นางสาวณัฐณิชา ชื่นมะนา 58050263

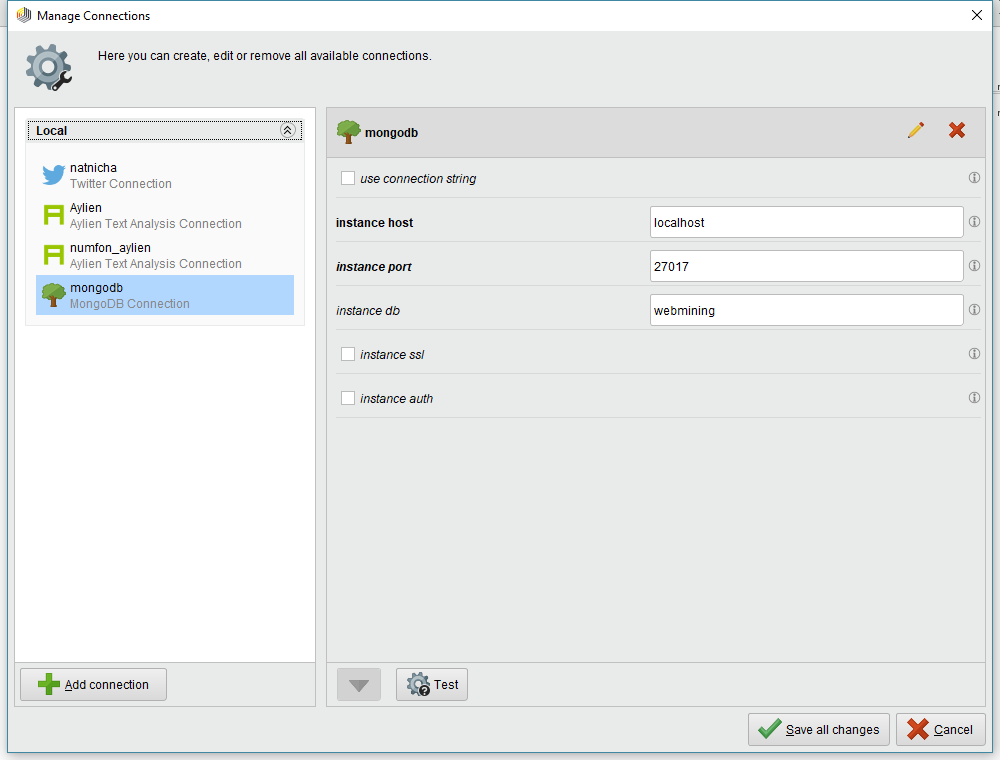
นางสาวปารียา วัยวุฒิ 58050327

นางสาวศุภิสรา ฉายาลักษณกุล 58050391

คำที่ใช้ในการค้นหา คือ **“Beyounce”**

จุดมุ่งหมายของการใช้คำ “Beyounce”

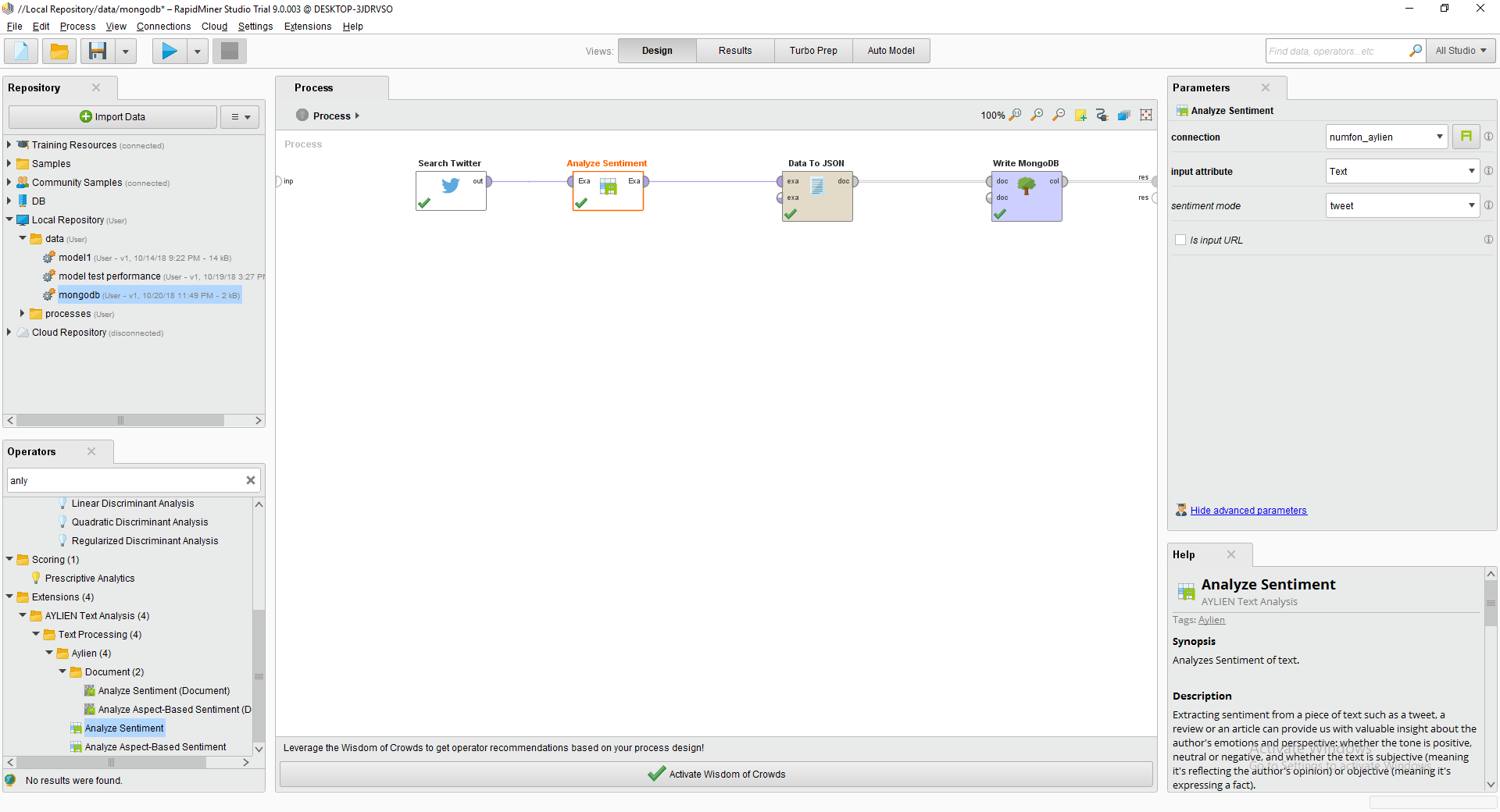
เนื่องจากการทางกลุ่มมีความสนใจในตัวของนักร้องที่มีชื่อว่า “Beyounce” จึงจัดทำการสืบค้นข้อมูลโดยใช้คำว่า “Beyounce” เป็นคำหลักที่เอาไว้ใช้ในการค้นหาข้อความที่ได้กล่าวถึงคำนี้ใน Twitter เพื่อมาทำSentiment Analysis วิเคราะห์ข้อดี ข้อเสีย ของข้อความที่ได้กล่าวถึง และจากนั้นเราก็ได้นำชุดข้อมูลที่ได้จากกระบวนการสืบค้นข้อมูลมาทำกระบวนการ Text Processing เพื่อทำการตัดคำ และหาประสิทธิภาพของข้อมูลที่เราได้สืบค้นมาได้ จากนั้นก็มาเข้ากระบวนการทำ WordCloud เพื่อนำผลลัพธ์ของคำที่ได้มาแสดงผลเป็นภาพเพื่อให้เห็นถึงความสำคัญของแต่ละคำที่มีให้เห็นได้ชัดเจนยิ่งขึ้น

**ขั้นตอนในการทำ  
1.**ใช้โปรแกรม RapidMiner ในการดึงข้อมูลจาก Twitter มาทำการวิเคราะห์ข้อมูล และจัดเก็บข้อมูลลงMongoDB โดยที่ต้องทำการลงplug in ทั้งหมด 3 ตัว นั้นก็คือ Twitter Connection ,Aylien Text Analysis Connection และ MongoDB Connection ดังภาพที่1

