶斯分类在此场景下分类效果非常好。

3、 给定某国家国民生产总值平减指数（GNP deflator）时间序列如excel文件（data1.xlsx），请用python编写程序，完成如下功能：

(1) 判断该序列的平稳性；

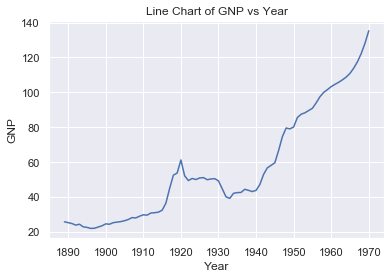
(2) 判断该序列是否为纯随机序列；

(3) 绘制该序列时序图、自相关图和偏自相关图；

(4) 采用合适的时间序列模型对上述序列进行拟合，并对未来10年的序列值进行预测。

1）使用seaborn库画出序列的折线图，代码和折线图如下：

1. # 数据导入
2. data = pd.read\_excel(r"data1.xlsx")
3. # 时序图
4. sns.lineplot('Year', 'GNP', data=data)
5. plt.title("Line Chart of GNP vs Year")

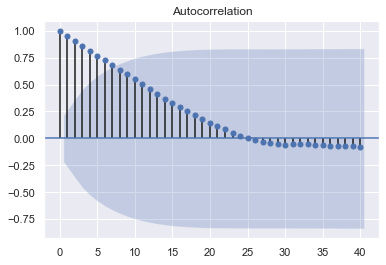


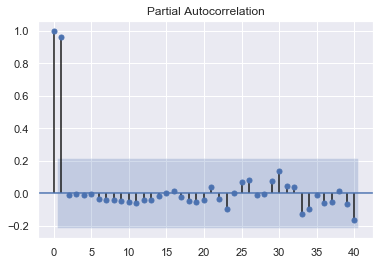
从折线图可以明显看到，GNP的均值大体上随着时间呈增长趋势，所以该序列不是一个平稳序列。

2）由于该序列不是平稳序列，所以也必然不是纯随机序列。

3）时序图已由1）给出，接下来使用statsmodel绘制自相关图和偏自相关图，代码和图如下：

1. # 绘制自相关图
2. **from** statsmodels.graphics.tsaplots **import** plot\_acf, plot\_pacf
3. plot\_acf(data['GNP'] , lags=40)
5. # 绘制偏自相关图
6. plot\_pacf(data['GNP'], lags=40)

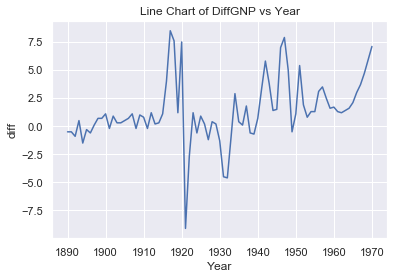




在自相关图中，横轴表示延迟期数，纵轴表示自相关系数，从图中可以看出自相关系数衰减到零的速度比较缓慢，在很长的延迟期内，自相关系数一直为正，然后为负，呈现出三角对称性。这是具有单调趋势的非平稳序列的一种典型的自相关图形式。

4）首先要先对非平稳序列做平稳性处理。下面使用差分法对序列进行处理：

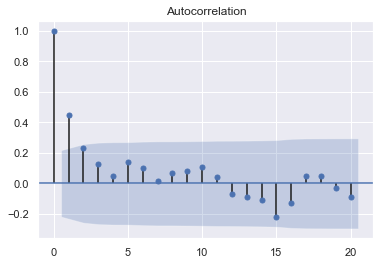
1. # 一阶差分
2. data['diff'] = data['GNP'].diff()
4. # 差分后时序图
5. sns.lineplot('Year', 'diff', data=data)
6. plt.title("Line Chart of DiffGNP vs Year")
8. # ADF检验
9. **from** statsmodels.tsa.stattools **import** adfuller
10. **print**("p-value:", adfuller(data['diff'].dropna())[1])

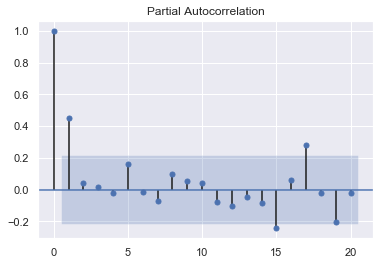




从差分处理后的时序图可以看出序列大体上在0轴附近上下波动，并且ADF检验的p-value远小于0.05，说明可以将差分序列视为平稳序列。重新绘制该差分序列的自相关图和偏相关图如下：

