Milad Nourian
ECE 131
HW3

P1)a)

$$k_{new}(\underline{x},\underline{y}) = \left(\sum_{i=1}^{n} \sqrt{x_i}\sqrt{z^i}\right)^d = \left(\left(x^T\right)^{\frac{1}{2}}\left(z^{\frac{1}{2}}\right)\right)^d$$

let $(x^T)^{\frac{1}{2}} = (x')^T$, $z^{\frac{1}{2}} = z' \Rightarrow g(x) = (x^{\frac{1}{2}})$

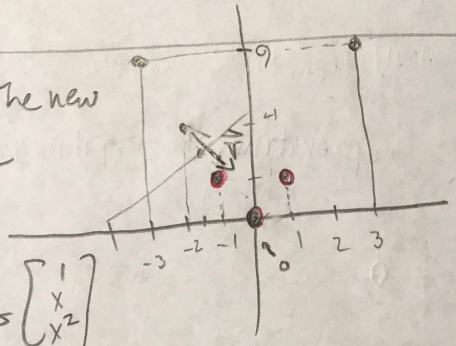
$\Rightarrow k_{new}(\underline{x},\underline{y}) = (\underline{x}'^T \underline{z}')^d \Rightarrow$ which is a valid kernel ✓ (we know if $k(x,z)$ is valid, $k_{new}(x,z) = k(g(x),g(z))$)

P1) $\Rightarrow$ method 1:

k_1 and k_2 are valid kernels $\Rightarrow$ so $\underline{\underline{K}}_1$ & $\underline{\underline{K}}_2$ are both semidefinite,

$\underline{\underline{K}} = \alpha \underline{\underline{K}}_1 + \beta \underline{\underline{K}}_2$ $(\alpha,b>0)$, $\underline{\underline{K}}$ is also semi-definite, so k. also is a valid kernel. $\Rightarrow a^T \underline{\underline{K}} a \geq 0$

P2)a) visualize the new feature space



$w = \begin{bmatrix} w_0 \\ w_1 \\ w_2 \end{bmatrix}$, $\phi(x) = \begin{bmatrix} 1 \\ x \\ x^2 \end{bmatrix}$

1) The hyperplane must pass through between $(-1,1)$, $(-2,4) \Rightarrow$ mid$p = \left(\frac{-3}{2}, \frac{5}{2}\right)$

2) It has to be normal to the line connecting $(-1,1)$, $(-2,4)$ $\Rightarrow$ slope $m_1 \cdot m_2 = -1$

$\vec{w} = \begin{bmatrix} w_1 \\ w_2 \end{bmatrix} = \begin{bmatrix} -1 \\ 3 \end{bmatrix}$ (normal vector of the hyperplance). we know, $w_1 y_1 + w_2 y_2 + w_0$

$(-1)\left(\frac{-3}{2}\right) + (3)\left(\frac{5}{2}\right) + w_0 = 0 \Rightarrow w_0 = -9 \Rightarrow \boxed{-1y_1 + 3y_2 - 9 = 0}$ ✓

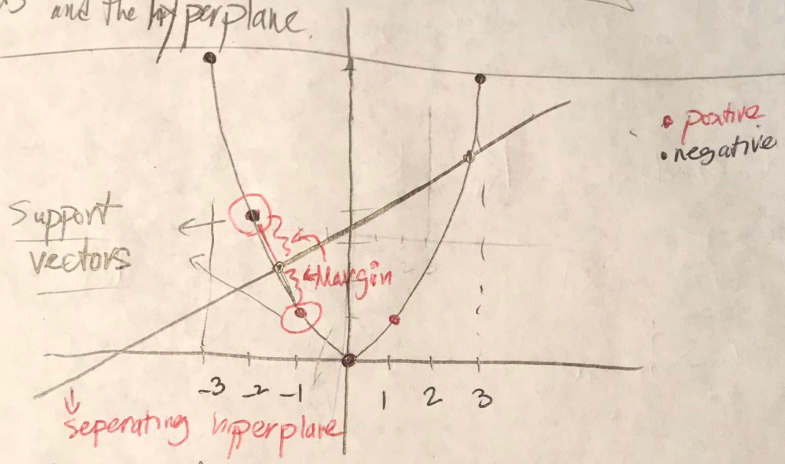
P2)a) Hyperplane seperating with max. Margin is:

$-y_1 + 3y_2 - 9 = 0 \Rightarrow$ we used the fact that $(-1,1)$ and $(-2,2)$ are closest points for classification.