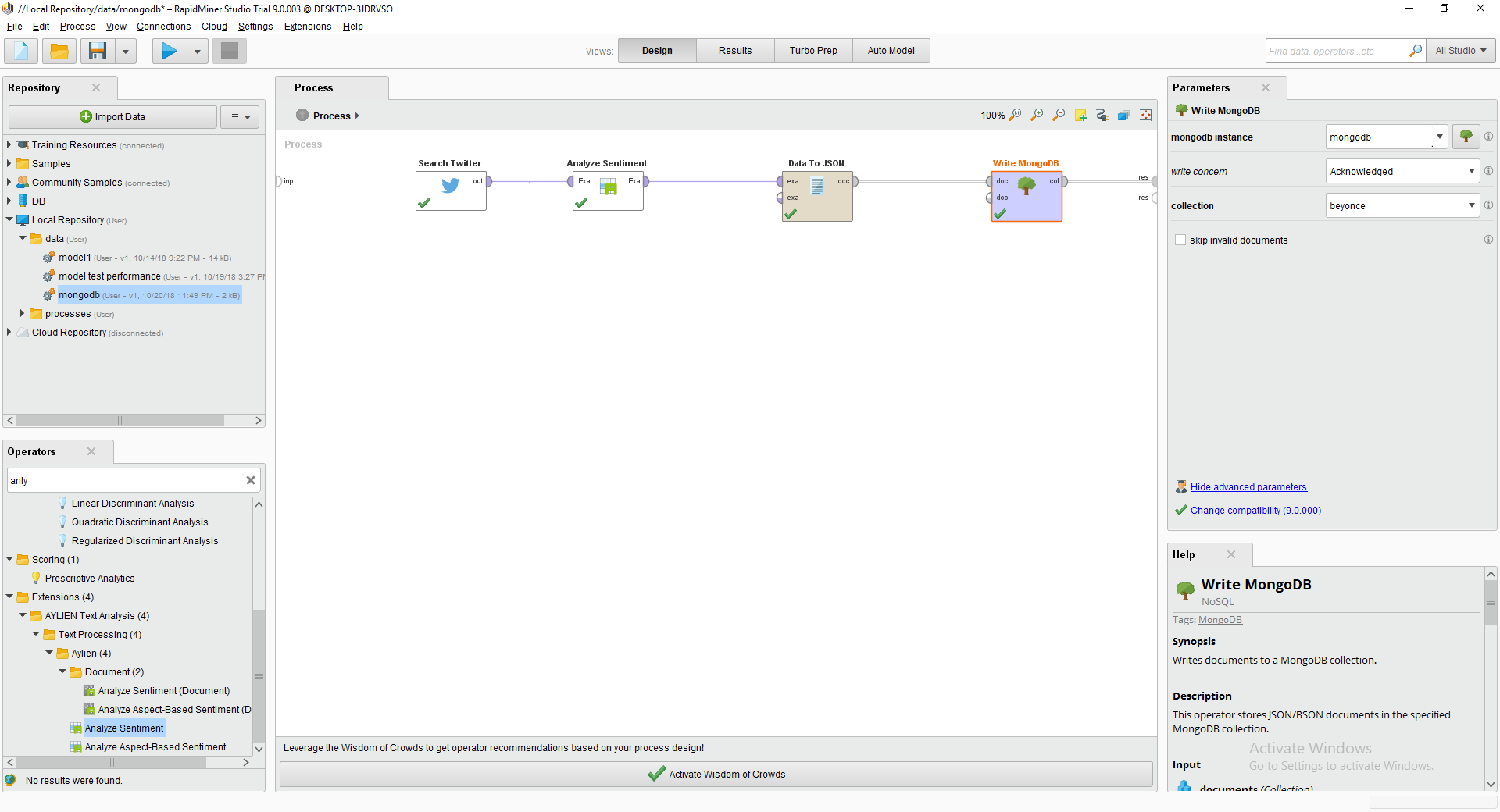
ภาพที่ 1 แสดง Plug in ที่ใช้การทำงาน

2 .เมื่อเราทำการลงPlug inเรียบร้อยแล้ว เราก็มาทำการใช้Operator ได้แก่ Search twitter , Analysis sentiment , Data to JASON และ Write MongoDB โดยใช้คำว่า “Beyounce” ในการค้นหาชุดข้อมูลมาทั้งหมด 2,000 ชุดข้อมูล แบ่งเป็นข้อมูลเชิงบวก (Positive) : 1,347 ข้อมูล และ ข้อมูลเชิงลบ (Negative) : 653 ข้อมูลและทำการเขียนข้อมูลลงในMongoDB ดังภาพที่ 2 , 3 และ 4 ตามลำดับ



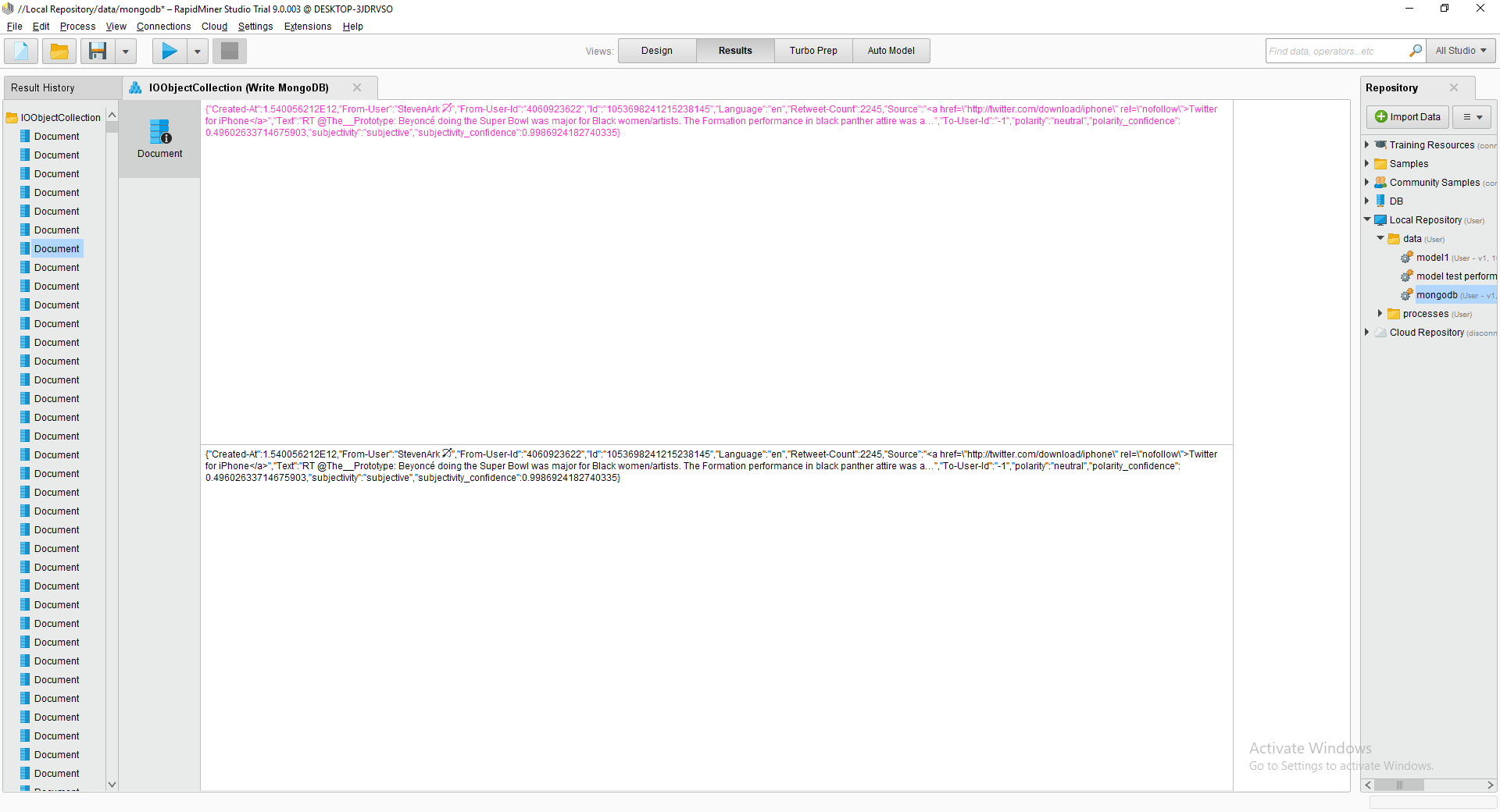
ภาพที่ 2 แสดง limit ในการดึงข้อมูลจากtwitterมา 2,000 ชุดข้อมูล

ภาพที่ 3 แสดงการทำ Analysis sentiment

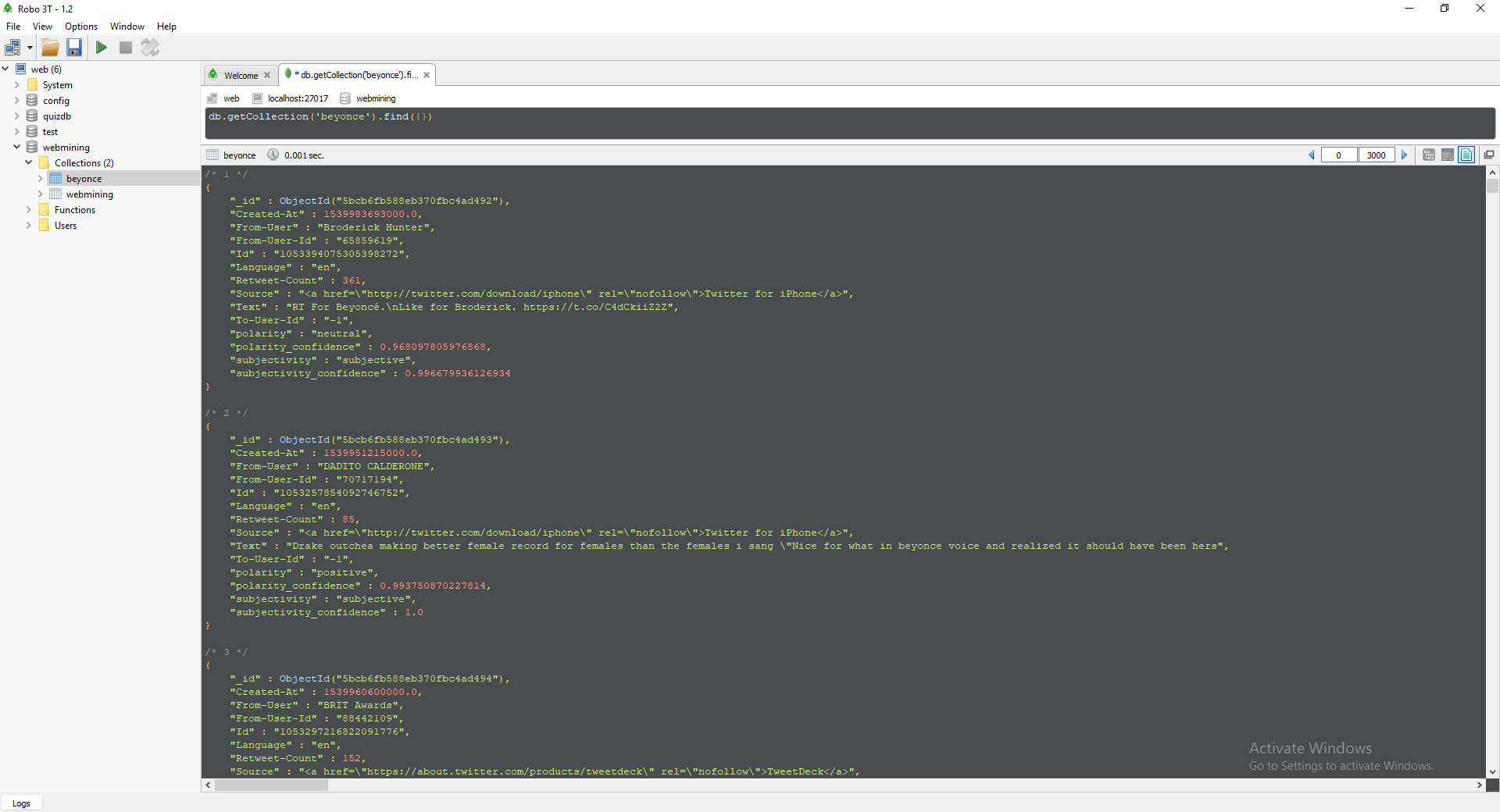


ภาพที่ 4 แสดงที่จัดเก็บข้อมูลที่ได้ทำการค้นหาในtwitter

3. เมื่อทำการ Process ดึงข้อมูลและทำการวิเคราะห์ข้อดี ข้อเสียของข้อมูลที่ได้ทำการดึงขึ้นมาแล้ว โปรแกรมจะทำการแสดงผลลัพธ์ในรูปแบบของ JSON ซึ่งชุดข้อมูลที่ได้นั้นระบบได้ทำการจัดเก็บลง MongoDB ที่ Collection ที่มีชื่อว่า Beyounce เรียบร้อยแล้ว ดังภาพที่ 5 และ 6



