**Data Mining with Titanic dataset**

**Crisp-DM**  
1. **Business Understanding** เป็นช่วงของการทำความเข้าใจในธุรกิจ ในที่นี้คือ   คนที่มี Feature แบบไหนถึงจะมีชีวิตรอด  
2. **Data Understanding** เป็นช่วงทำความเข้าใจในข้อมูล โดยเป็นการทำความเข้าใจกับข้อมูลที่จัดเก็บ รวบรวมข้อมูล ศึกษาและทำความคุ้นเคยกับข้อมูล ตลอดจนประเมินคุณภาพของข้อมูลที่ได้มา

3. **Data Preparation Phase** ช่วงเตรียมข้อมูล โดยเตรียมข้อมูลดิบที่จะต้องใช้ในขั้นตอนที่เหลือ ตลอดจนเลือกตัวแปรที่ต้องการมาวิเคราะห์ให้เหมาะสม อีกทั้งแปลงรูปแบบของตัวแปรให้อยู่ในรูปแบบเดียวกัน เพื่อให้ข้อมูลพร้อมสำหรับการนำไปใช้ไปสร้างแบบจำลอง

4. **Model Phase** ช่วงสร้างแบบจำลองโดยเป็นการคัดเลือก แบบจำลองที่เหมาะสม ปรับปรุงตัวแปร ลักษณะแบบจำลองเพื่อให้ได้ผลลัพธ์ที่ดีที่สุด อาจจะใช้เทคนิคหลายๆเทคนิคเข้ามาช่วยวิเคราะห์ได้ ถ้าจำเป็น ก็สามารถย้อนกลับไปช่วงการเตรียมข้อมูล เพื่อเตรียมข้อมูลให้เหมาะสมกับการสร้างแบบจำลองใหม่ได้

5. **Evaluation Phase** ช่วงประเมินผล โดยเป็นการประเมินแบบจำลองที่ใช้ในการวิเคราะห์ทั้งหมด ประเมินว่าแบบจำลองไหนตอบโจทย์ในขั้นตอนแรกได้ดีที่สุด ตรวจสอบความถูกต้องและสภาพแวดล้อมต่างๆ ตัดสินใจในการนำผลลัพธ์ไปใช้

6. **Deployment Phase** โดยเป็นการนำไปใช้งานจริง รวมถึงนำเสนอตัวอย่างจากการนำไปใช้จริง

**Crisp-DM with Titanic dataset**

**1.1. Business Understanding**

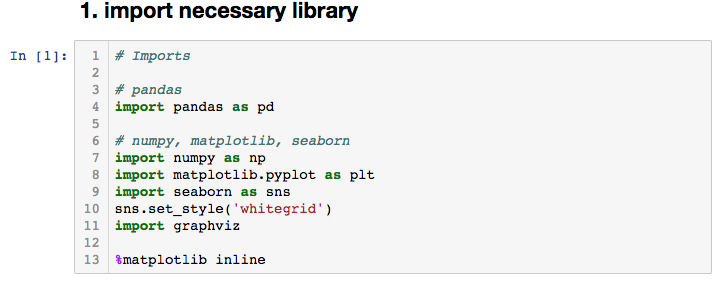
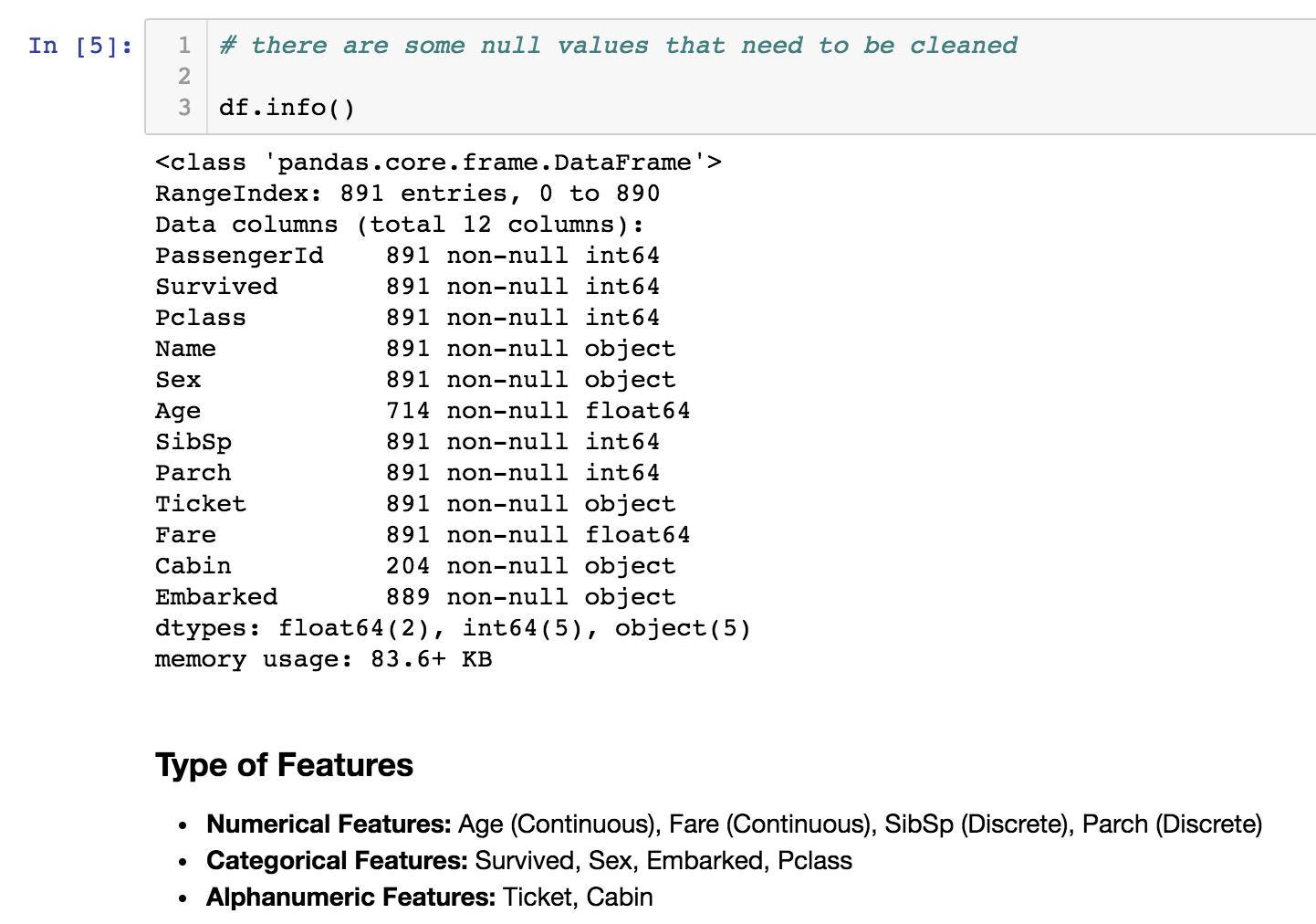
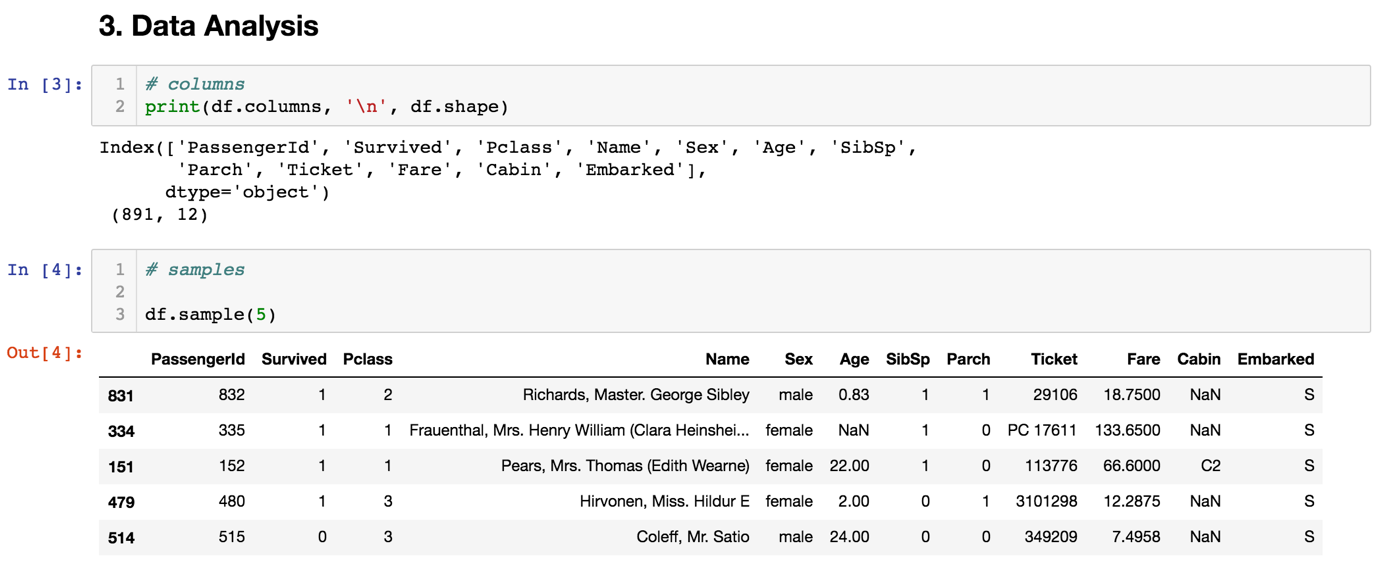
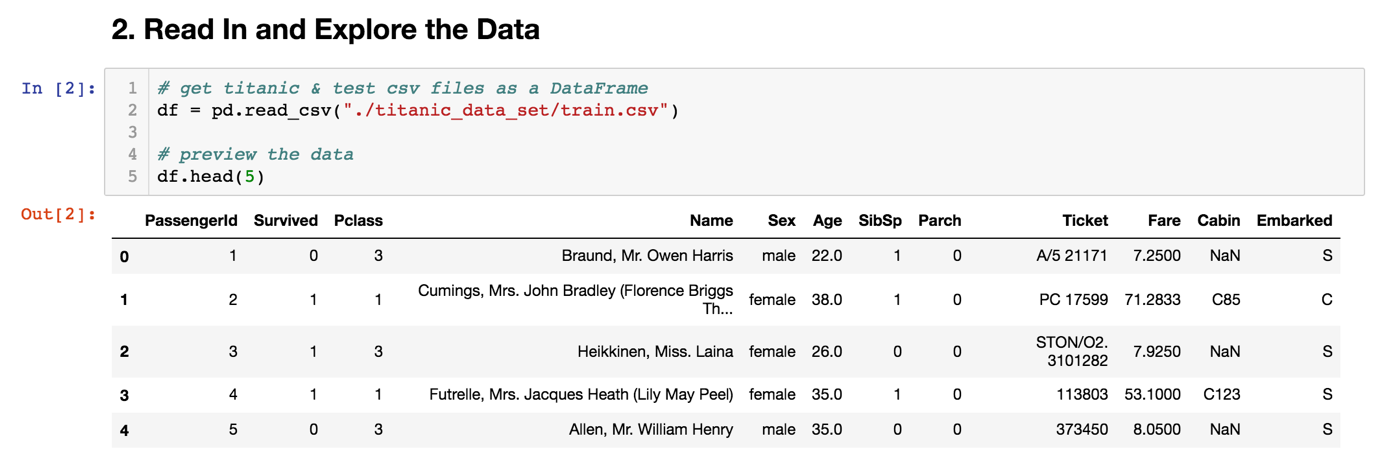
ให้แบ่งกลุ่มของคนที่มีอยู่ใน dataset ว่าลักษณะแบบไหนถึงจะมีชีวิตรอด หรือ ไม่มีชีวิตรอด

