Assignment 2

作者: 付大为 学号: 2201110122

本次作业仍采用作业一使用的乳腺癌预测数据集。

1. 读取数据并将数据集划分为训练集和测试集，使用随机森林方法进行预测，并输出在测试集上的准确率，绘制ROC 曲线并获得AUC 指标。

导入第三方库并读取样本数据:

```
import pandas as pd
from sklearn.metrics import accuracy_score, roc_auc_score,
roc_curve
from sklearn.model_selection import train_test_split

df = pd.read_csv('../assignment1/breast_cancer.csv')
```

补全缺失数据(均值填充):

```
df.fillna({k: df.mean(skipna=True)[k] for k in df.columns if
df[k].isnull().any()}, inplace=True)
```

获取feature数据与label数据, 分别记为X和y:

```
X = df.iloc[:, :-1].values
y = df.iloc[:, -1].values
```

仍然按照assignment1中设置依据80% : 20%比例划分训练样本和测试样本:

```
X_train, X_test, y_train, y_test = train_test_split(X, y,
test_size=0.2)
```

搭建树数量为100的随机森林模型:

```
from sklearn.ensemble import RandomForestClassifier

model = RandomForestClassifier(n_estimators = 100)
```

拿训练集进行拟合:

```
model.fit(X_train, y_train)
```

计算在测试集上预测的准确率:

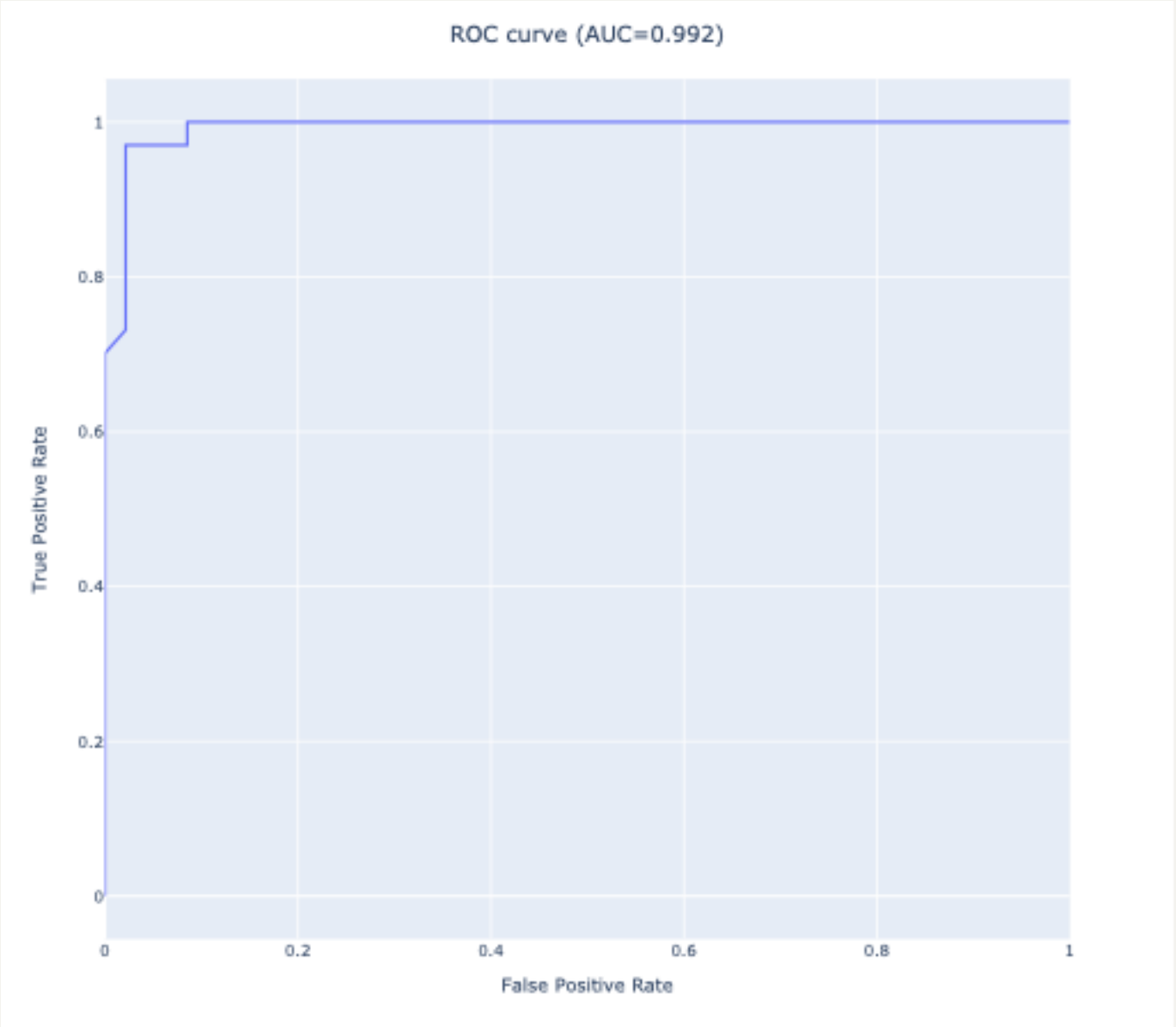
```
model.fit(X_train, y_train)
y_predict = model.predict(X_test)
print("ACCURACY OF THE MODEL: ", accuracy_score(y_test, y_predict))
# ACCURACY OF THE MODEL:  0.9649122807017544
```