ภาพที่ 5 แสดงผลลัพธ์จากการทำกระบวนการดึงข้อมูลและวิเคราะห์ข้อมูลคำว่า “Beyounce”

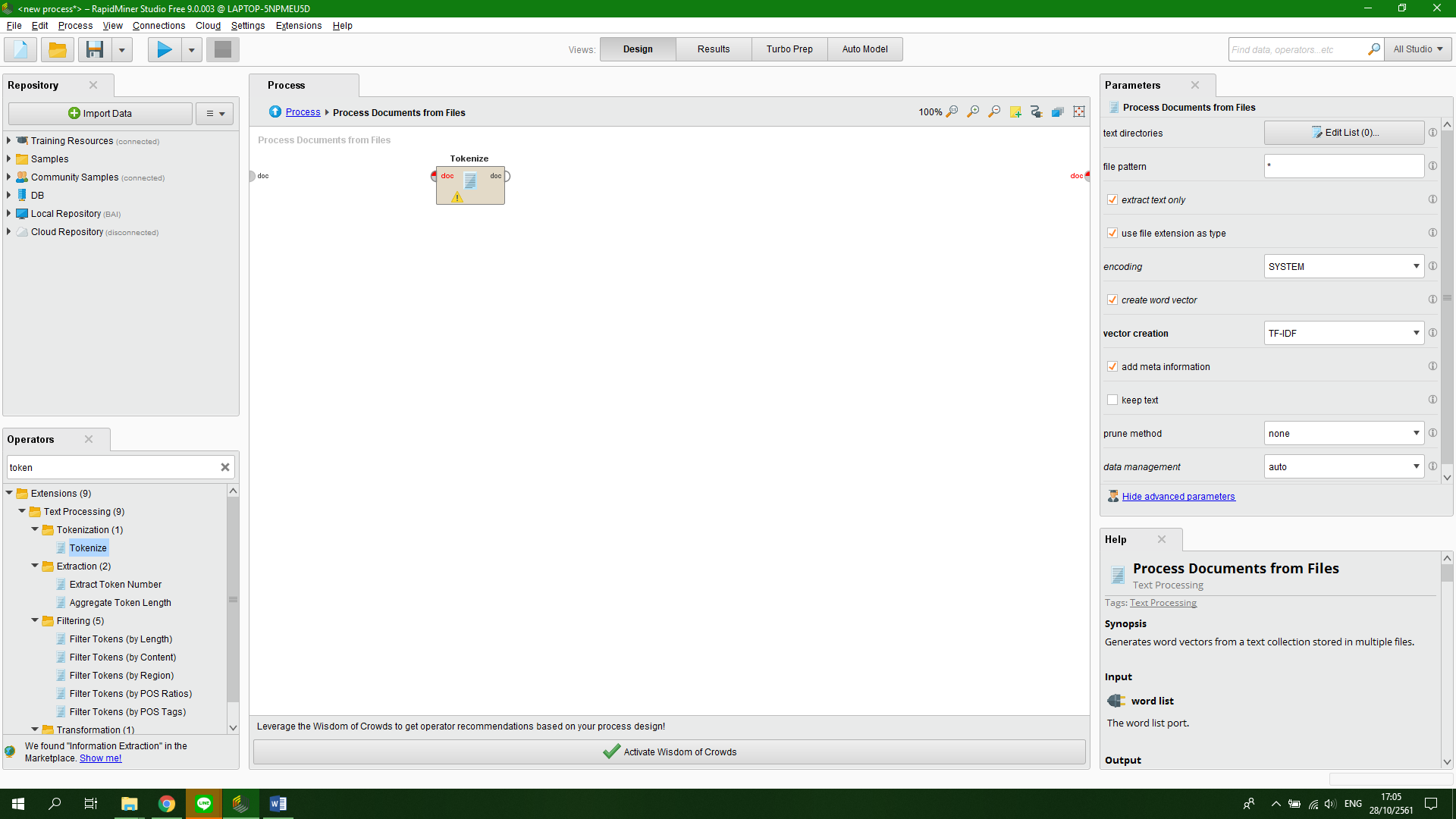
  
ภาพที่ 6 แสดงผลลัพธ์ที่ได้จากการสืบค้นข้อมูลมาจัดเก็บลง MongoDB ในรูปแบบ JSON

4. เมื่อเราได้ชุดของข้อมูลแล้วเราจำเป็นต้องทำการวิเคราะห์เพื่อหาประสิทธิภาพของชุดข้อมูลนั้นๆโดยการเพิ่มโอเปอเรเตอร์ Process Document from Files เพื่อทำ Preprocess ข้อมูล สำหรับจัดการข้อความ (text) จากคอลเล็กชันข้อความที่เก็บไว้ในหลายไฟล์ จากนั้น Double Click ที่โอเปอเรเตอร์ ดังภาพที่7



ภาพที่ 7 โอเปอเรเตอร์ Process Document from Files

5. เพิ่มโอเปอเรเตอร์ Tokenize สำหรับตัดคำจากข้อความให้ออกมาเป็นคำเดี่ยว ๆ ดังภาพที่8



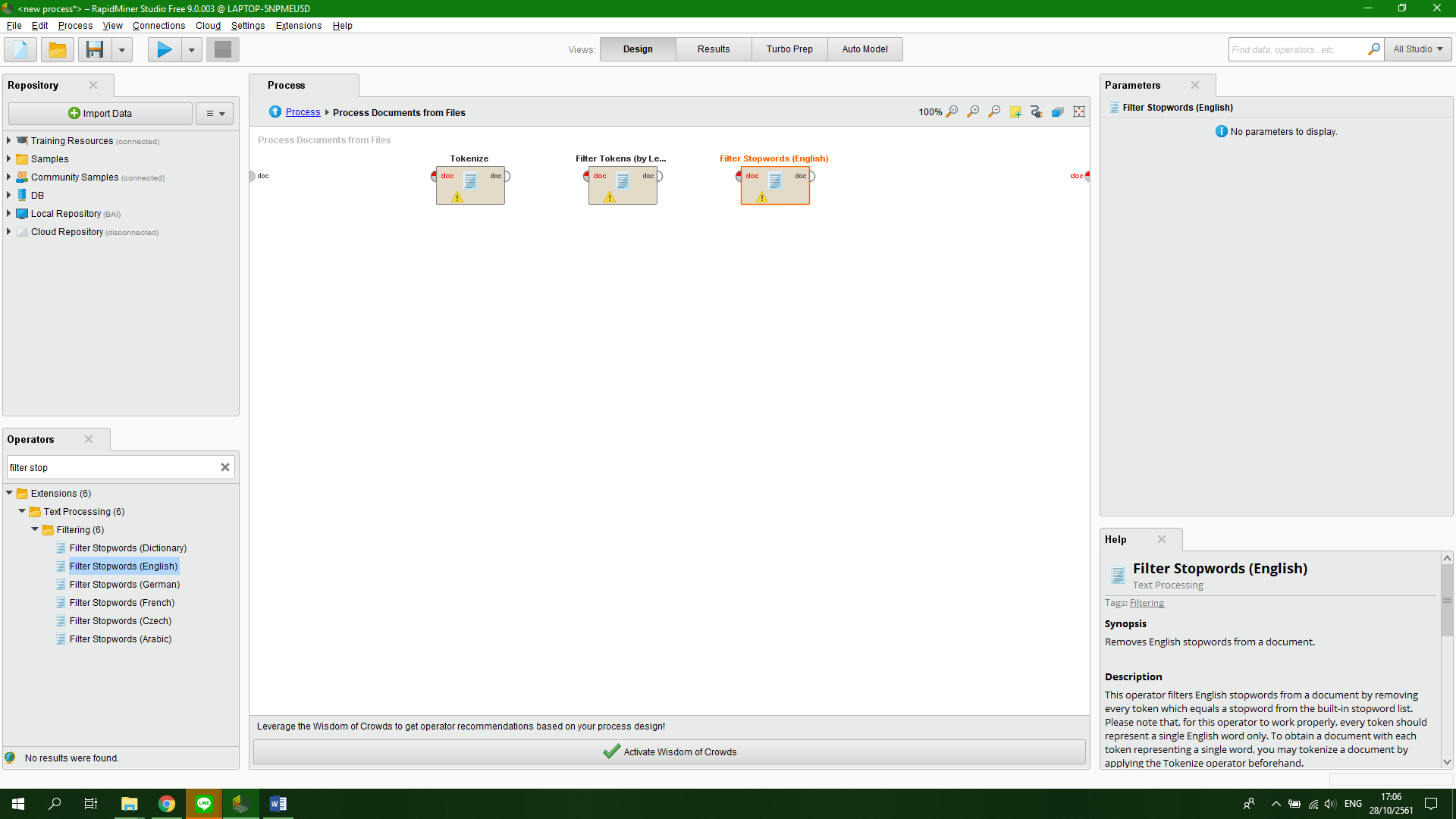
ภาพที่ 8 โอเปอเรเตอร์ Tokenize

6. เพิ่มโอเปอเรเตอร์ Filter Tokens (by Length) สำหรับกรองคำศัพท์ (Token) ที่มีค่าน้อยกว่าหรือมากกว่าที่กำหนด ดังภาพที่9



