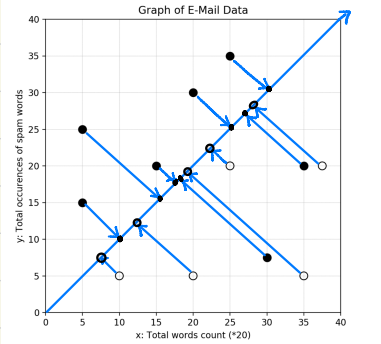
a. Draw the first principal component as a line ($y = x$ in this case) on the graph. Also draw the data points projected to this subspace with lines connecting them to the corresponding original data points.




b. How would **1**-Nearest Neighbour algorithm classify an e-mail with 21 spam words in a total of 100 ($x$=5) words? Explain your answer.

c. How would **1**-Nearest Neighbour algorithm classify the same e-mail given in the previous question after transforming the data to 1-dimension? Explain your answer.

d. Are your answers same for part b and c? Why or why not?

b. In 2-D, the nearest neighbor to the point (5,21) is (5,25). Since (5,25) is labeled as a spam email, then (5,21) would also be classified as a spam email since we're only classifying based on the single nearest neighbor.

c. We can draw the point (5,21), and draw it's projection to the first principal component. This would transform it into 1-dimension. Mathematically, $\begin{bmatrix} 5 \\ 21 \end{bmatrix} \times \begin{bmatrix} \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{bmatrix} = \frac{5}{\sqrt{2}} + \frac{21}{\sqrt{2}} = 13\sqrt{2}$

Its nearest neighbor is the point (20,5) in 2 dimensions, which is labeled as non-spam. Therefore, (5,21) would be labeled as non-spam. Note: $\begin{bmatrix} 20 \\ 5 \end{bmatrix} \times \begin{bmatrix} \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{bmatrix} = \frac{20}{\sqrt{2}} + \frac{5}{\sqrt{2}} = \frac{25\sqrt{2}}{2}$, closest to $13\sqrt{2}$

d. The answers for part b and c are different. Although we are using the same classification method (1-nearest neighbor), the reduction of data from 2-D to 1-D along the first principal component line ($y=x$) results in a different representation of the distances between points. What was the closest point for one point before dimensionality reduction might not be the same after. The results in part b and c show this discrepancy.

## Problem 2: Exact Recovery of a Linear Compression Scheme

In this exercise we show that in the general case, exact recovery of a linear compression scheme is impossible.

a. Let $A \in \mathbb{R}^{n,d}$ be an arbitrary compression matrix where $n \leq d-1$. Show that there exists $\mathbf{u}, \mathbf{v} \in \mathbb{R}^d, \mathbf{u} \neq \mathbf{v}$, such that $A\mathbf{u} = A\mathbf{v}$.

*Hint: Show that there exists $u \neq 0, v = 0$ such that $Au = Av = 0$.*

*Hint: Consider using the rank-nullity theorem.*

b. Conclude that exact recovery of a linear compression scheme is impossible.

a. By rank-nullity, $\text{rank}(A) + \text{nullity}(A) = d$.

Since $A$ has $n$ rows, its rank satisfies

$$\text{rank}(A) \leq n$$
$$\leq d-1 \quad \text{since } n \leq d-1$$

If we substitute this bound into the rank-nullity theorem,

$$\text{nullity}(A) = d - \text{rank}(A)$$
$$\geq d - (d-1)$$
$$\geq 1$$

Since the nullity of $A$ is $\geq 1$, this means by definition that there exists a nonzero vector $u \in \mathbb{R}^d$ such that $Au = 0$. We assumed that $Av = 0$, and $u \neq v$, so then we have $Au = Av = 0$ s.t. $u \neq v$.

□

b. From the above proof, we reasoned that $\text{null}(A) \geq 1$. Suppose there exists a linear recovery map $R$ such that $R(Ax) = x$ for all $x \in \mathbb{R}^d$, $R: \mathbb{R}^n \to \mathbb{R}^d$.