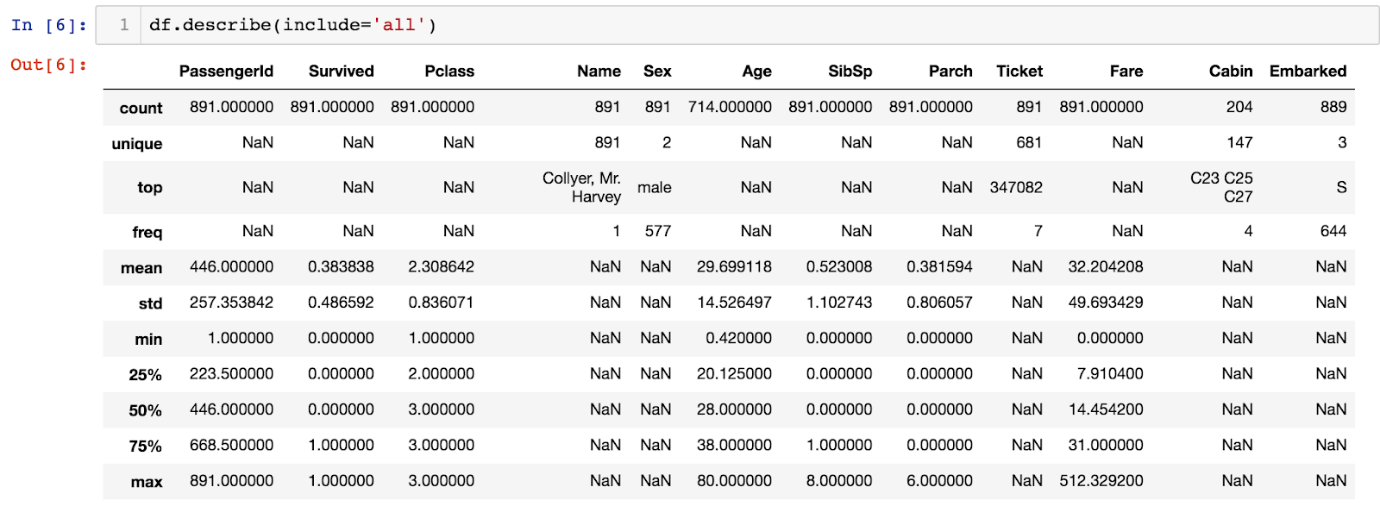
**1.2   Data Understanding**

import csv file ซึ่งเป็นเซทของข้อมูล ไททานิค โดยใช้ python tools ในการทำเหมืองข้อมูล พบว่ามีทั้งหมด 891 แถว และ มี 12 คอลัมน์ โดยข้อมูลแต่ละแถวคือ ข้อมูลของผู้โดยสารเรือแต่ละคน  และข้อมูลของแต่ละคอลัมน์มีดังต่อไปนี้

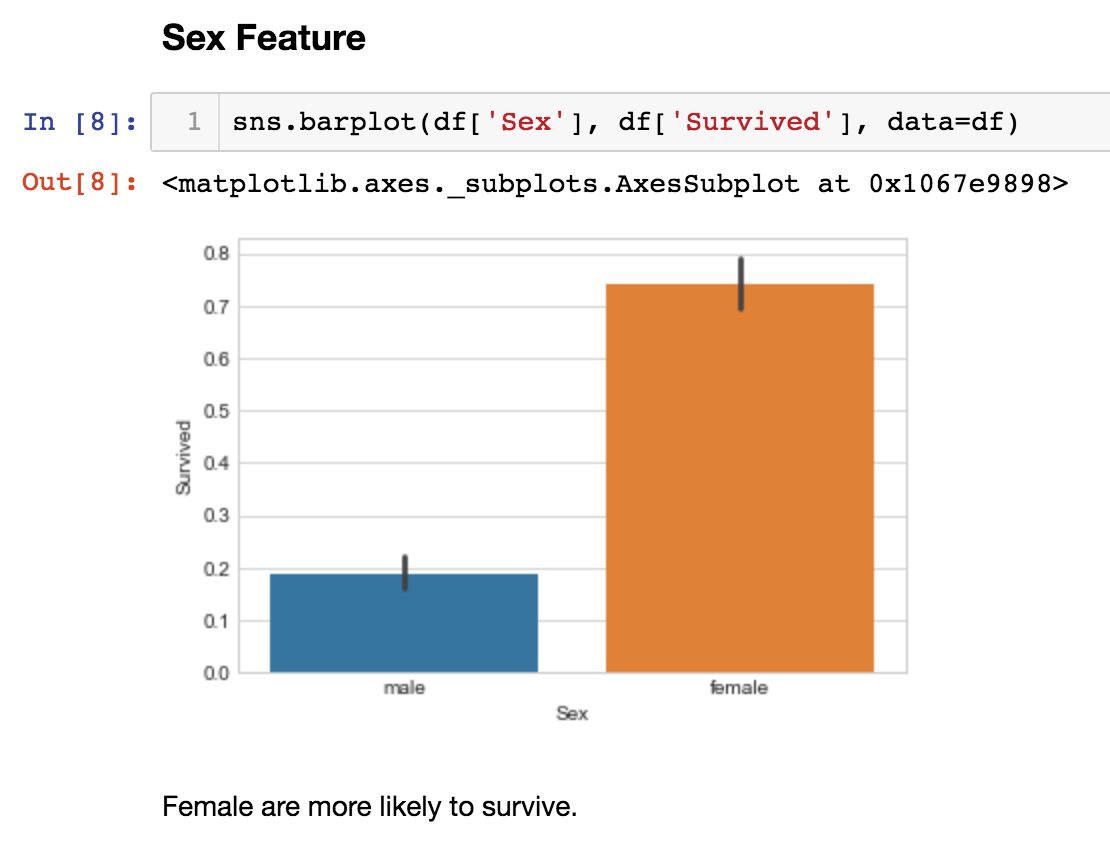
1. 'PassengerId' คือ หมายเลขผู้โดยสาร
2. 'Survived' คือ ผู้รอดชีวิต โดย 1 เป็นการรอดชีวิต ส่วน 0 คือผู้เสียชีวิต
3. 'Pclass' คือ สถานะภาพทางเศรษฐกิจและสังคม แบ่งได้เป็น การศึกษา + อาชีพ + รายได้
4. 'Name' คือ ชื่อของผู้โดยสาร
5. 'Sex' คือ เพศ
6. 'Age' คือ อายุ
7. 'SibSp' คือ จำนวนความสัมพันธ์ของผู้โดยสารที่มาด้วยกัน เช่น พี่น้อง หรือ คู่สมรส
8. 'Parch' คือ จำนวนความสัมพันธ์ของผู้โดยสารที่มาด้วยกัน เช่น พ่อ แม่ ลูก
9. 'Ticket' คือ หมายเลขตั๋ว
10. 'Fare' คือ ค่าโดยสาร
11. 'Cabin' คือ หมายเลขห้องโดยสาร
12. 'Embarked' คือ ท่าเรือที่รับผู้โดยสารขึ้นเรือ