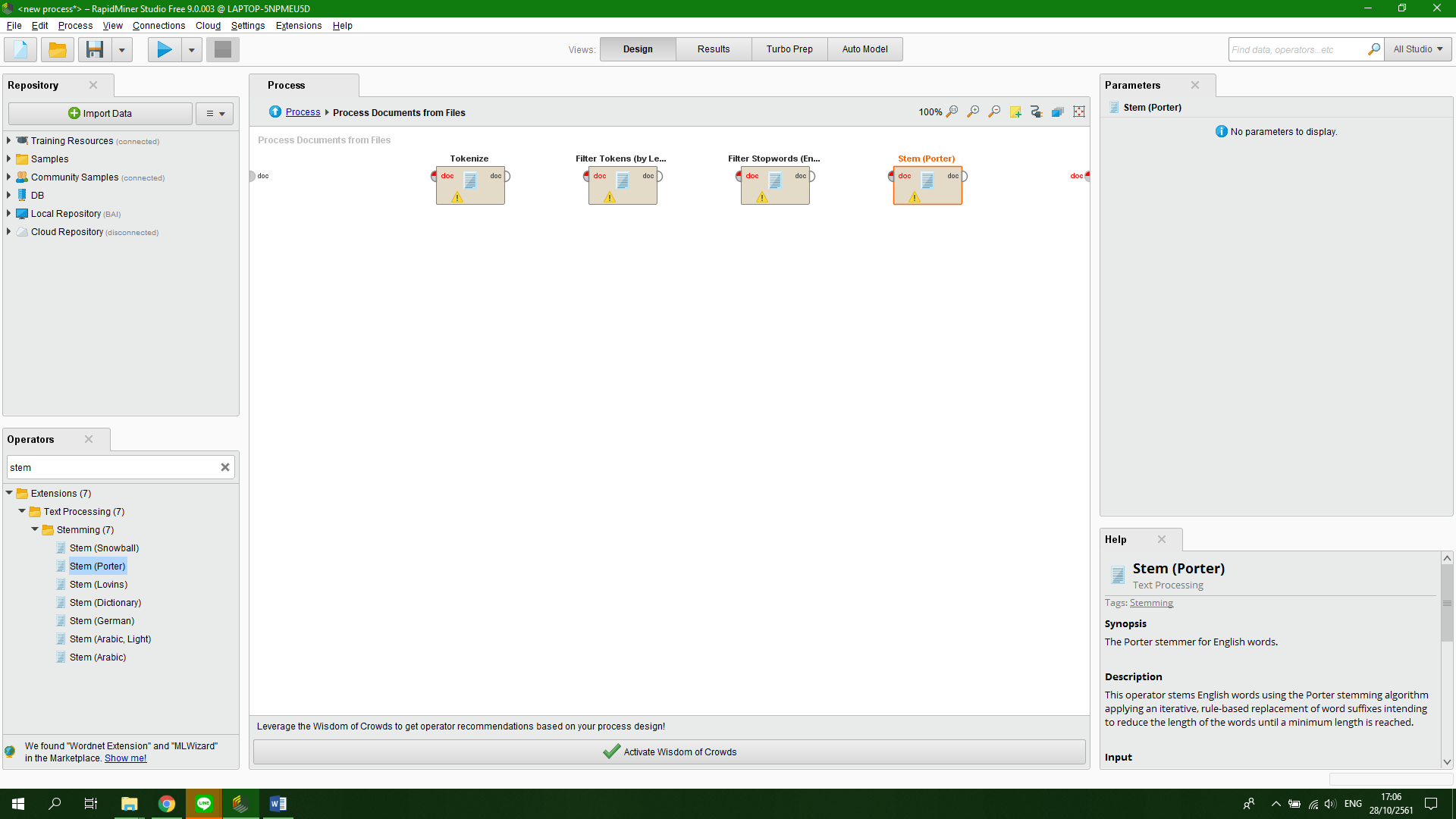
ภาพที่ 9 โอเปอเรเตอร์ Filter Tokens (by Length) สำหรับกรองคำศัพท์ (Token)

7. เพิ่มโอเปอเรเตอร์ Filter Stopwords (English) สำหรับตัดคำเชื่อมหรือคำที่ไม่จำเป็นทิ้ง ดังภาพที่10



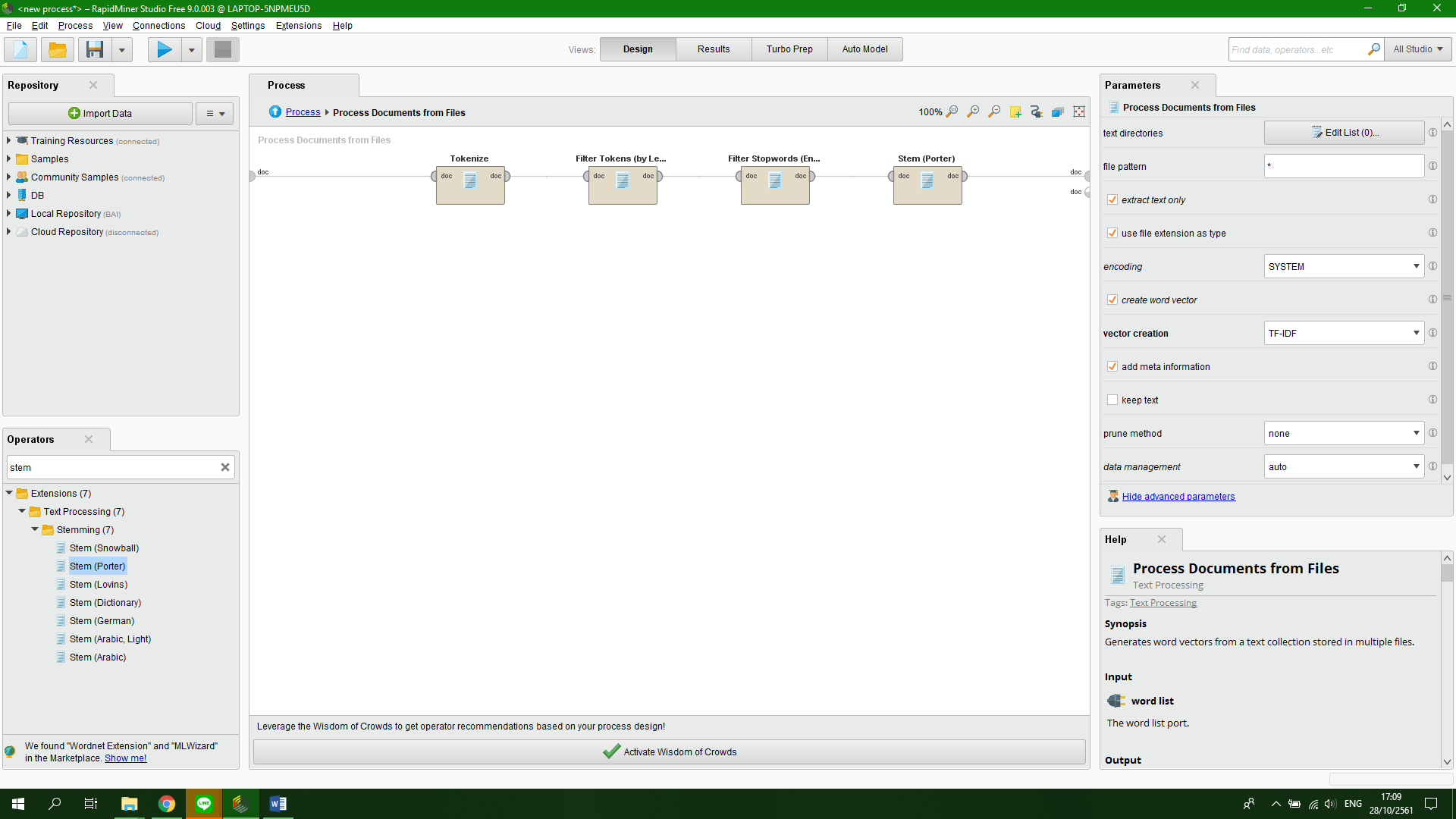
ภาทที่ 10 โอเปอเรเตอร์ Filter Stopwords (English)

8. เพิ่มโอเปอเรเตอร์ Stem (Porter) สำหรับแปลงคำศัพท์ให้อยู่ในรูปของรากศัพท์ภาษาอังกฤษ (Root) ดังภาพที่11



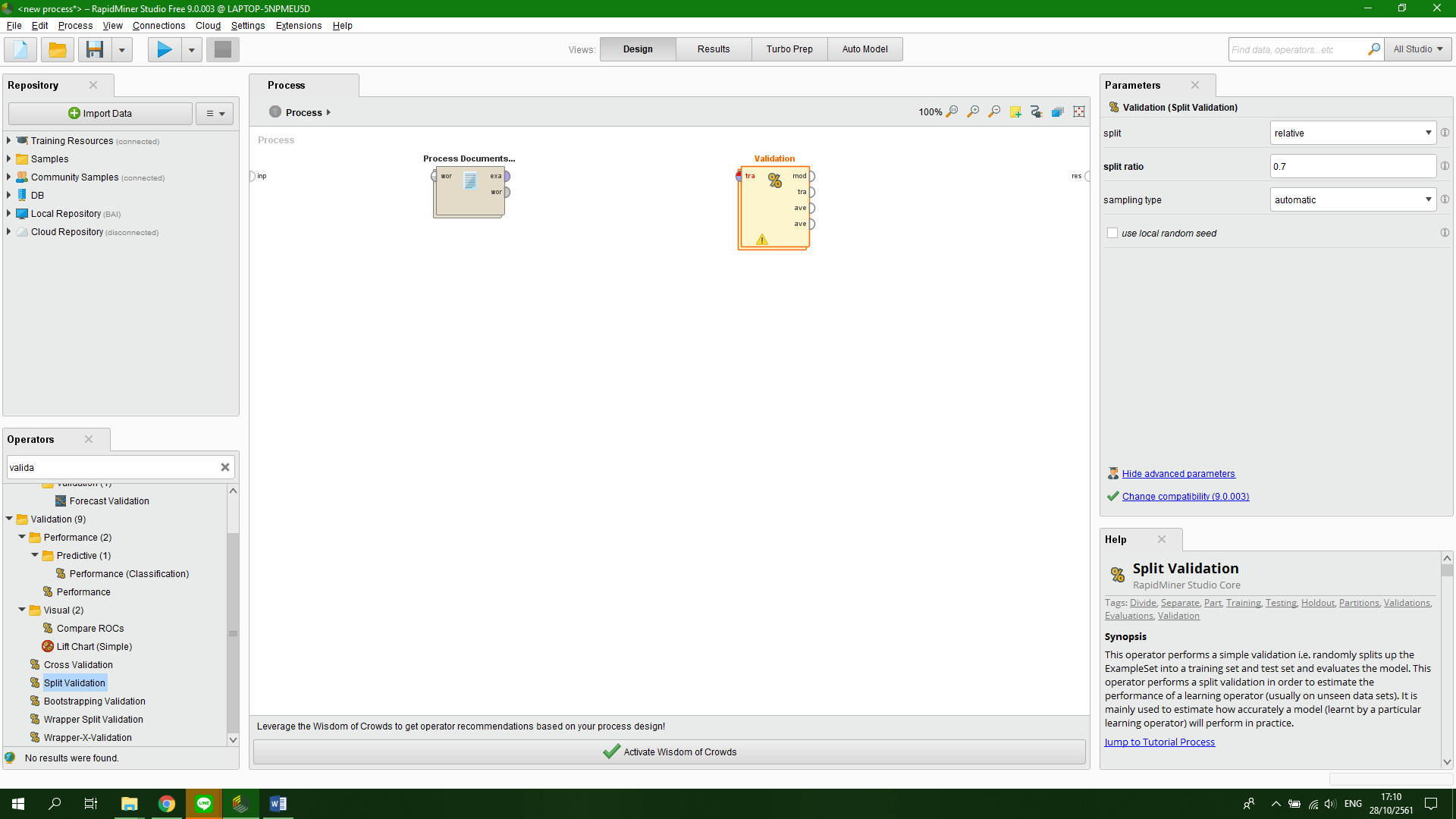
ภาทที่ 11 โอเปอเรเตอร์ Stem (Porter)