From linearity, we know that $A(x+u) = Ax + Au$

$$= Ax + 0 \quad \text{from part a, } \exists\, u \neq 0 \text{ s.t. } Au = 0 \text{ since } \text{null}(A) \geq 1$$
$$= Ax$$

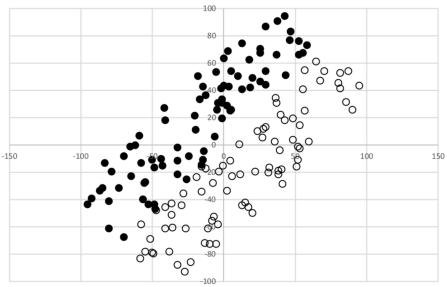
This would mean that $R(A(x+u)) = R(Ax + Au) = R(Ax) = x$.

However, linearity of $R$ requires that $R(A(x+u)) = x+u$.

$x = x+u \Rightarrow u = 0$. However, $u \neq 0$, so this is a contradiction. Therefore, no such $R$ can exist, and, therefore, exact recovery is impossible.

**Problem 3: Limitations of PCA (20 points)**

Consider the following two-dimensional dataset:




a. Describe one type of machine learning model that would classify this data well. Explain your reasoning.

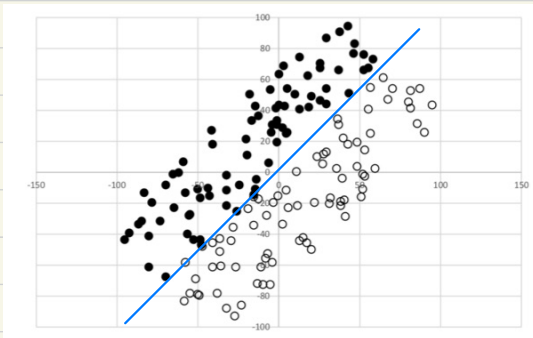
b. Draw the first principal component on the graph above. If PCA was used to reduce this data to one dimension, would the machine learning model from part (a) still classify the data well? Why or why not?

c. Is it possible to project the above data into a one-dimensional linear subspace in which the data remains linearly separable? If so, draw the subspace on the graph above. If not, explain your reasoning.

d. What does this tell example tell you about the limitations of PCA when used to pre-process data before classification?

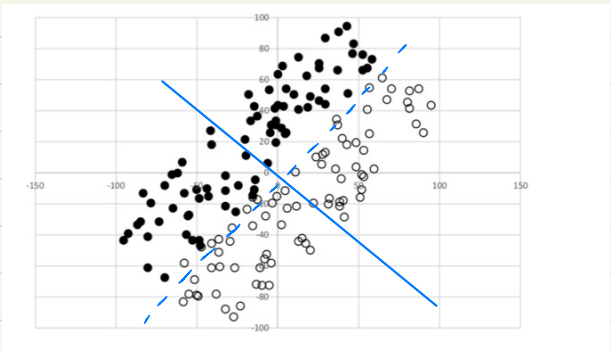
a. A linear support vector machine would classify this data well. The data seems clearly linearly separable (the boundary between the two labels can be defined as a line separating the data into two halfspaces), therefore a linear svm would work well.

b.




If we use PCA to reduce the data to 1-D, it would be hard for the linear svm to still correctly classify the data. If we project the original points onto the first principal component, the labels would be randomly scattered on the line, resulting in a not linearly separable dataset. Therefore, it wouldn't help the linear svm at all.

c.




Yes, it is possible to project the data into a 1-D subspace s.t. the data remains linearly separable. As shown above, reducing the data into 1-D by projecting them onto this principal component would preserve separability. The data would be separated, for example, by something like the dotted line.

d. PCA is unsupervised, meaning that it ignores class labels and, instead, maximizes variance. As seen in this problem, if there is a clearly linearly separable dataset, the first principal component aligns with the direction of greatest total variance, which may or may not separate classes (in this case, it doesn't). Therefore, PCA is risky for classification pre-processing since it may erase discriminative features and keep irrelevant high-variance noise.