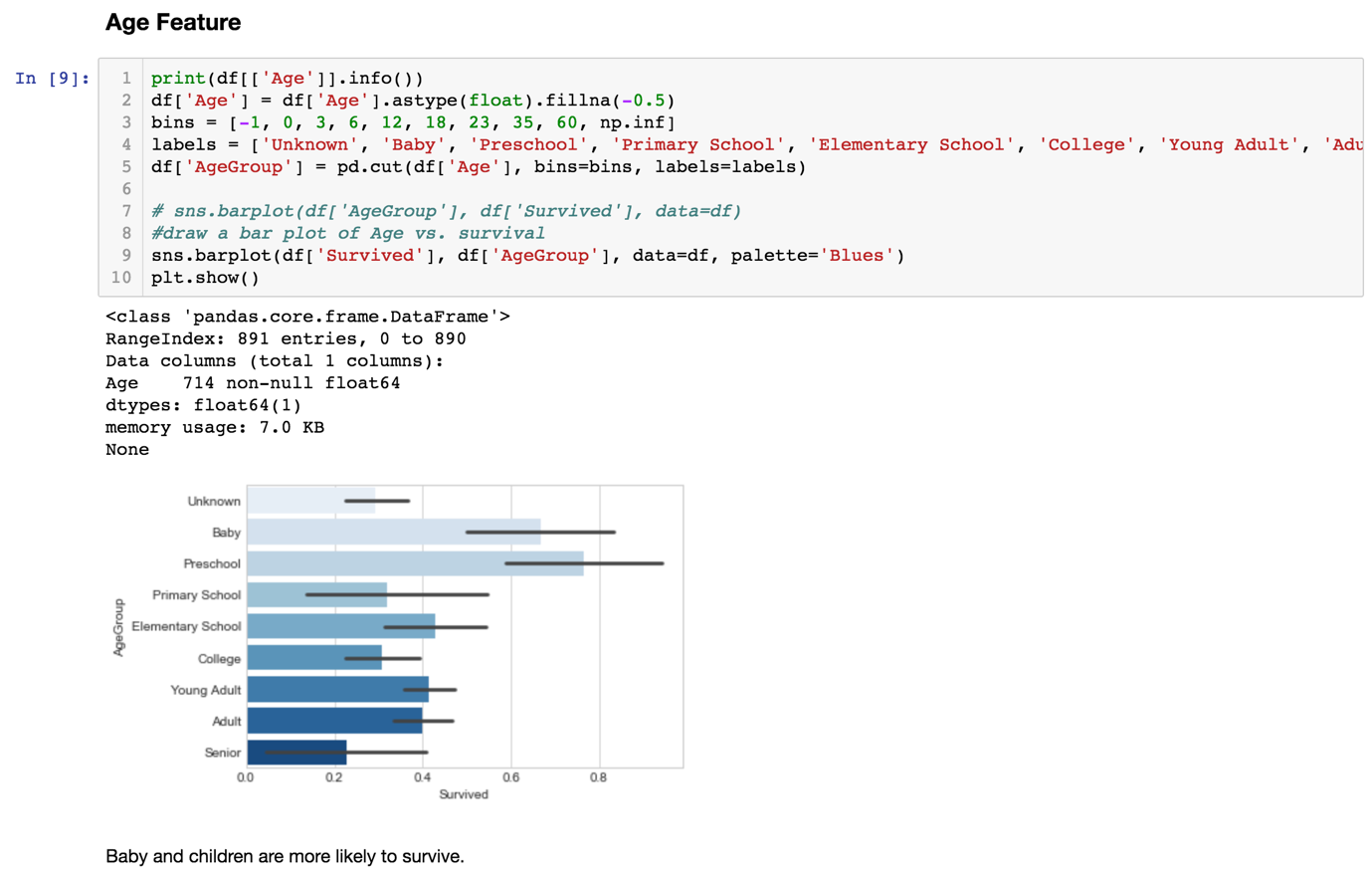
รายละเอียดจะเพิ่มเติมให้รูปต่อไป

****

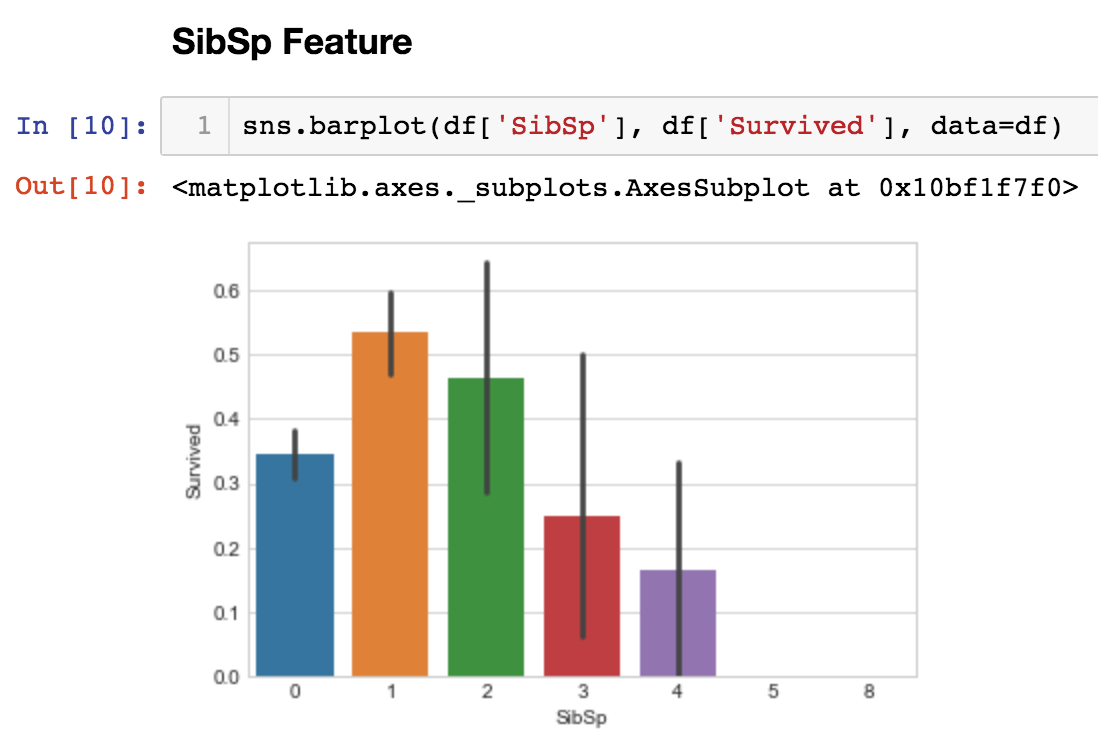


**Pclass Feature** ผู้ที่รอดชีวิตส่วนใหญ่คือ กลุ่มคนที่ไม่ได้มาเป็นครอบครัว

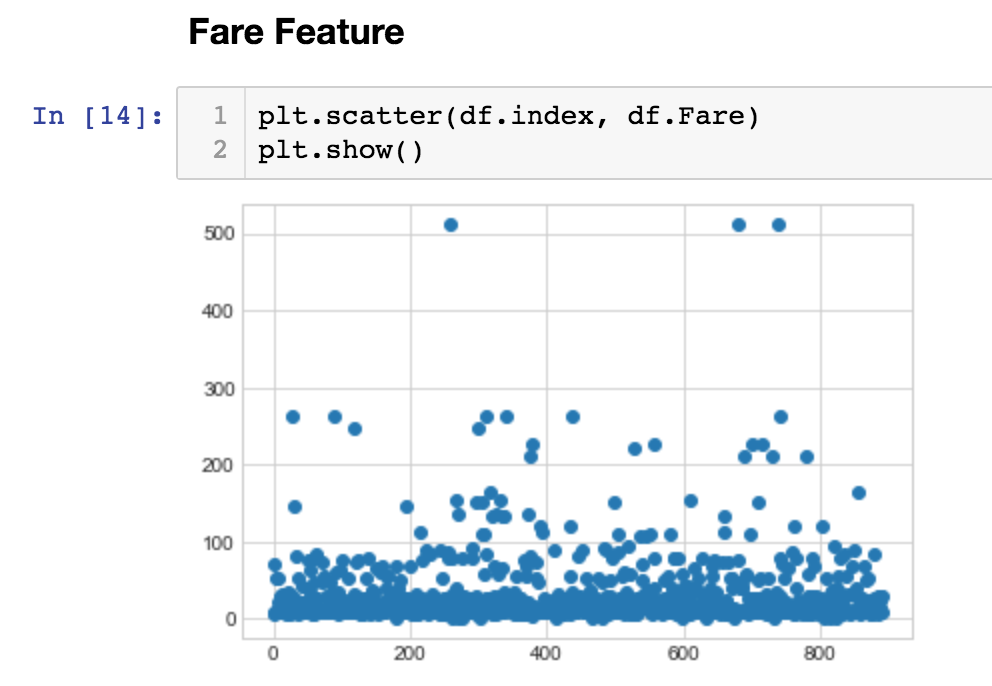


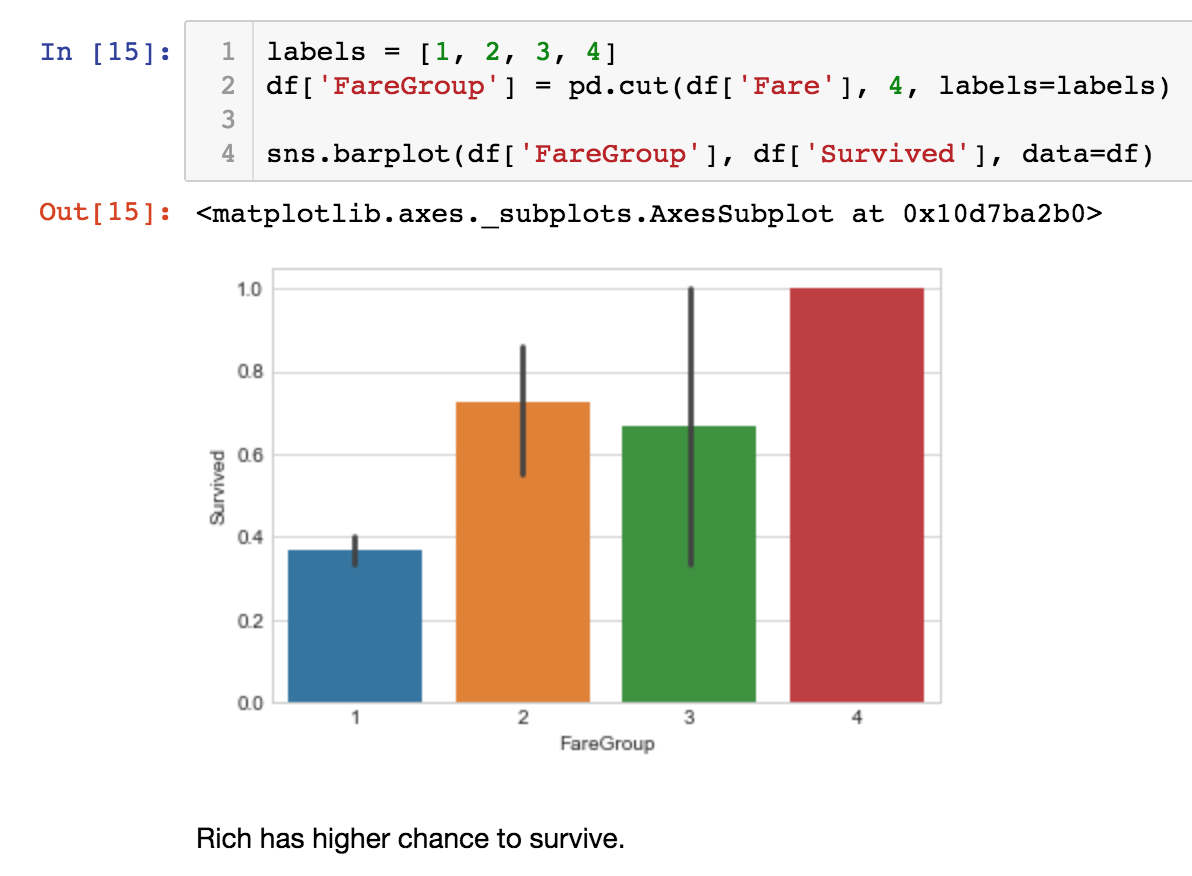
**Sex Feature** ผู้ที่รอดชีวิตส่วนใหญ่คือ ผู้หญิง