9. จากนั้นเชื่อมโอเปอเรเตอร์ Tokenize ต่อเข้ากับ Port doc (ด้านซ้าย) แล้วต่อ Port จาก Tokenize (ด้านขวา) ไปยัง Filter Tokens (by Length) (ด้านซ้าย) แล้วต่อ Port จาก Filter Tokens (by Length) (ด้านขวา) ไปยัง Filter Stopwords (English) (ด้านซ้าย) แล้วต่อ Port จาก Filter Stopwords (English) (ด้านขวา) ไปยัง Stem (Porter) (ด้านซ้าย) และต่อ Stem (Porter) (ด้านขวา) ไปยัง Port doc ดังภาพที่12



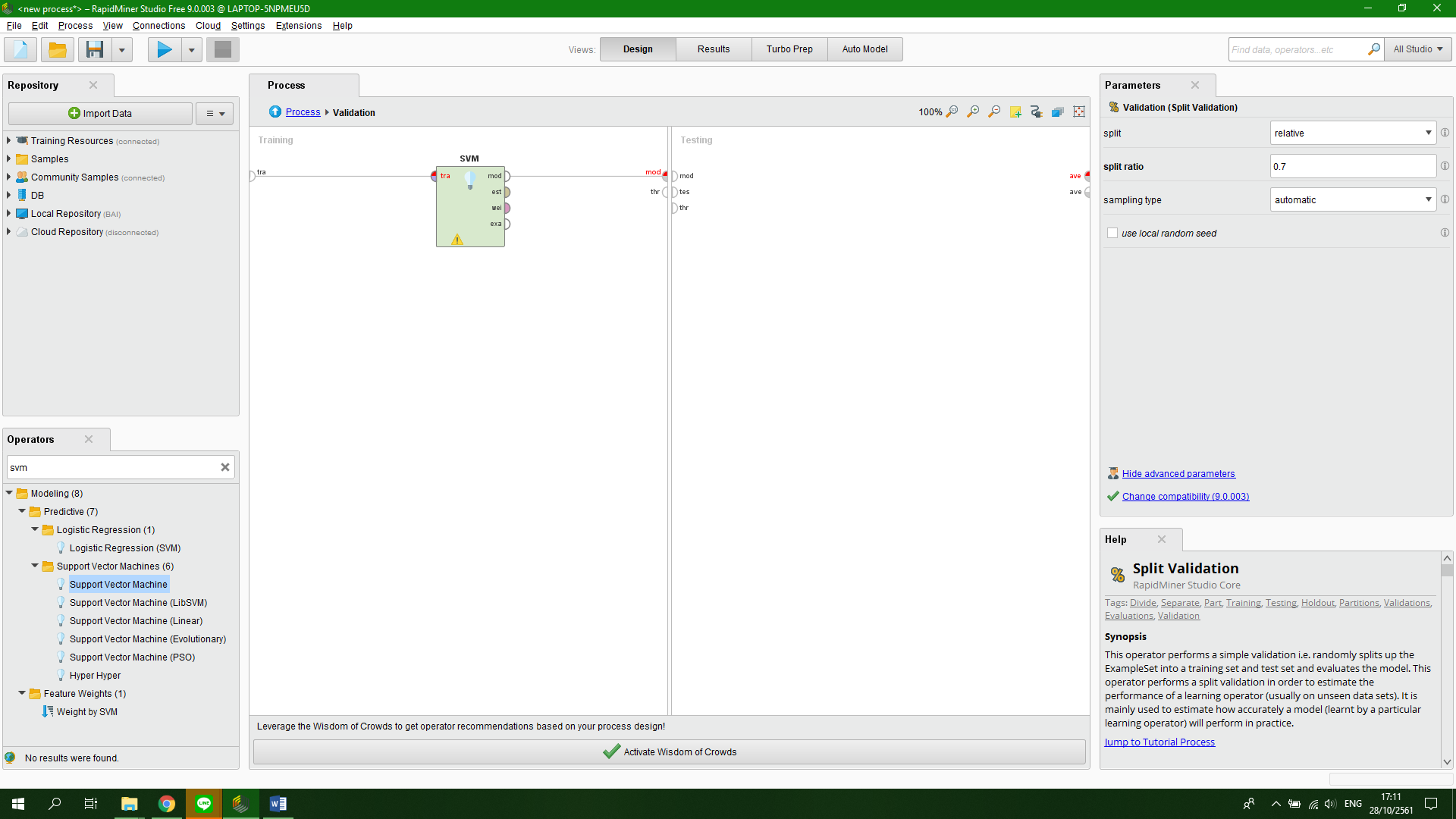
ภาพที่ 12 การเชื่อมต่อแต่ละโอเปอเรเตอร์เพื่อจะนำไปแสดงผล

10. เพิ่มโอเปอเรเตอร์ Validation สำหรับแบ่งข้อมูลและทำการทดสอบแบบ Cross-Validation จากนั้น Double Click ที่โอเปอเรเตอร์ ดังภาพที่13



ภาพที่ 13 เพิ่มโอเปอเรเตอร์ Validation ตรง Process

11. เพิ่มโอเปอเรเตอร์ Support Vector Machine (SVM) ในการ Training เพราะสนับสนุนการประมวลผลที่เป็นข้อความได้ดีที่สุด เหมาะกับข้อมูลที่มีปริมาณมาก เนื่องจากไม่ค่อยเกิดปัญหา Overfitting จากนั้นต่อ Port tra ไปยัง Port tra ของSVM (ด้านซ้าย) และ Port mod ของ SVM (ด้านขวา) ต่อกับ Port mod ดังภาพที่14



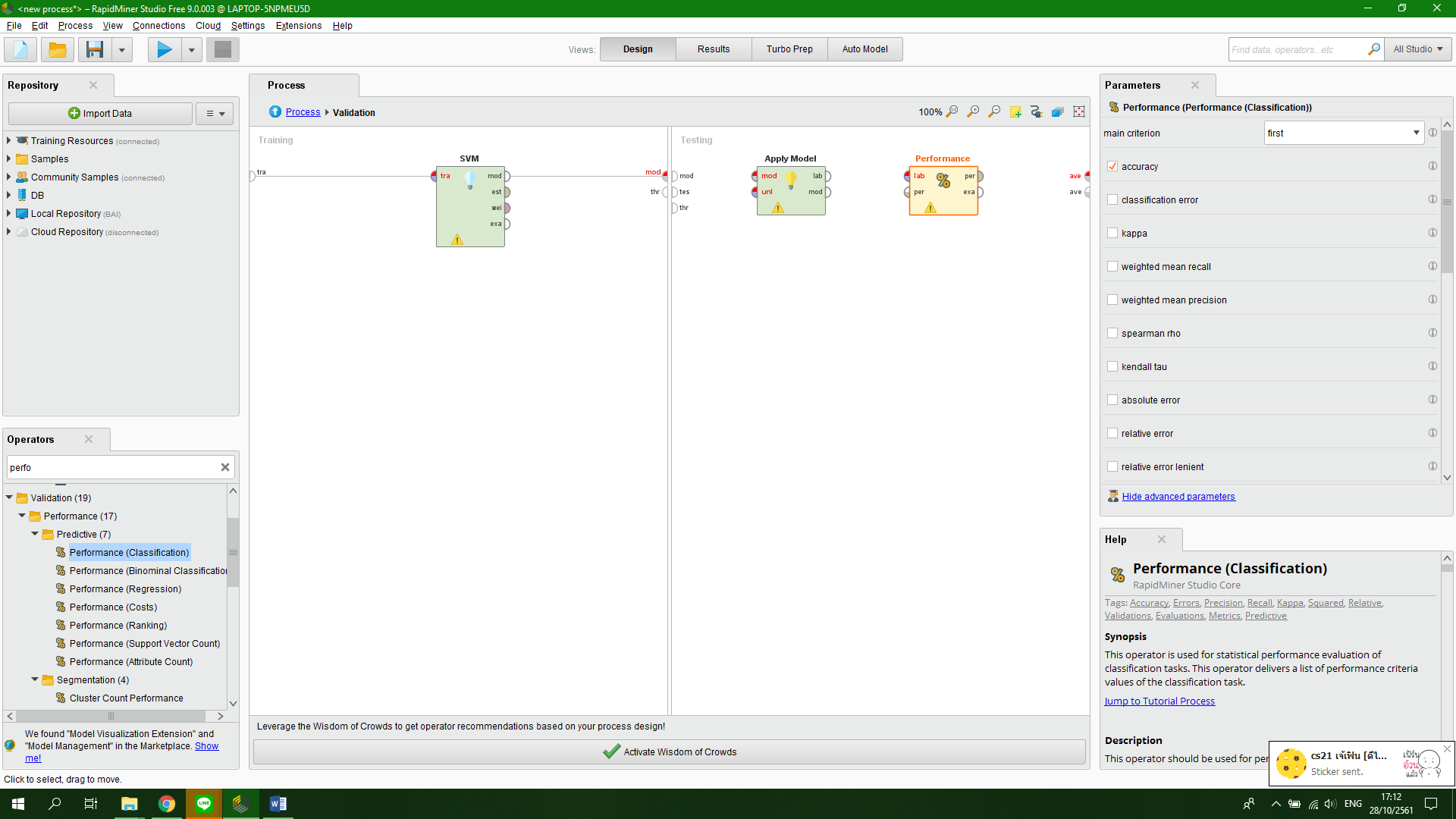
ภาพที่ 14 โอเปอเรเตอร์ Support Vector Machine (SVM) ในการ Training