绘制ROC曲线和计算AUC前需要计算随机森林模型在测试集上计算得到的y_score(这里我们定义为所有树中对某一样本预测为positive label的百分比)

```
y_predict_score = model.predict_proba(X_test)[:, 1]
fpr, tpr, thresholds = roc_curve(y_true=y_test,
y_score=y_predict_score)
```

接下来使用第三方库plotly画图

```
import plotly.express as px
fig = px.line(x=fpr, y=tpr)
fig.update_layout(
    width=600, height=500,
    title=dict(text="ROC curve (AUC=%.3f)"%roc_auc_score(y_test, y_predict_score), x=0.5, xanchor='center'),
    xaxis=dict(title_text="False Positive Rate"), yaxis=dict(title_text="True Positive Rate")
)
fig.write_image('./roc.pdf')
fig.show()
```



2. 采用决策树桩（深度为1 的决策树，即每次只采用一个特征的一个阈值作为划分依据）作为基本分类器，探究随基本分类器个数变化（1-100 个），随机森林在测试集上的准确率的变化，画出变化曲线并进行分析。

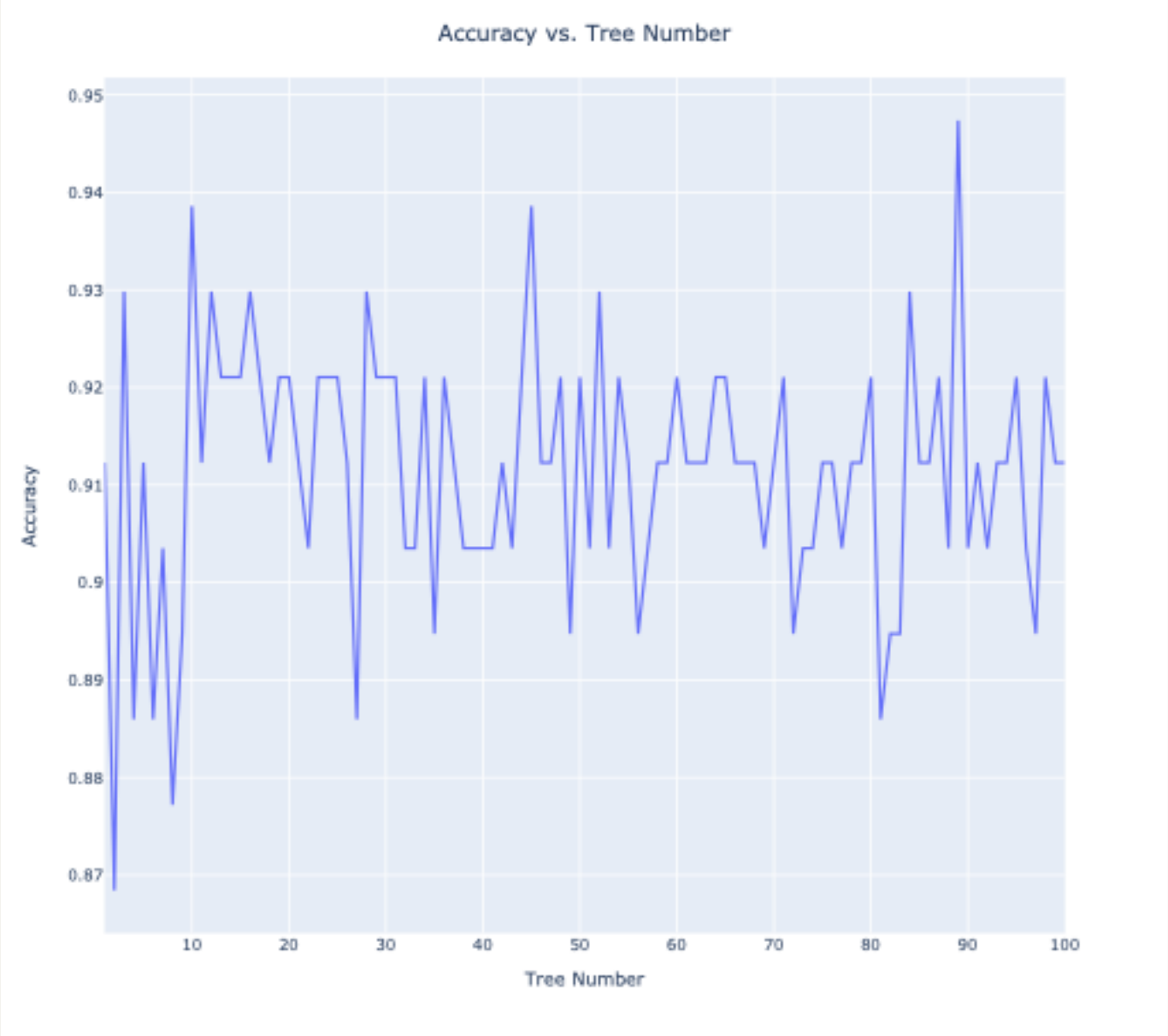
我们对模型中的树木数量从1循环到100, 在相同训练集上训练, 在同一测试集上预测, 并储存对应accuracy结果. (这里已经令树深度为1)

```
accuracy = []

for n in range(1, 101):
    n_tree_model = RandomForestClassifier(n_estimators=n,
max_depth=1)
    n_tree_model.fit(X_train, y_train)
    y_predict = n_tree_model.predict(X_test)
    accuracy.append(accuracy_score(y_true=y_test,
y_pred=y_predict))
```

对得到的准确率(Accuracy)与基本分类器个数(Tree Number)关系做可视化如下

```
fig = px.line(x=range(1, 101), y=accuracy)
fig.update_layout(
    width=900, height=800,
    title=dict(text="Accuracy vs. Tree Number", x=0.5,
xanchor='center'),
    xaxis=dict(title_text="Tree Number"),
yaxis=dict(title_text="Accuracy")
)
fig.write_image('./accuray.pdf')
fig.show()
```