**Age Feature** ผู้ที่รอดชีวิตส่วนใหญ่คือ เด็ก

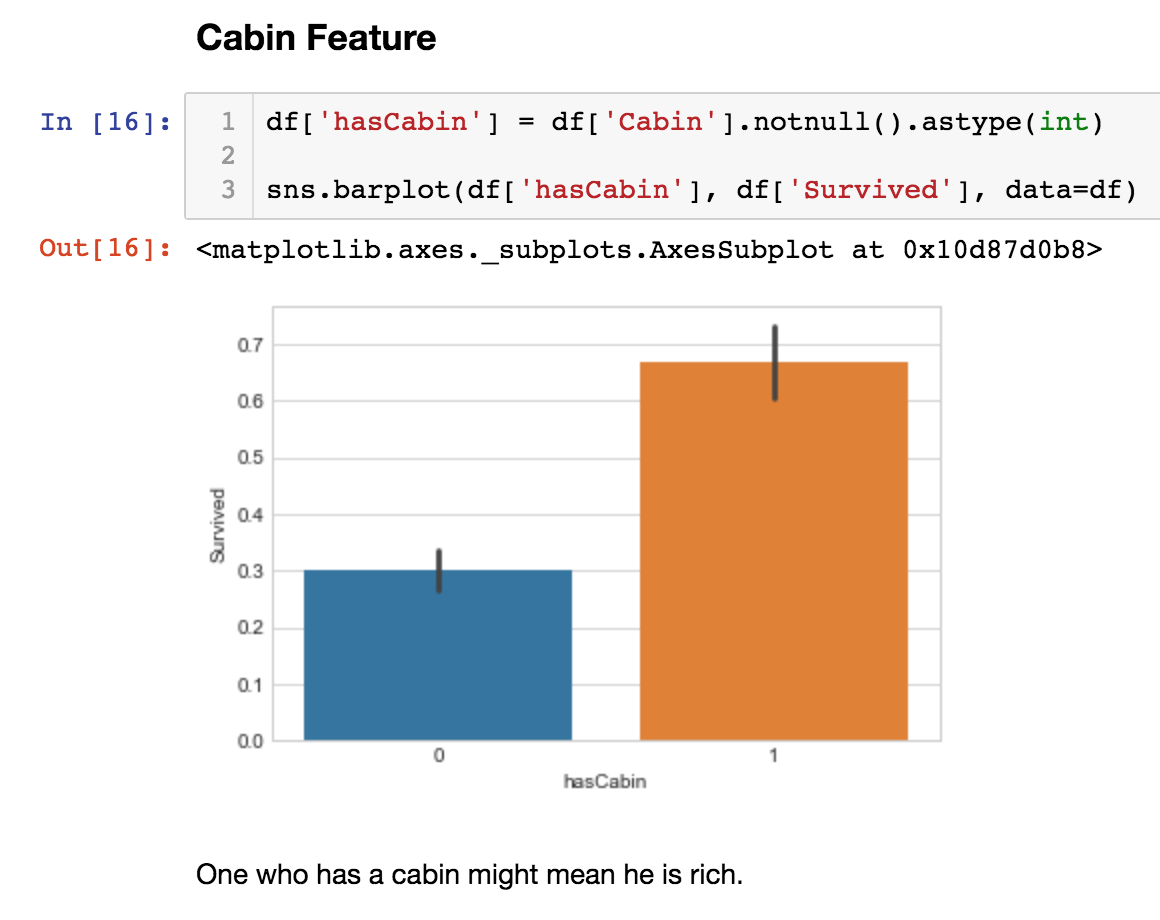


**SibSp Feature** ผู้ที่รอดชีวิตส่วนใหญ่คือ คนที่มาเป็นคู่จะมีโอกาสรอดสูงกว่า

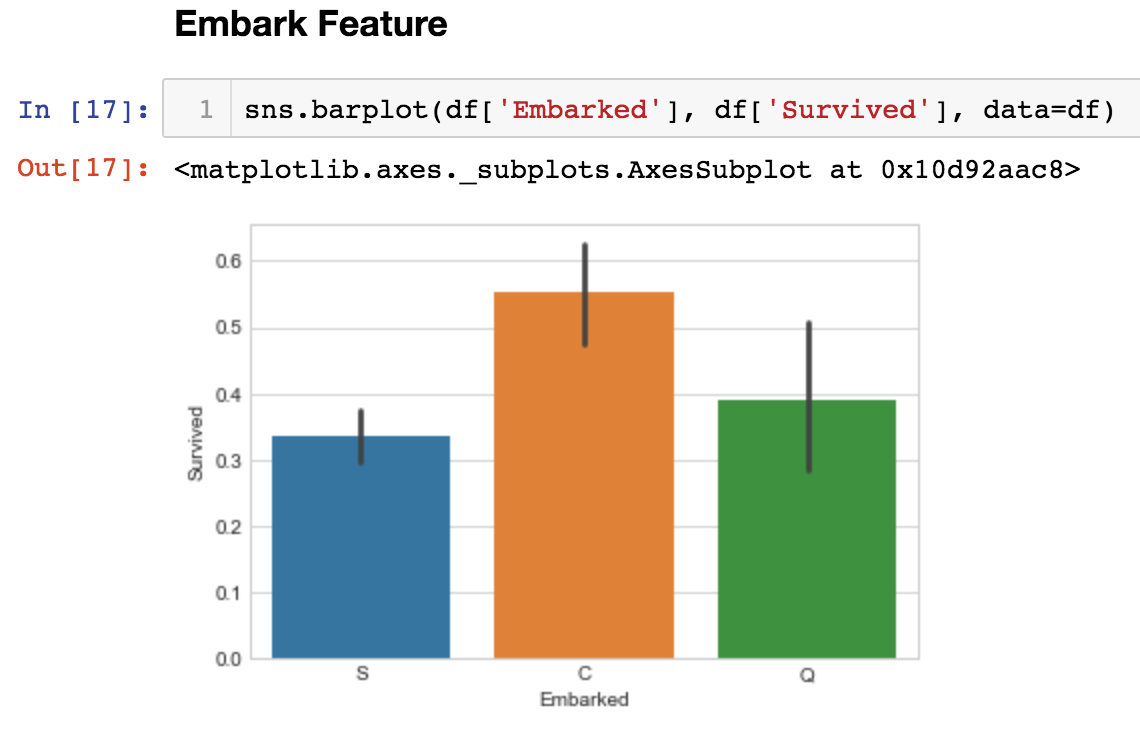




**Fare Feature** ผู้ที่รอดชีวิตส่วนใหญ่มาจากค่าโดยสารแพงมากสุด



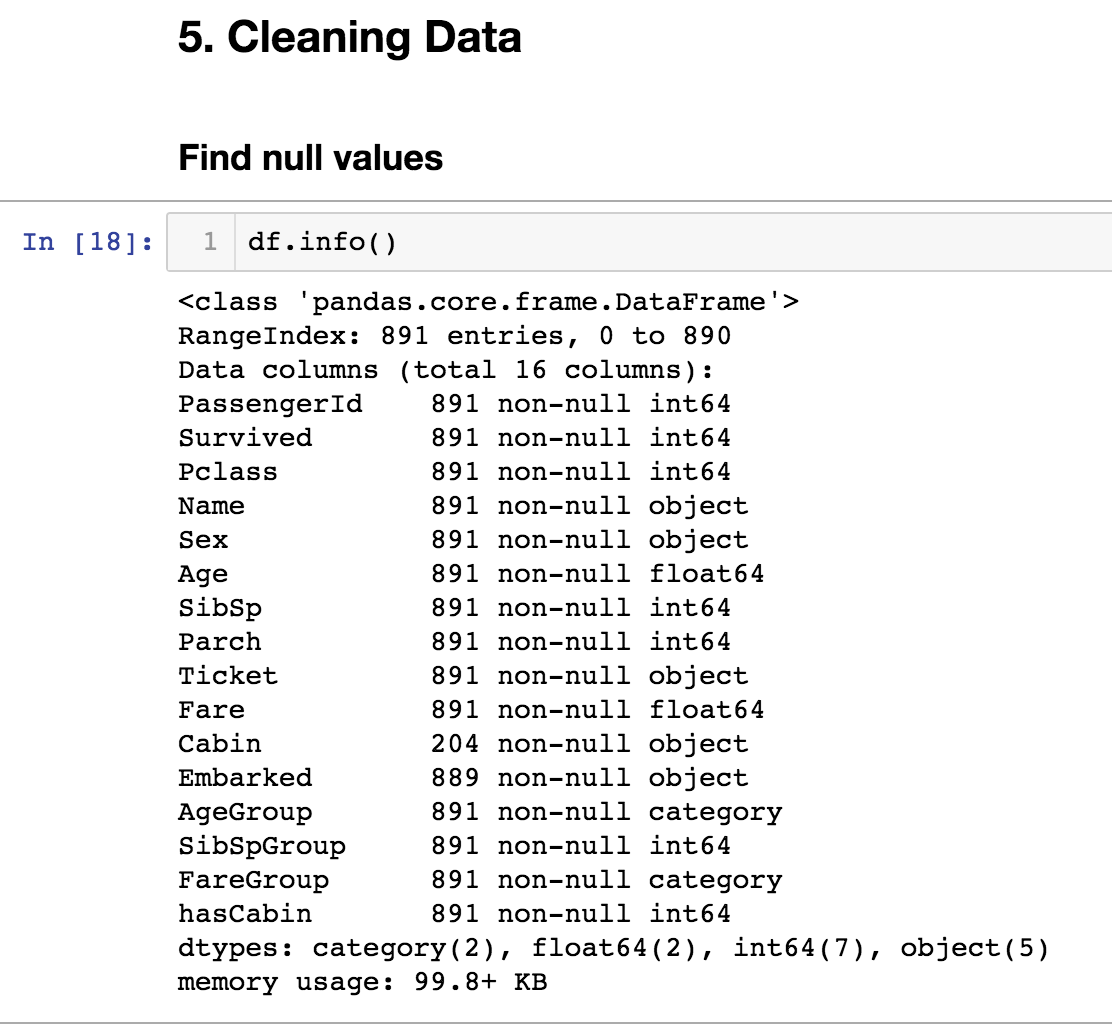