12. เพิ่มโอเปอเรเตอร์ Apply Model สำหรับนำโมเดลไปทำนาย (Predict) ข้อมูลใหม่ ดังภาพที่15



ภาพที่ 15 โอเปอเรเตอร์ Apply Model สำหรับนำโมเดลไปทำนาย (Predict) ข้อมูลใหม่

13. เพิ่มโอเปอเรเตอร์ Performance สำหรับวัดประสิทธิภาพของโมเดลที่สร้างขึ้น ดังภาพที่16



ภาพที่ 16 โอเปอเรเตอร์ Performance

14. จากนั้นต่อ Port mod และ tes ไปยัง Port mod และ unl ของ Apply Model (ด้านซ้าย) ตามลำดับ และต่อ Port lab และ mod ของ Apply Model (ด้านขวา) ไปยัง Port lab ของ Performance (ด้านซ้าย) แล้วต่อ Port per ของ Performance (ด้านขวา) ไปยัง Port ave ดังภาพที่17



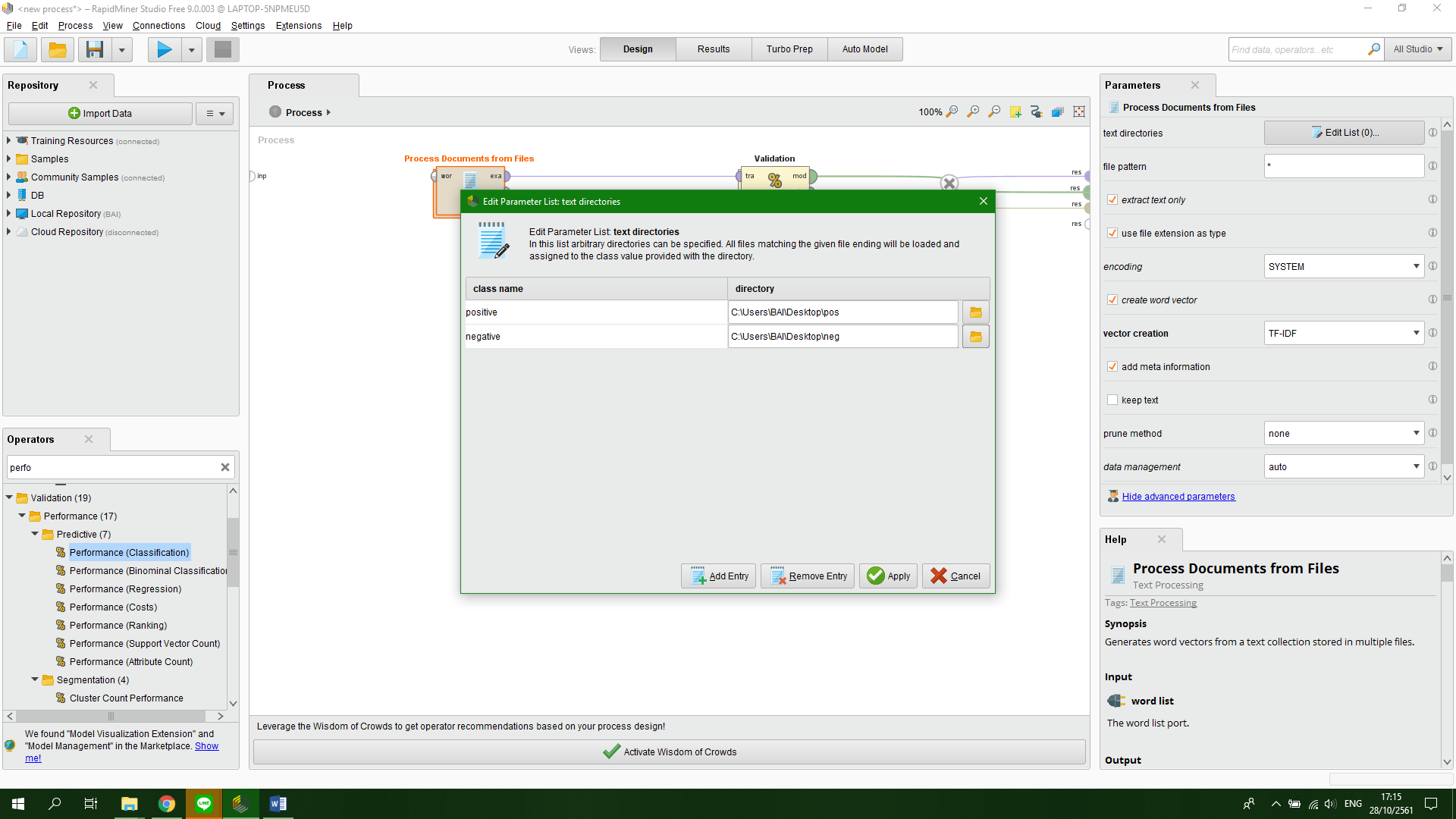
ภาพที่ 17 เชื่อมต่อระหว่าง Training และ Testing

15. จากนั้นต่อ Port exa ของ Process Document from file (ด้านขวา) ไปยัง Port tra ของ Validation (ด้านซ้าย) แล้วต่อ Port Mod และ Port ave ของ Validation (ด้านซ้าย) ไปยัง Port res ดังภาพที่18



ภาพที่ 18 เชื่อมต่อระหว่าง Process Document from files กับ Validation

16. จากนั้นกดที่ Process Document from files แล้วเลือก Edit List (หมายเลข 1) เพื่อเลือกโฟลเดอร์ที่เก็บข้อความเชิงบวก (Positive) และ ข้อความเชิงลบ (Negative) โดยตั้งชื่อ Class Name : Positive และกดหมายเลข 2 เพื่อเลือกโฟลเดอร์ที่เก็บข้อความ จากนั้นกดหมายเลข 3 เพื่อเพิ่ม Entity หรือเพิ่ม Class ขึ้นมาอีกหนึ่ง ตั้งชื่อ Class Name : Negative และกดหมายเลข 2 เพื่อเลือกโฟลเดอร์ที่เก็บข้อความ ดังภาพที่19