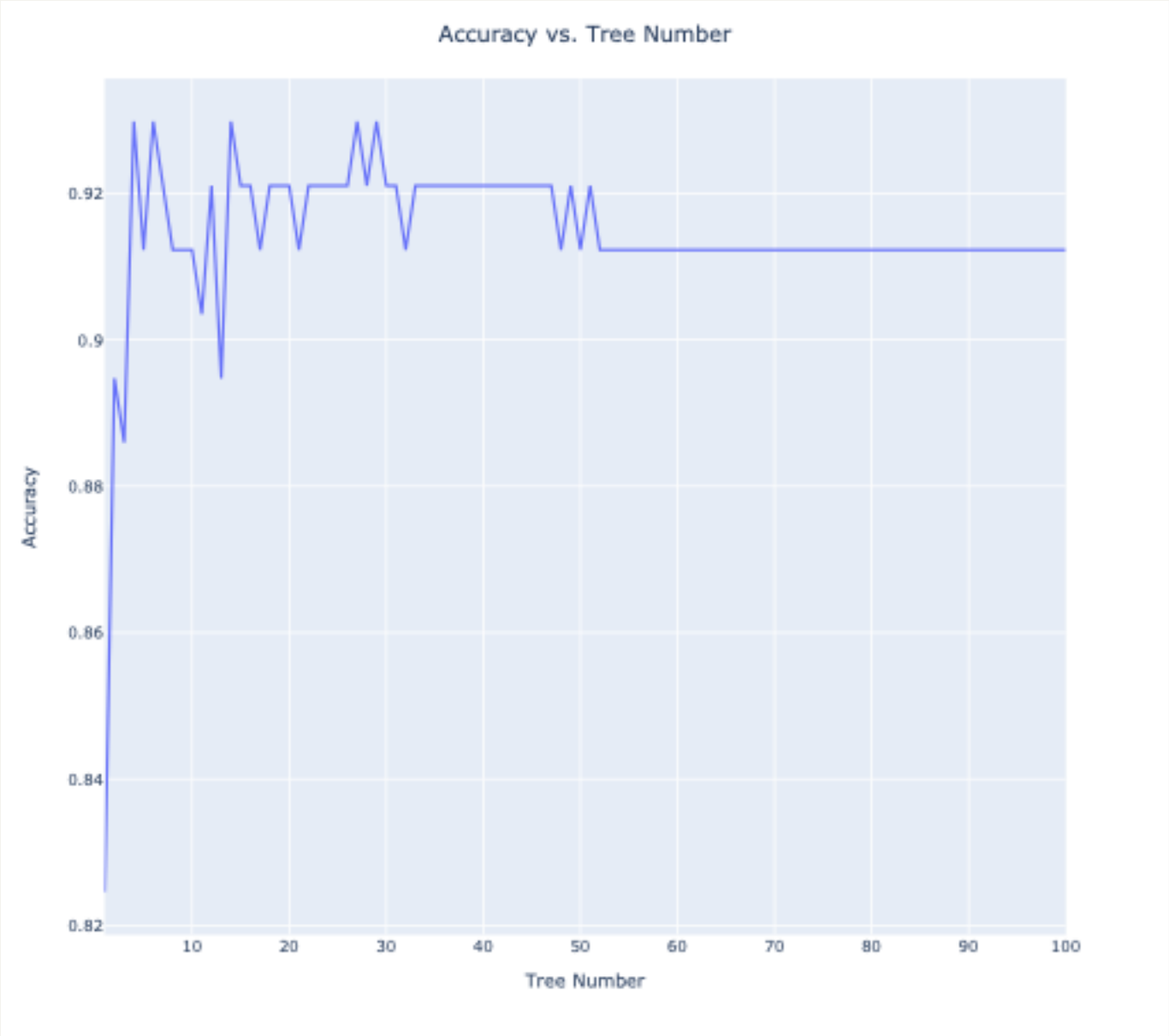


这里我们发现accuracy随树木数量的变化为, 从1开始先上升, 然后出现巨大波动, 这里我认为是因为每次构建模型时的random state不一致导致出现较大的随机误差, 所以我重新固定random state为同一值, 然后重新重复上述分析步骤, 新代码如下(区别是加了令random_state=9572):

```
accuracy = []

for n in range(1, 101):
    n_tree_model = RandomForestClassifier(n_estimators=n,
max_depth=1, random_state=9572)
    n_tree_model.fit(X_train, y_train)
    y_predict = n_tree_model.predict(X_test)
    accuracy.append(accuracy_score(y_true=y_test,
y_pred=y_predict))
```

可视化代码不变,得到结果如下



这里我们可以看到, 随着Tree Number上升, Accuracy先从0.91上升到0.96(在Tree Number=10附近达到最大, 即从欠拟合达到最佳拟合状态).

随后随着Tree Number上升进入饱和状态, 在Tree Number>60后甚至Accuracy略微下降, 可以认为是模型过大出现了过拟合现象.

代码实现在`assignment2.ipynb`文件中.