**Cabin Feature** ผู้ที่รอดชีวิตส่วนใหญ่คือ ผู้ที่มีหมายเลขห้องโดยสาร

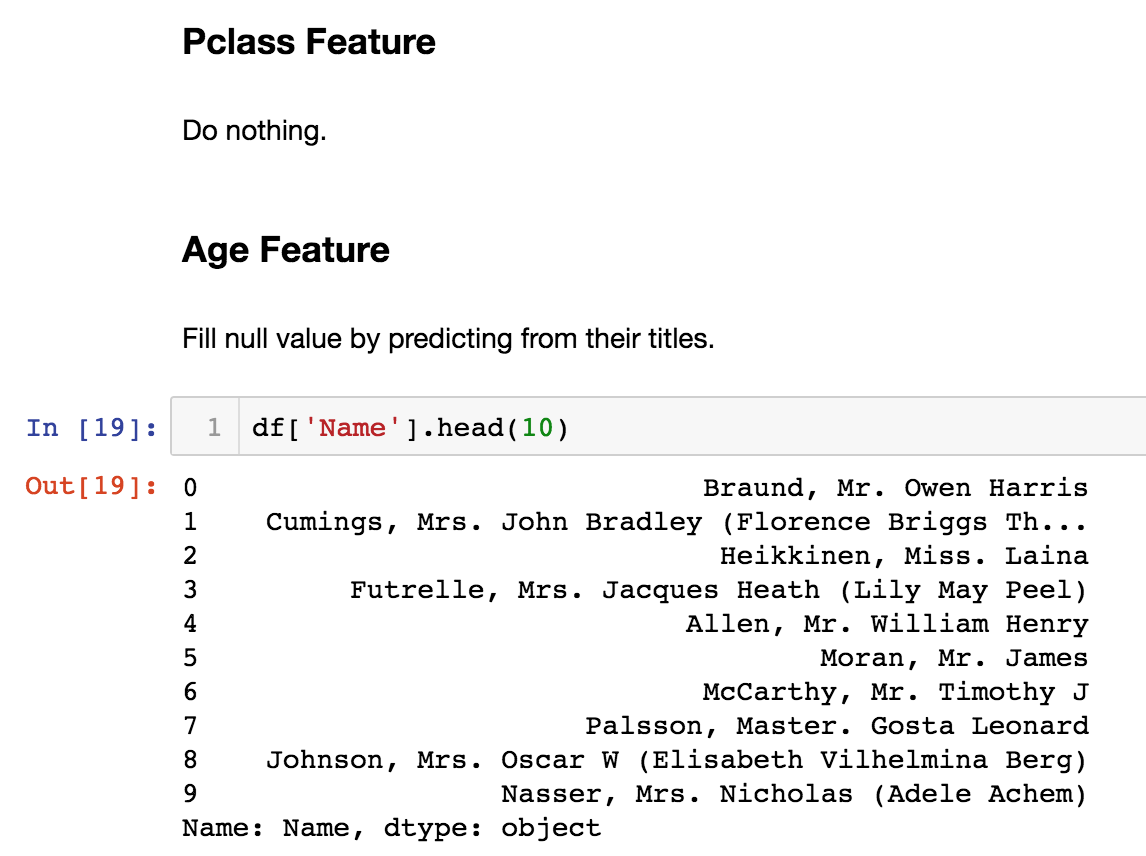


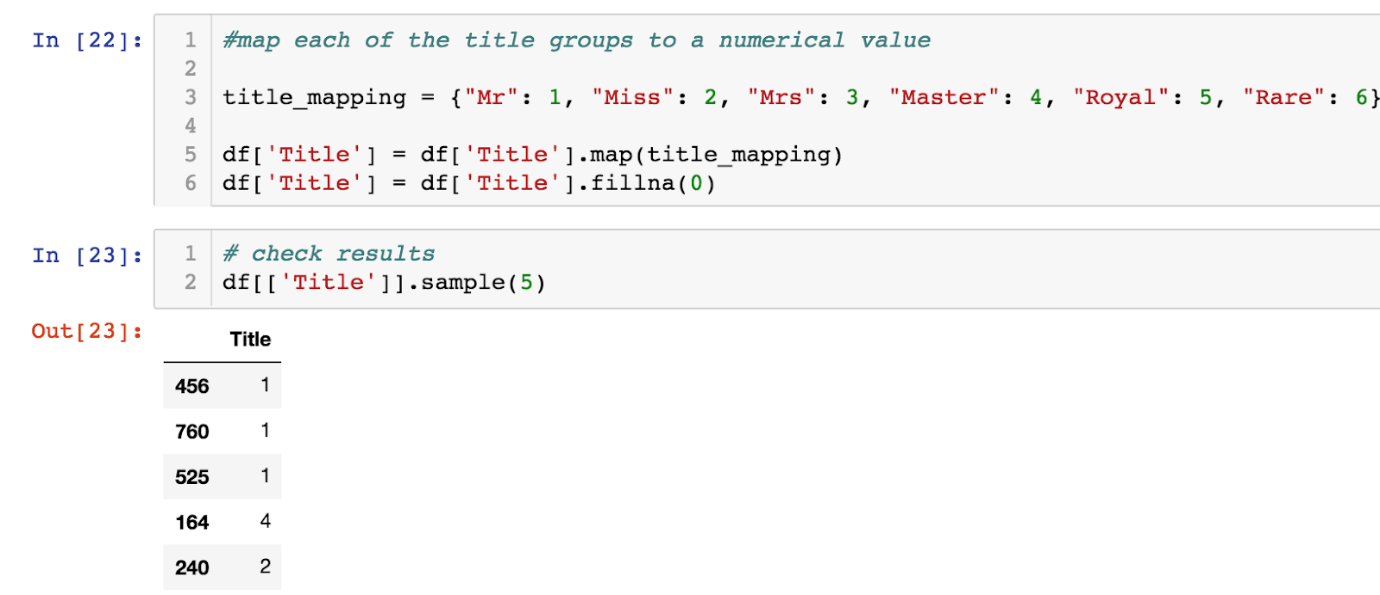
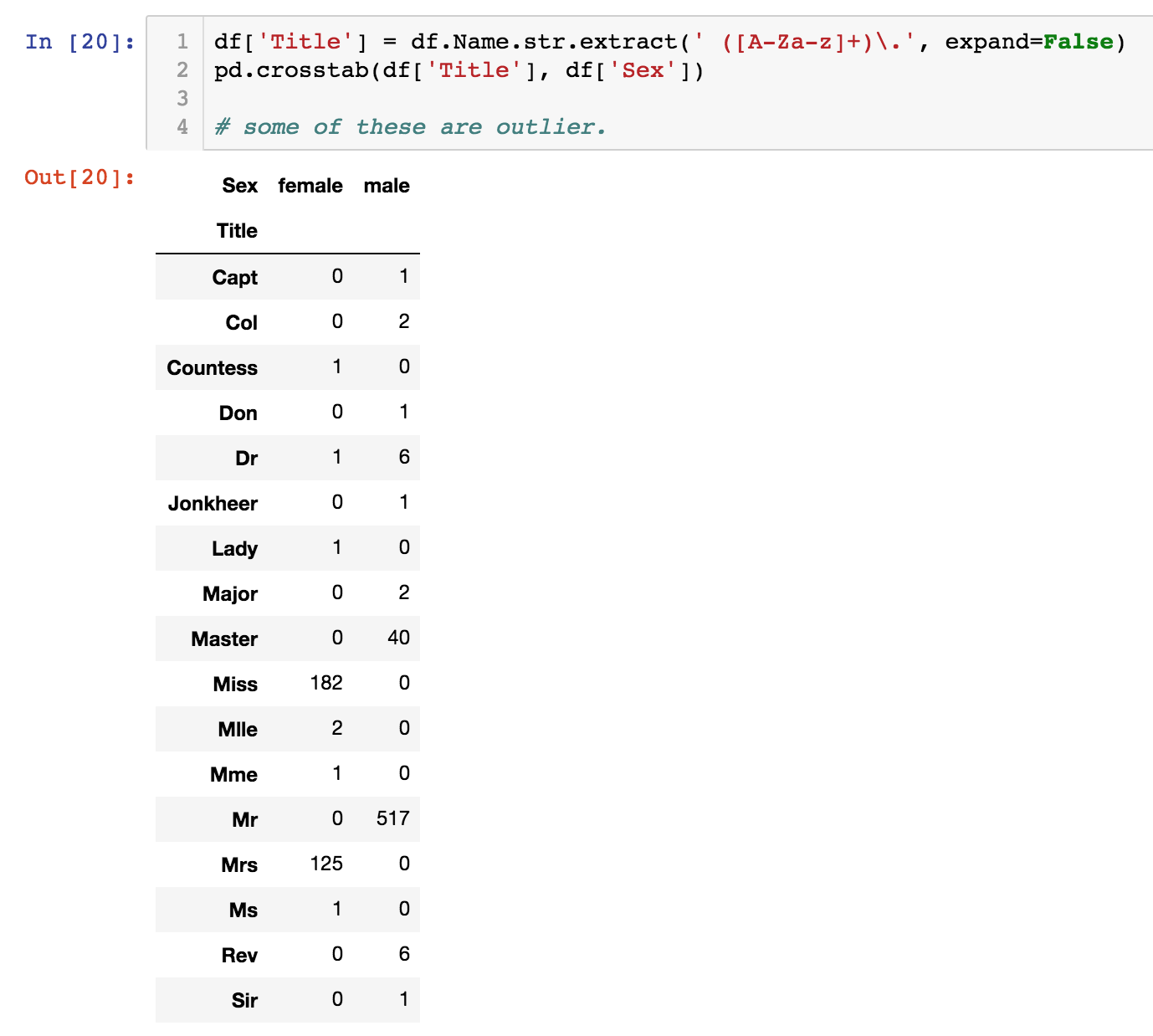
**Embark Feature** ผู้ที่รอดชีวิตส่วนใหญ่มาจากท่าเรือ Cherbourge