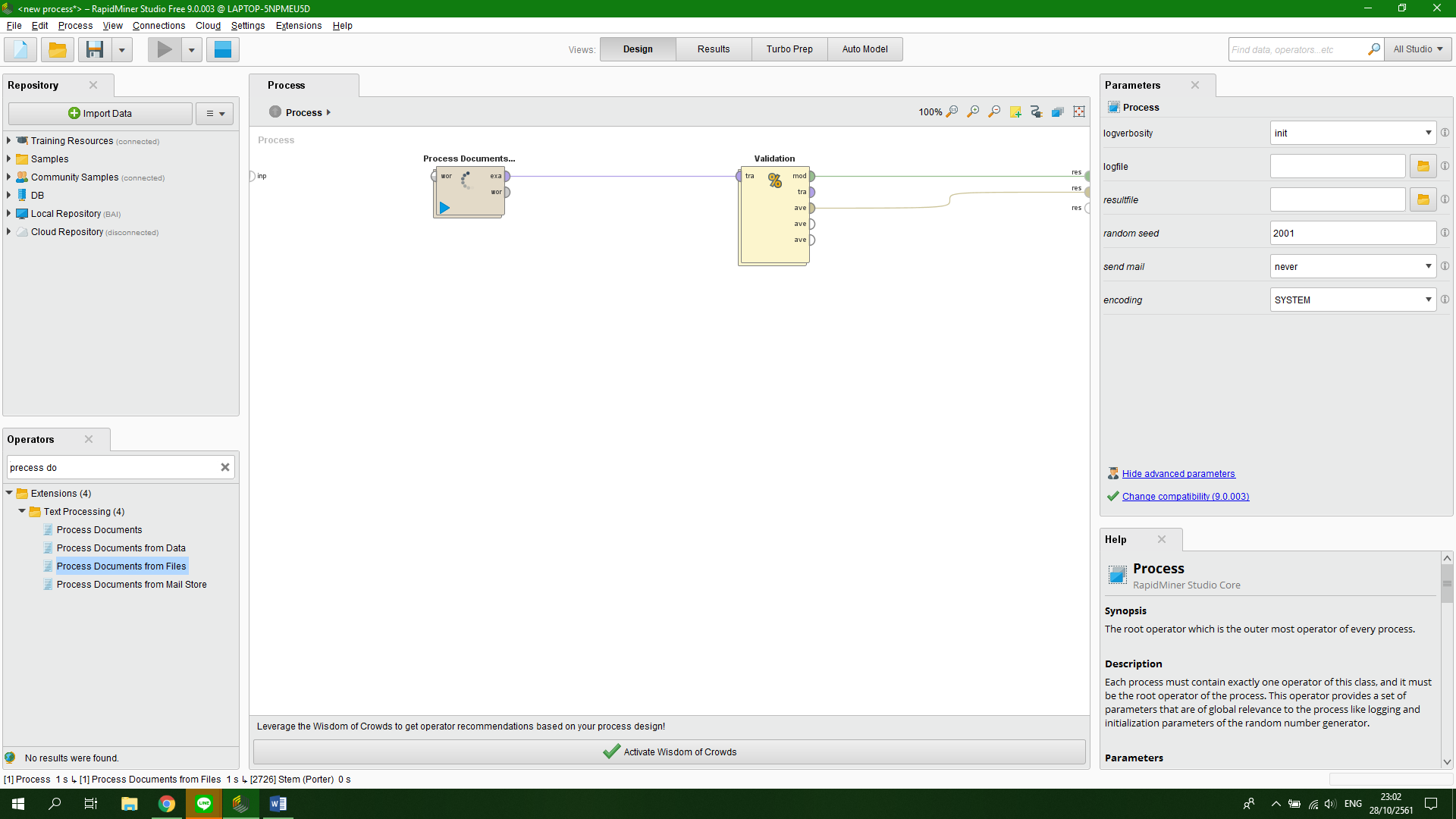
1

3

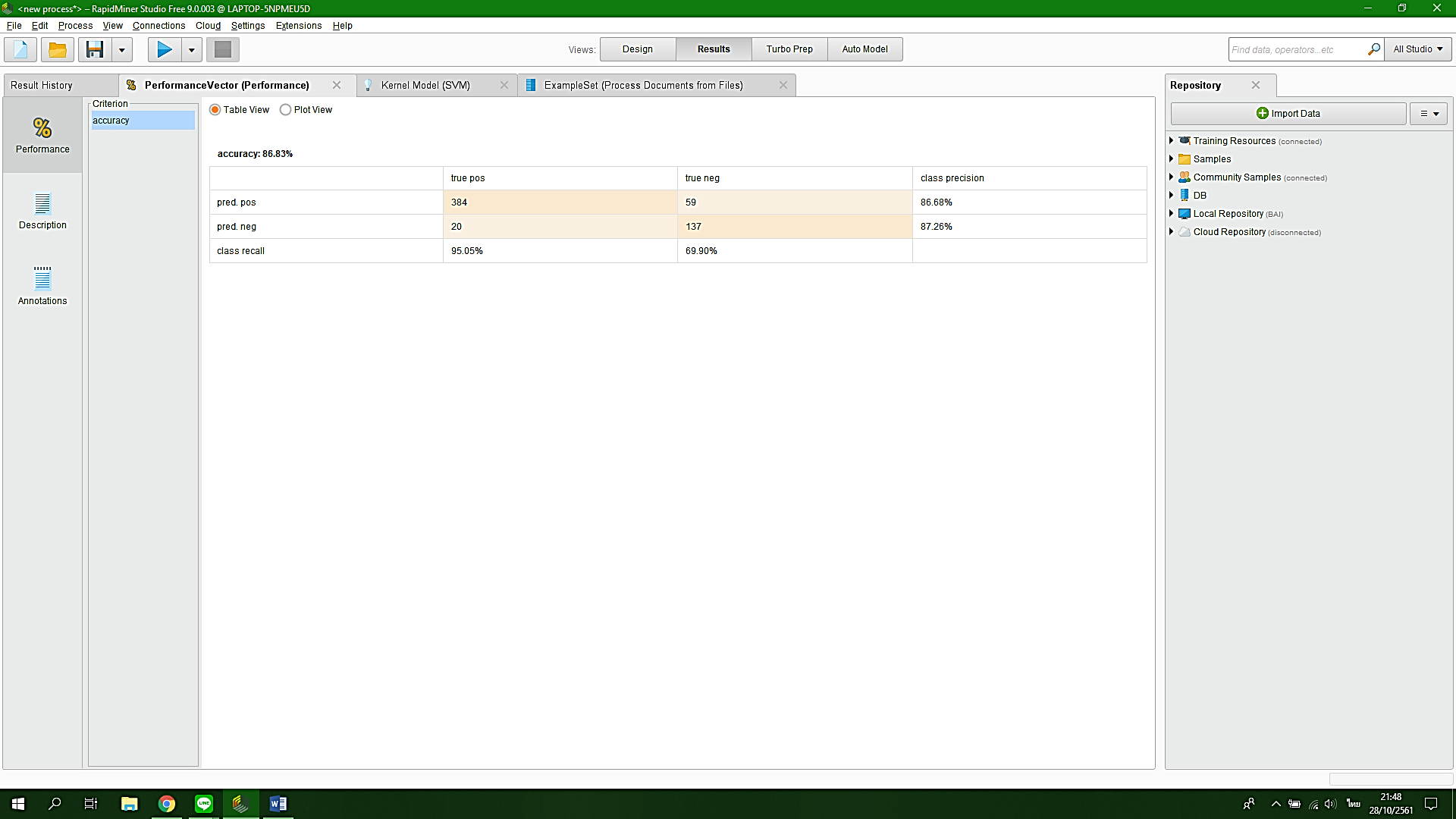
2

ภาพที่ 19 นำเอาfiles positive กับ negative เข้ามา

17. จากนั้นกดปุ่ม Run Process Locally เพื่อดูผลการทดสอบประสิทธิภาพของโมเดล ดังภาพที่20-21



ภาพที่ 20 Run Process Locally เพื่อแสดงผลลัพธ์



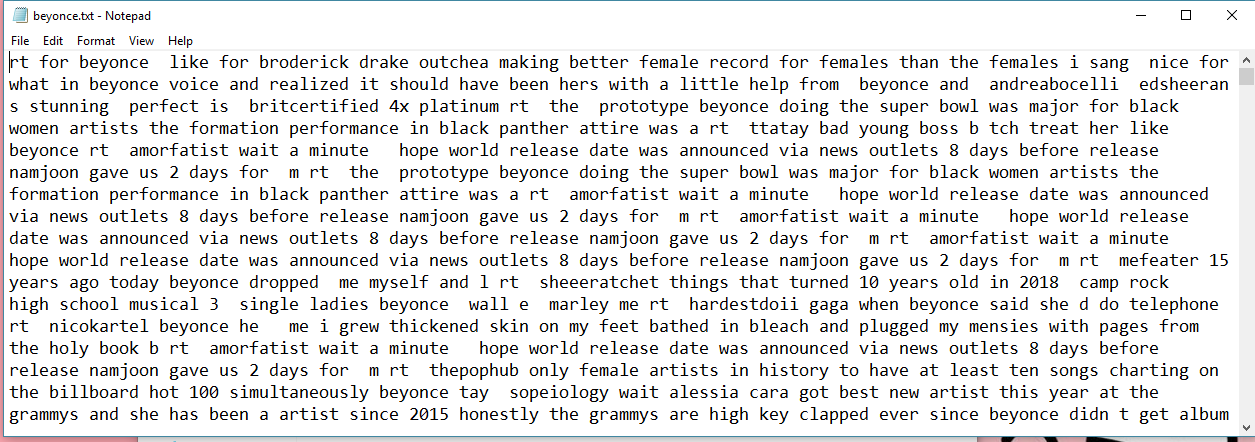
ภาพที่ 21 ผลการทดสอบประสิทธิภาพของโมเดล = 86.83 %

18.เมื่อเราได้ทำการทดสอบชุดของข้อมูลแล้วได้ประสิทธิภาพที่ดี เป็นน่าพึงพอใจตามดังภาพที่21 เราก็มาสามารถนำชุดข้อมูลนั้นมาเข้ากระบวนการดึงชุดข้อมูลที่เราได้ทำการจัดเก็บในMongoDBจากขั้นตอนที่3.  
โดยเราจะใช้Spring boot framework ในการทำ ซึ่งเราจะนำชุดข้อมูลนั้นมาประมวลผลตัดคำ ตัดอักขระพิเศษออก และทำการ saveลงไฟล์ .txt ตามขั้นตอนดังภาพที่ 22-23

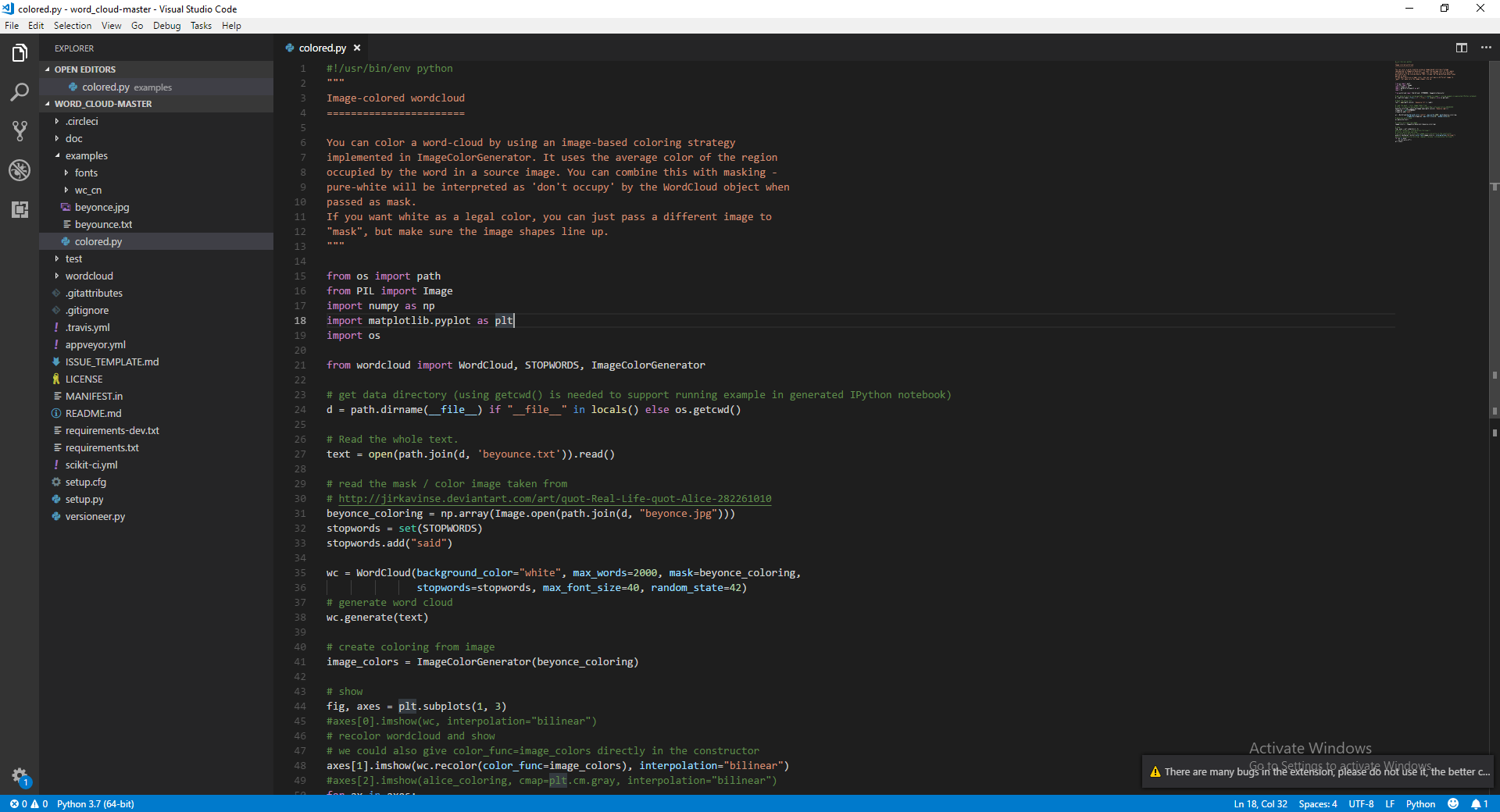
ภาพที่ 22 แสดงกระบวนการดึงข้อมูล และ มาทำการตัดคำ