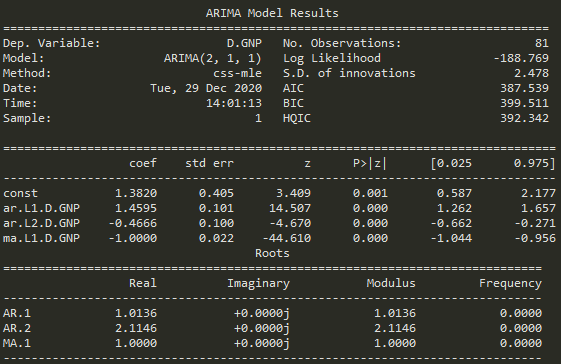
1. # 差分绘图
2. plot\_acf(data['diff'].dropna())
3. plot\_pacf(data['diff'].dropna())





可以看到，自相关图呈现出一个逐渐趋向0的拖尾，而偏自相关图在2阶后趋向于0。 综上，可以选择ARIMA（一阶差分，二阶自相关，一阶移动平滑）模型。使用statsmodel中的ARIMA模块对数据进行拟合：

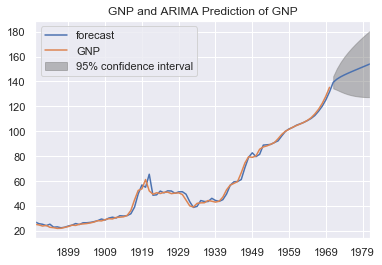
1. # ARIMA模型
2. **from** statsmodels.tsa.arima\_model **import** ARIMA
3. model = ARIMA(data['GNP'], order=(2, 1, 1))
4. arima = model.fit()
5. arima.summary()



模型结果显示，所有系数的p值均较小，表明所有系数均显著。

下面使用得到的模型对原数据进行拟合，并对未来十年的GNP做出预测，并整合绘制时序图：

1. # 绘制预测图像
2. arima.plot\_predict(1, 92)
3. plt.xticks(range(10,100,10), list(range(1899,1990,10)))
4. plt.title("GNP and ARIMA Prediction of GNP")



可以看到重预测部分与原始数据较为接近，证明了模型的合理性。

4、分析和理解SVR方法原理，通过查阅相关资料，简述求解SVR问题的SMO（Sequential Minimal Optimization，序列最小化算法）原理，并结合教材第6章案例，分析SVR方法有什么优点和缺点？如何改进SVR？

SVR原理：支持向量机的学习问题可以形式化为求解凸二次规划问题。这样的凸二次规划问题具有全局最优解，并且有许多最优化算法可以用于这一问题的求解。但是当训练样本容量很大时，这些算法往往变得非常低效，以致无法使用。所以诞生了一种快速实现算法——序列最小最优化(sequential minimal optimization)算法。SMO算法是一种启发式算法，其基本思路是：如果所有变量的解都满足此最优化问题的KKT条件，那么这个最优化问题的解就得到了，因为KKT条件是该最优化问题的充分必要条件；否则，选择两个变量，固定其他变量，针对这两个变量构建一个二次规划子问题。这个子问题关于这两个变量的解应该接近原始二次规划问题的解，因为这会使得原始二次规划问题的目标函数值变得更小。更重要的是，这时子问题可以通过解析方法求解，这样就可以大大提高整个算法的计算速度。子问题有两个变量，一个是违反KKT条件最严重的那一个，另一个由约束条件自动确定。如此，SMO算法将原问题不断分解为子问题并对子问题求解，进而达到求解原问题的目的。

整个SMO算法包括两个部分：求解两个变量二次规划的解析方法和选择变量的启发式方法。

SVR的优点：不仅适用于线性模型，非线性关系也能很好抓住；不需要担心多重共线性问题，可以避免局部极小化问题，提高泛化性能；由异常点引起的偏差更小。缺点：计算复杂度高，在面临数据量大的时候，计算耗时长。解决方法：可以使用LDA、PCA等降维方法预先对数据进行处理，或选择适当的核函数来避免维数爆炸问题。

5、阅读和理解教材11章中的“电子商务网站用户行为分析及推荐服务”综合案例，并成功运行该案例，分析运行结果，掌握协同过滤算法的原理。通过查阅相关文献，了解对协同过滤算法的改进方法。本题需要提交：

1）运行结果截图，包括准确率、召回率和F1三个指标截图

2）至少一种改进的协同过滤算法的描述(主要方法或流程等)