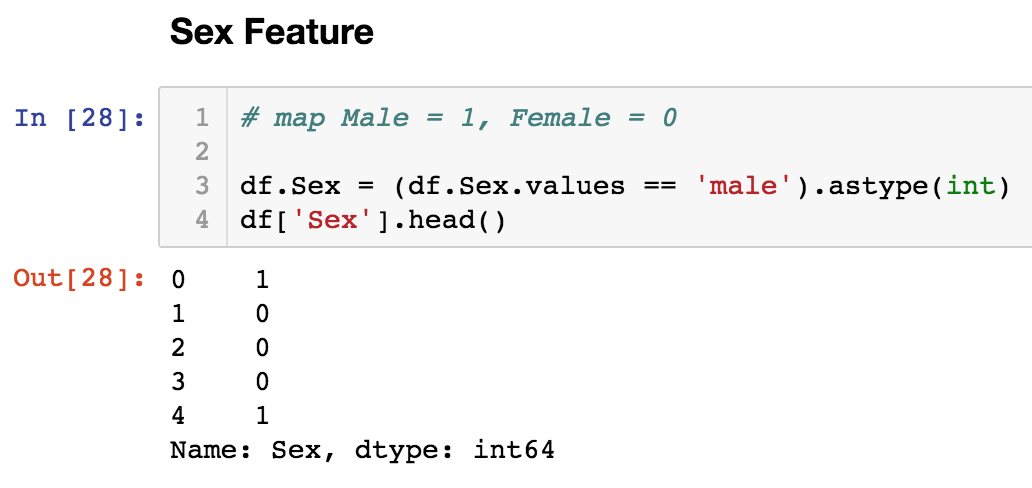
**1.3 Data Preparation Phase**

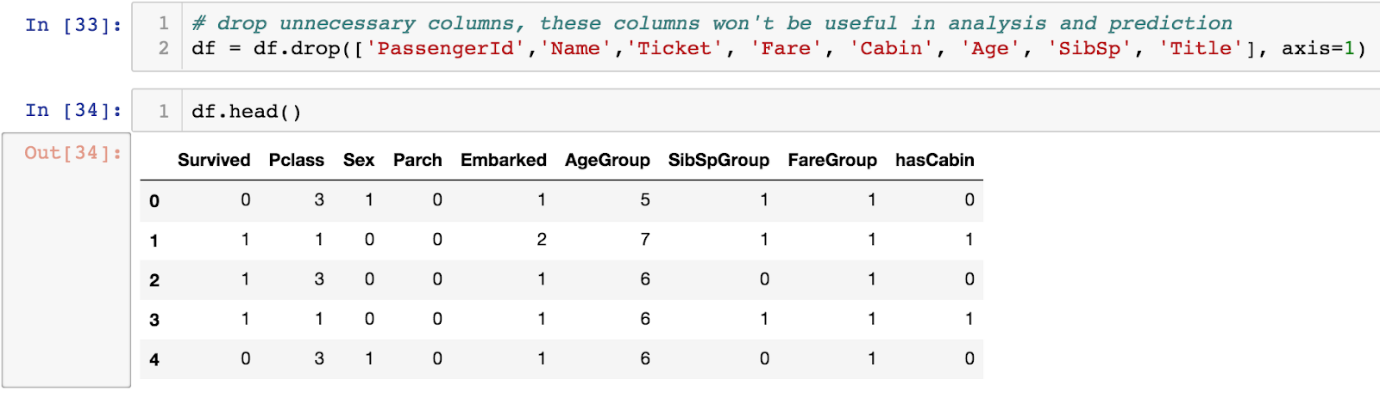
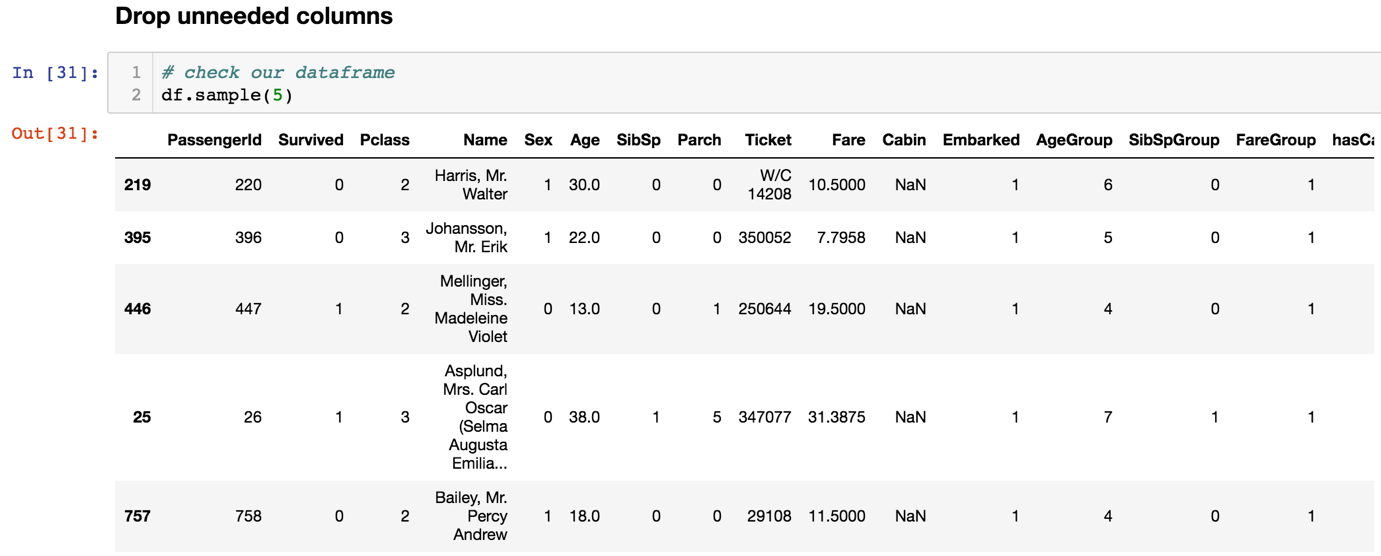
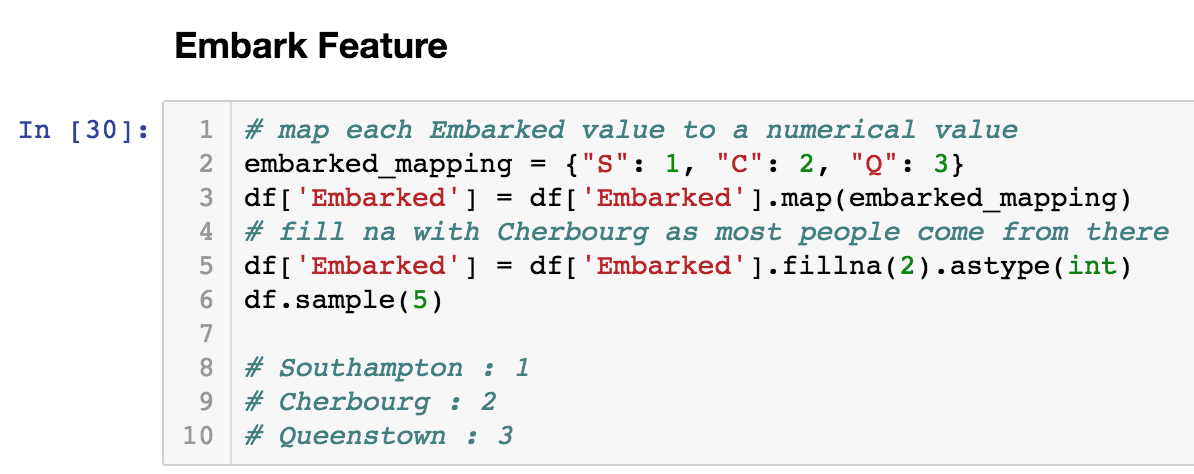
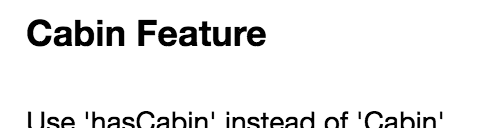
Cleaning data จากการกำจัดค่า null และจัดให้ข้อมูลอยู่ในรูปของตัวเลข integer เพราะ library python รับแต่ค่า integer