 ภาพที่ 23 แสดงกระบวนการตัดอักขระพิเศษ และ เขียนข้อมูลลง.txt

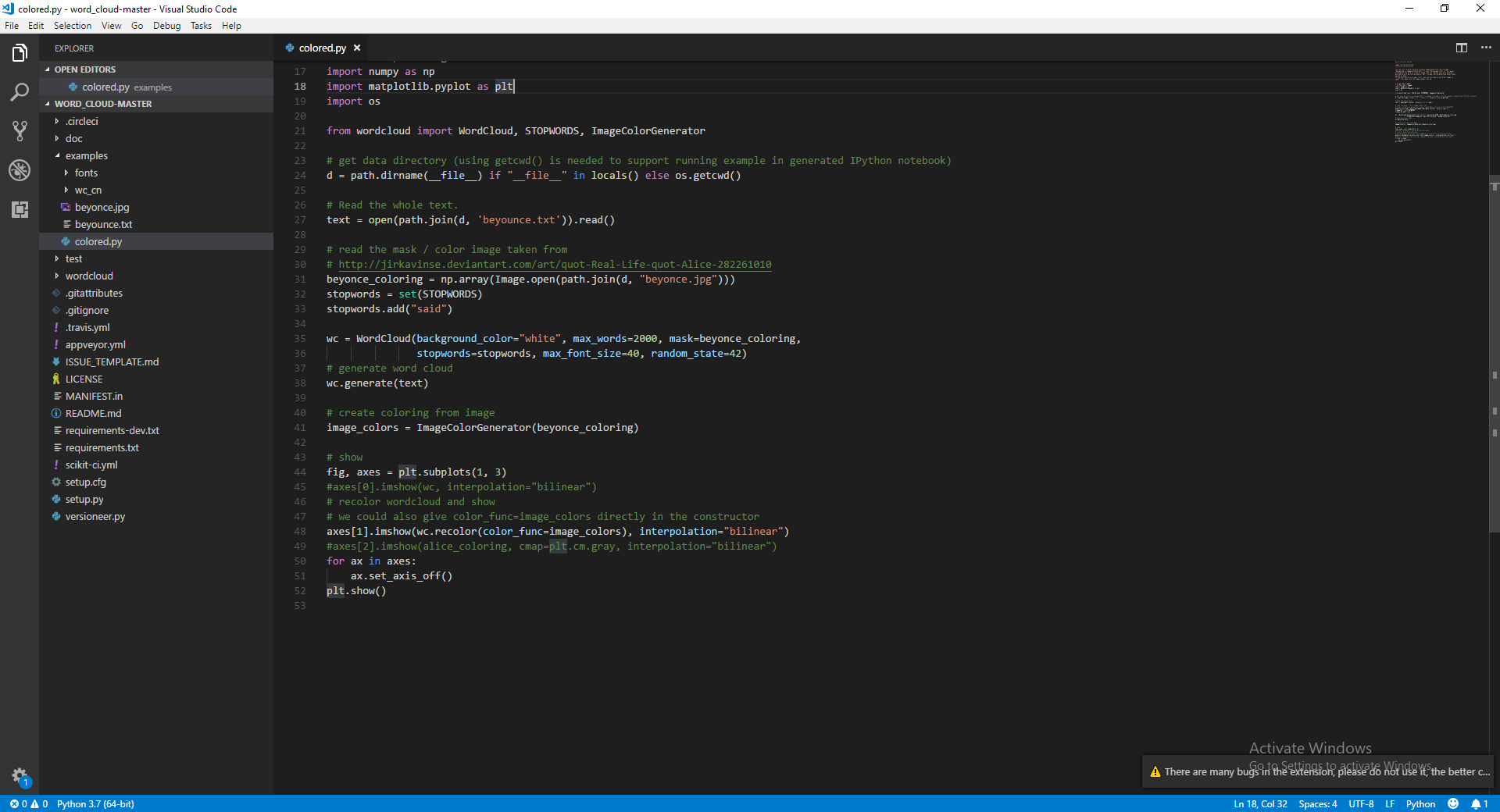
19. เมื่อเราทำตามกระบวนการข้อที่18. แล้ว เราก็จะได้ผลลัพธ์ออกมาเป็นไฟล์beyonce.txt ดังภาพที่ 24

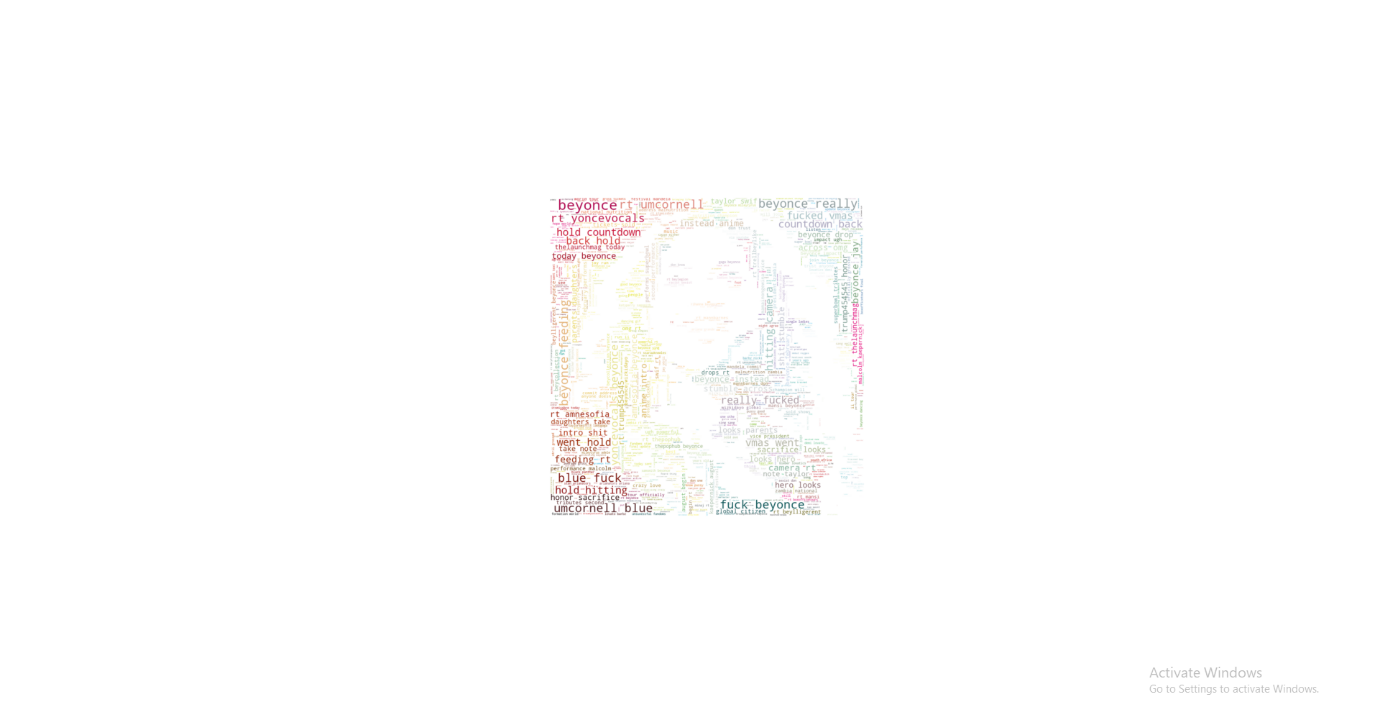


ภาพที่ 24 แสดงผลลัพธ์beyonce.txt

20. แล้วเราก็นำเอา beyonce.txt ไปใส่ในกระบวนการทำ wordcloud ดังภาพที่25-26 เพื่อที่จะทำการประมวลผลออกมาเป็นดังภาพที่27

ภาพที่ 25 code เพื่อเข้าทำwordcloud

  
ภาพที่ 26 code เพื่อเข้าทำwordcloud



ภาพที่ 27 ผลลัพธ์ที่ได้จากการนำ beyonce.txt ไปทำ wordcloud

สรุปผลลัพธ์ที่ได้

จากกระบวนการที่ได้ทำมาตั้งแต่ต้นจนสรุปได้ออกมาเป็นผลลัพธ์สุดท้ายออกมาเป็น wordcloud ได้ว่าจำนวนคำในแต่ล่ะคำนั้นๆที่ได้ผ่านกระบวนการตัดคำจะส่งผลถึงขนาดของคำที่จะนำแสดงมาในผลลัพธ์ของwordcloud ดังภาพที่27 ซึ่งในเอกสารนี้ได้ใช้คำหลักในการสืบค้นข้อมูลก็คือคำว่า “beyonce” และผลที่ได้จากชุดข้อมูลในช่วงนี้ผู้คนได้กล่าวถึง beyonce ในทิศทางที่เป็นแง่ลบมากกว่าทิศทางที่ดีต่อตัวเขา และเป็นที่ยืนยันได้อีกว่าชุดข้อมูลนี้ที่ได้ทำการสืบค้นมาได้ตรงตามคำหลักที่ผู้จัดทำได้ตั้งใจที่จะสืบค้นข้อมูล beyonce เพราะในภาพ wordcloud ได้แสดงคำว่า beyonce อย่างเด่นชัด