**Age** เติมอายุที่หายไปโดยทำนายจากชื่อนำหน้าว่าควรจะอยู่ในกลุ่มคนอายุเท่าไหร่

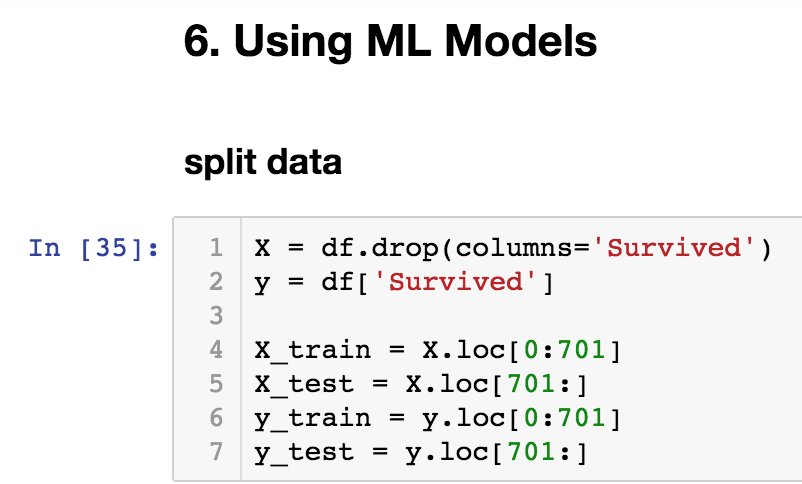
Sex กำหนดให้เพศหญิงเป็น 0 และเพศชายเป็น 1  
  


SibSp, Parch ไม่ต้องทำอะไรเพราะเป็นเลขอยุ่แล้ว  
  
  
fare feature จัดกลุ่มค่าโดยสารเป็น 4 กลุ่ม   
  
**cabin feature** จัดกลุ่มว่ามีห้องโดยสารว่ามีหรือไม่มี โดยมีห้องโดยสารมีโอกาสรอดสูงกว่า  


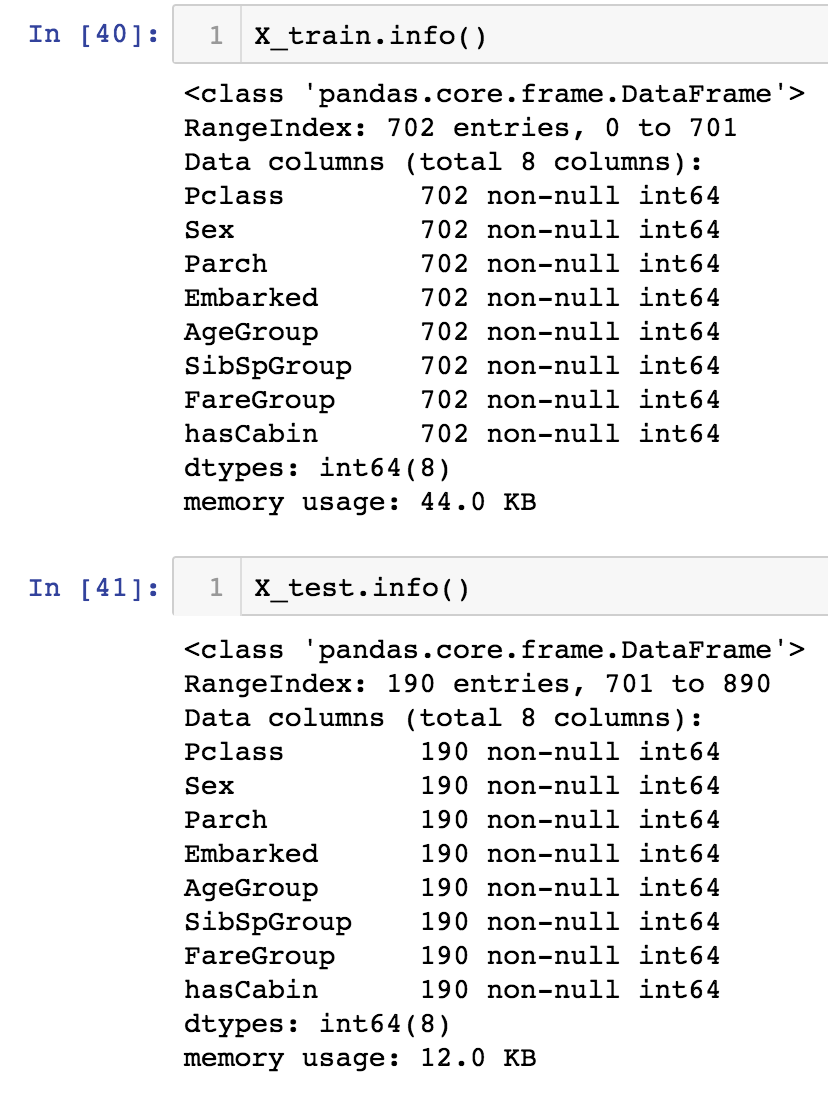
**1.4 Model Phase**

การแบ่งข้อมูลในการเทรนตั้งแต่คนแรก - คนที่ 700 ส่วนที่เหลือเป็นค่าในการนำไปทดสอบ โดยแบ่งข้อมูลออกเป็น

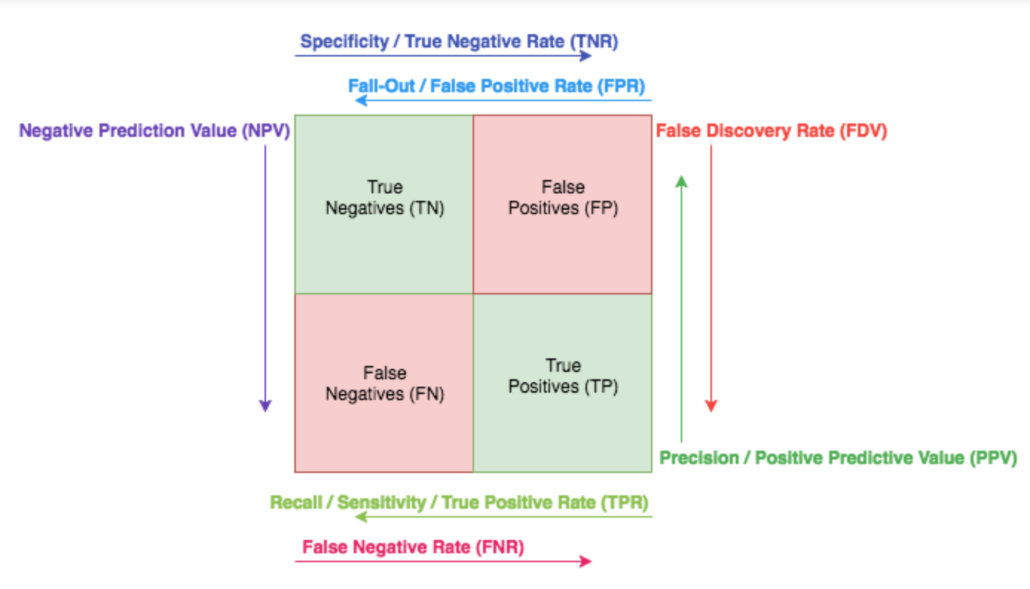
1. กลุ่ม x\_train คือ คอลัมน์ทั้งหมดยกเว้นคอลัมน์Survived ที่จะนำมาเทรน
2. กลุ่ม x\_test คือ คอลัมน์ทั้งหมดยกเว้นคอลัมน์Survived ที่จะนำมาทดสอบ
3. y\_train คือ คอลัมน์Survivedเท่านั้น ที่จะนำมาเทรน
4. y\_test คือ คอลัมน์Survivedเท่านั้น ที่จะนำมาทดสอบ

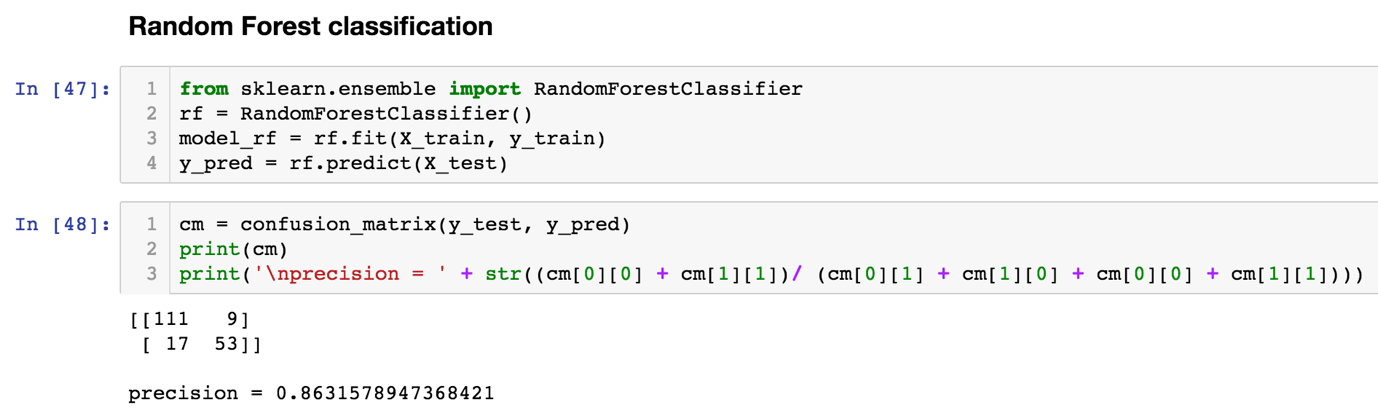
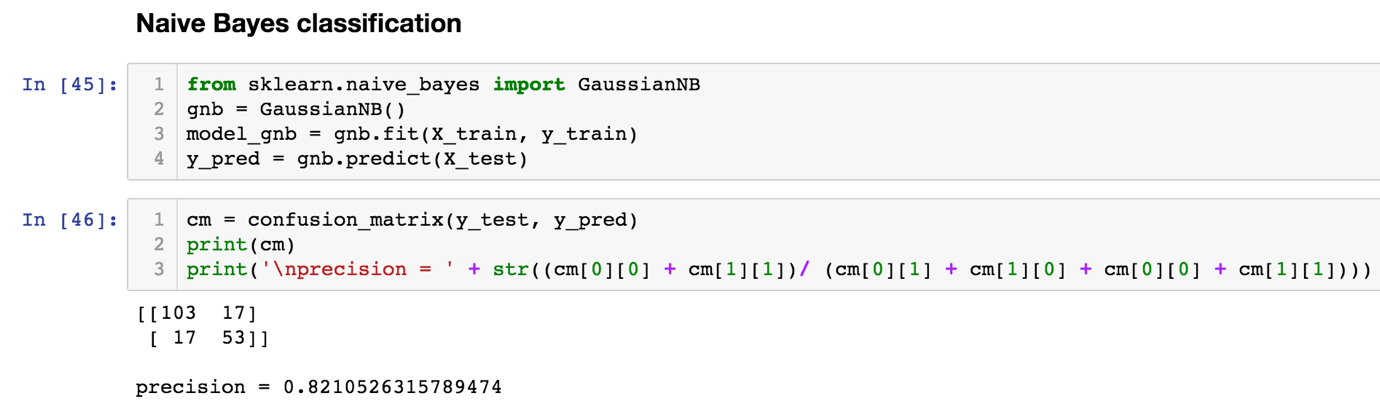


เช็คข้อมูลว่าไม่มี null และเป็น integer ทุกคอลัมน์



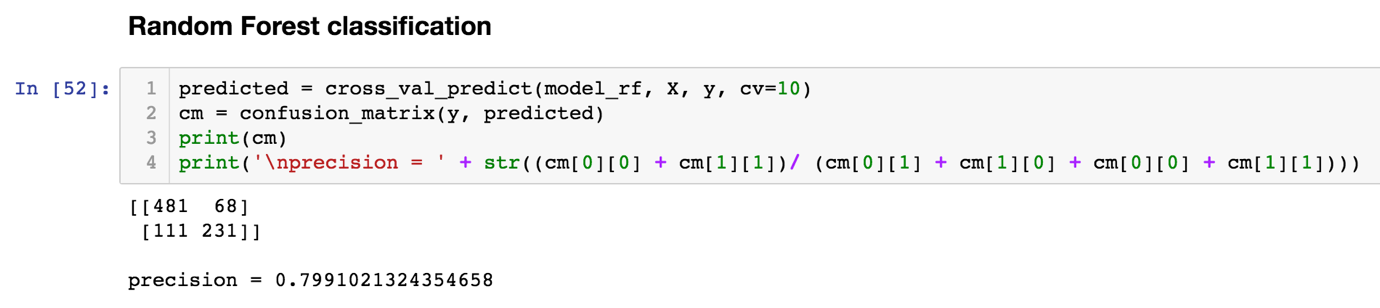
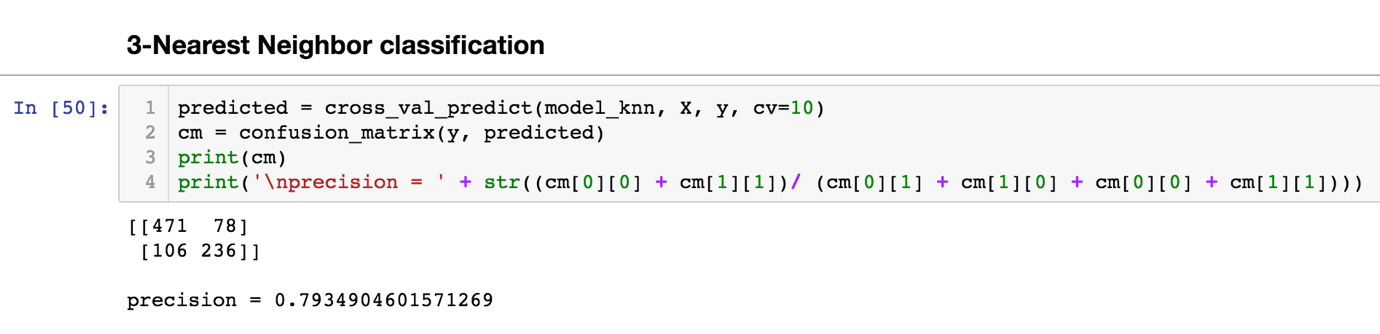
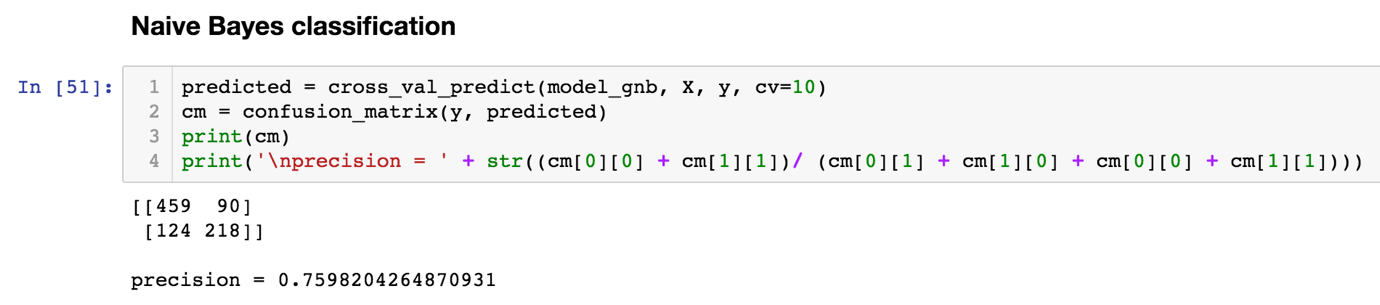
Confusion matrix เพื่อเทียบผลการทดลอง





หลังจากใช้ 3NN, Naïve Bayes, Random forest พบว่า Random forest มีความแม่นยำสูงที่สุดที่ 86.316% ทีนี้ลองใช้ 10 fold-cv ดูบ้าง





5. **Evaluation Phase** หลังจากการใช้โมเดล พบว่าการแบ่งข้อมูลธรรมดามีความแม่นยำสูงกว่าแบบ 10 fold cv ประมาณ 10% แต่อาจจะเกิด over fitting เราจึงเลือกใช้การแบ่งข้อมูลแบบ 10 fold cv โดยโมเดลที่ให้ค่าความแม่นยำสูงสุดคือ Random Forest classification ที่ 79.91% แต่ความแม่นยำใกล้เคียงกับ 3-Nearest Neighbor classification ที่ 79.349% ซึ่งจากโมเดลนี้สามารถนำไปใช้คาดการณ์ในการขึ้นเรือโดยสารในอนาคตว่าถ้าเกิดอุบัติเหตุ ผู้โดยสารกลุ่มไหนจะมีโอกาสรอดชีวิตและไม่รอดชีวิต ดังนั้น false positive ควรมีค่าน้อยกว่า โดย Random forest มี false positive น้อยกว่าเหมือนเดิม   
\*\*จึงสรุปได้ว่าควรใช้ Random forest แบบ 10fold-cv

6. **Deployment Phase** งานฉบับนี้ยังไม่ถึงช่วงนี้

**References**

http://digi.library.tu.ac.th/thesis/it/0729/03chapter2.pdf

https://www.kaggle.com/omarelgabry/a-journey-through-titanic?scriptVersionId=447802/notebook

http://scikit-learn.org/stable/modules/naive\_bayes.html

http://scikit-learn.org/stable/modules/generated/sklearn.neighbors.KNeighborsClassifier.html

http://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html

https://www.sanyamkapoor.com/machine-learning/confusion-matrix-visualization/

https://pandas.pydata.org/pandas-docs/stable/generated/pandas.Series.str.extract.html

http://www.ritchieng.com/machinelearning-one-hot-encoding/

https://towardsdatascience.com/train-test-split-and-cross-validation-in-python-80b61beca4b6

http://scikit-learn.org/stable/modules/generated/sklearn.model\_selection.cross\_val\_score.html

http://scikit-learn.org/stable/modules/cross\_validation.html

https://stackoverflow.com/questions/41458834/how-is-scikit-learn-cross-val-predict-accuracy-score-calculated