$(-\frac{3}{2}, \frac{5}{2})$

Margin $= \min_n d(x_n, \text{Hyperplane})$ → distance. Margin is distance between midpoint and $(-1,1) \Rightarrow$ Margin $= \sqrt{(\frac{1}{2})^2 + (\frac{3}{2})^2} = \sqrt{\frac{10}{2}} \Rightarrow$ Margin $\Rightarrow$ Smallest distance between points and the hyperplane.

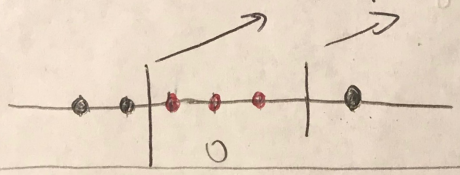
P2)b)



Support vectors are data points that would lead to the same model, even if we remove all non-support vectors.

P2)c)

seperating hyperplanes in $\mathbb{R}^1$



P2)d)

P2)d)

The primal problem we want to solve is:

$$\min \frac{1}{2}\|w\|^2 + C\sum_n \varepsilon_n$$

subject to constraint: $1-\varepsilon_n - y_n[w^T\phi(x)+b] \leq 0$

Take lagrangian $-\varepsilon_n \leq 0$

$$\mathcal{L}(\alpha, \beta, w, \varepsilon_n) = \frac{1}{2}\|w\|^2 + C\sum_n \varepsilon_n + \sum_n \alpha_n (1-\varepsilon_n - y_n[w^T\phi(x)+b]) + \sum_{n=?} (-\varepsilon_n)$$

Now: Take gradients. $\frac{\partial \mathcal{L}}{\partial w} = w - \sum \alpha_n y_n \phi(x_n) = 0 \Rightarrow w^* = \sum \alpha_n y_n \phi(x_n)$

we can now rewrite $h(x) = \text{sign}(w^T\phi(x)+b)$ and plug in w^* for w.

$$h(x) = \text{sign}\left(\sum_n \alpha_n y_n \phi(x_n)\phi(x) + b\right)$$

$$h(x) = \text{sign}\left(\sum_n \alpha_n y_n K(x_n, x) + b\right) = \text{sign}(w^T x + b)$$

1) We have already found $w^*_{optimal}$ to maximize the margin.

2) From class derivations, we know none_support vectors have $\alpha_n = 0$

$w^* = w_{opt} = \begin{bmatrix} -1 \\ 3 \end{bmatrix} \Rightarrow$ we know $w^* = \sum \alpha_n y_n \phi(x_n)$

$$= \begin{bmatrix} -1 \\ 3 \end{bmatrix} = \alpha_1 (-1) \begin{bmatrix} -2 \\ 4 \end{bmatrix} + \alpha_2 (+1) \begin{bmatrix} -1 \\ 1 \end{bmatrix}$$

$$= \alpha_1 \begin{bmatrix} 2 \\ -4 \end{bmatrix} + \alpha_2 \begin{bmatrix} -1 \\ 1 \end{bmatrix}$$

$\boxed{\alpha_1 = -1, \alpha_2 = -1} \Rightarrow h(x) = \text{sign}\left(\sum_n \alpha_n y_n K(x_n, x) + b\right)$

2) e) If we add a negative point at $x=0.5$, the transformed feature vector is not linearly seperable anymore, that means that we need to use a soft margin instead of a hard margin by using slack variables.. So, yes, since $x=0.5$ will be a support vector, the hyperplane can change since now we have to compensate for the point at $x=0.5$. Also, the margin will decrease as a support vector is added to the training set. (It is possible for margin to also stay the same).

---

3) a)

1) Hard margin problem (no slack variables)

2) $b=0$

$$\min \frac{1}{2}\|w\|^2 \qquad \underline{w} = \begin{bmatrix} w_1 \\ w_2 \end{bmatrix}$$

$$\text{s.t. } g_n w^T x_n \geq 1, \text{ for } n = 1, \dots, N$$

$$\underline{x} = \begin{bmatrix} a \\ e \end{bmatrix} \text{ constraint} \Rightarrow (-1)\left[w_1 a + w_2 e\right] \geq 1$$

$$w_1 a + w_2 e \leq -1$$

$\Rightarrow$ So problem: $\min \frac{1}{2}\|w\|^2 = \frac{1}{2} w_1^2 + w_2^2$

$$w_1 a + w_2 e + 1 \leq 0$$

use duality to find this:

Lagrangian $\mathcal{L}(w, a) = \frac{1}{2} w_1^2 + \frac{1}{2} w_2^2 + \alpha (w_1 a + w_2 e + 1)$

$\frac{\partial \mathcal{L}}{\partial w_1} = w_1 + \alpha a = 0 \Rightarrow w_1 = -\alpha a$

$\frac{\partial \mathcal{L}}{\partial w_2} = w_2 + \alpha e = 0 \Rightarrow w_2 = -\alpha e$

$$\mathcal{L}(\alpha) = \left(\frac{1}{2}\right)\alpha^2 a^2 + \frac{1}{2}\alpha^2 e^2 + \alpha\left[1 - (\alpha a)a + (-\alpha e)e + 1\right]$$

$$\mathcal{L}(\alpha) = \frac{1}{2}\alpha^2 a^2 + \frac{1}{2}\alpha^2 e^2 - \alpha^2 a^2 - \alpha^2 e^2 + \alpha$$

$$= -\frac{1}{2}\alpha^2 a^2 - \frac{1}{2}\alpha^2 e^2 + \alpha$$

$\Rightarrow$ find $\alpha \Rightarrow \frac{\partial \mathcal{L}}{\partial \alpha} = \left[\frac{-1}{2}\right][2\alpha]a^2 - \left[\frac{1}{2}\right][2\alpha][e^2] + 1 = 0$

$$\alpha[-a^2 - e^2] = -1$$

$$\alpha = \frac{1}{a^2 + e^2}$$

$$w_1 = -\alpha a = \frac{-a}{a^2+e^2}$$

$$w_2 = -\alpha e = \frac{-e}{a^2+e^2}$$

$$\underline{w}^* = \begin{bmatrix} \frac{-a}{a^2+e^2} \\ \frac{-e}{a^2+e^2} \end{bmatrix}$$

---

3) b) $\min \frac{1}{2}\|w\|^2 \Rightarrow \min \frac{1}{2}w_1^2 + \frac{1}{2}w_2^2$: objective

$y_1 w^T x_1 \geq 1$

$y_2 w^T x_2 \geq 1$

constraint: 1) $w_1 + w_2 \geq 1$ 1) $[w_1 + w_2]$

2) $(-1)[w_1] \geq 1$

≤ -1

$w_1 < -1$

opt. problem: $\min \frac{1}{2}w_1^2 + \frac{1}{2}w_2^2$

$-w_1 - w_2 + 1 \leq 0$

$+w_1 + 1 \leq 0$

$$\mathcal{L}(\alpha_1, \alpha_2 / w_1, w_2) = \inf\left(\frac{1}{2}w_1^2 + \frac{1}{2}w_2^2 + \alpha_1(-w_1 - w_2 + 1) + \alpha_2(+w_1 + 1)\right)$$

$$\frac{\partial \mathcal{L}}{\partial w_1} = w_1 - \alpha_1 + \alpha_2 = 0 \Rightarrow w_1 = \alpha_1 - \alpha_2$$

$$\frac{\partial \mathcal{L}}{\partial w_2} = w_2 - \alpha_1 = 0 \Rightarrow w_2 = \alpha_1$$

3)b) continued:

$$\mathcal{L}(\alpha_1,\alpha_2) = \frac{1}{2}(\alpha_1^2 + \alpha_2^2 + 2\alpha_1\alpha_2) + \frac{1}{2}\alpha_1^2 + \alpha_1\alpha_2 - \alpha_1 + 1) + \alpha_2(+\alpha_1 - \alpha_2 + 1)$$

$$= -\alpha_1^2 + \frac{1}{2}\alpha_2^2 - \alpha_1\alpha_2 + \alpha_1\alpha_2 + \alpha_1 + \alpha_1\alpha_2 - \alpha_2^2 + \alpha_2$$

$$\mathcal{L}(\alpha_1,\alpha_2) = -\alpha_1^2 - \frac{1}{2}\alpha_2^2 + \alpha_1 + \alpha_2 + \alpha_1\alpha_2$$

$$\frac{\partial \mathcal{L}}{\partial \alpha_1} = -2\alpha_1 + 1 + \alpha_2 = 0$$

$$\frac{\partial \mathcal{L}}{\partial \alpha_2} = -\alpha_2 + 1 + \alpha_1 = 0$$

$$\rightarrow \quad 2 - \alpha_1 = 0 \Rightarrow \boxed{\alpha_1 = 2}$$

$$\alpha_2 = 3$$

$$w_1 = \alpha_1 - \alpha_2 \rightarrow w_1 = -1$$

$$w_2 = \alpha_1 \rightarrow w_2 = 2$$

$$\Rightarrow w^* = \begin{bmatrix} -1 \\ 2 \end{bmatrix} \quad \checkmark$$

---

3)c) If we allow offset:

Problem $\min \frac{1}{2}w_1^2 + \frac{1}{2}w_2^2$

s.t. $-w_1 - w_2 - b + 1 \leq 0$

$+w_1 + b + 1 \leq 0$

Solve the problem same way (find dual)

$$\mathcal{L}(w_1, w_2, \alpha_1, \alpha_2) = \frac{1}{2}w_1^2 + \frac{1}{2}w_2^2 + \alpha_1(-w_1 - w_2 - b + 1) + \alpha_2(+w_1 + b + 1)$$

$$\frac{\partial \mathcal{L}}{\partial w_1} = w_1 - \alpha_1 + \alpha_2 = 0 \Rightarrow w_1 = \alpha_1 - \alpha_2$$

$$\frac{\partial \mathcal{L}}{\partial w_2} = w_2 - \alpha_1 = 0 \rightarrow w_2 = \alpha_1$$

$$\frac{\partial \mathcal{L}}{\partial b} = -\alpha_1 + \alpha_2 = 0 \Rightarrow \alpha_1 = \alpha_2$$

$\mathcal{L}(w, \alpha_1, \alpha_2, b) = \left[\frac{1}{2}\right]\left[\alpha_1^2 + \alpha_2^2 - 2\alpha_1\alpha_2\right] + \frac{1}{2}\alpha_1^2 + \alpha_1\left[\alpha_2 - \alpha_1 - \alpha_1 - b + 1\right] + \alpha_2\left[\alpha_1 - \alpha_2 + b + 1\right]$

$= \alpha_1^2 + \frac{1}{2}\alpha_2^2 - \alpha_1\alpha_2 + \alpha_1\alpha_2 - 2\alpha_1^2 - \alpha_1 b + \alpha_1 + \alpha_1\alpha_2 - \alpha_2^2 + \alpha_2 b + \alpha_2$

$\mathcal{L}(\alpha) = -\alpha_1^2 - \frac{1}{2}\alpha_2^2 - \alpha_1 b + \alpha_2 b + \alpha_1 + \alpha_2 + \alpha_1\alpha_2$

$\frac{\partial \mathcal{L}}{\partial \alpha_1} = -2\alpha_1 - b + 1 + \alpha_2 = 0 \Rightarrow -2\alpha_1 + 1 + 1 + \alpha_1 = 0 \Rightarrow \boxed{+\alpha_1 = 2}$

$\frac{\partial \mathcal{L}}{\partial \alpha_2} = -\alpha_2 + b + 1 + \alpha_1 = 0 \qquad \boxed{\alpha_2 = 3 + b}$

From $\frac{\partial \mathcal{L}}{\partial b} = -\alpha_1 + \alpha_2 = 0 \rightarrow \alpha_1 = \alpha_2$

$2 = 3 + b \Rightarrow \boxed{b = -1}$

$w^* = \begin{bmatrix} 0 \\ 2 \end{bmatrix}, b^* = -1$

Check if it classifies the training data correctly:

$x = \begin{bmatrix} 1 \\ 1 \end{bmatrix} \rightarrow 2 + (-1) = +1$ ✓

$x = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \rightarrow 0 + (-1) = -1$ ✓

Since our objective function is $\min \frac{1}{2}\|w\|_2^2$, we see that

with b: $w = \begin{bmatrix} 0 \\ 2 \end{bmatrix}$

without b: $w = \begin{bmatrix} -1 \\ 2 \end{bmatrix}$

$\Rightarrow$ so with b, we get a classifier with bigger margin (bias gives us a larger margin and a more robust classifier)

4.2)b)

It is very helpful for our SVM model to keep the proportion of classes (in this cases +1 and -1) the same. **The reason** is that if we train our model the same proportion of +1 and -1, the data is better representative of the whole and generalizes better. Stratification helps with this issue. In addition, keeping the same proportion across different classes of CV helps to obtain a better bias and variances (lowers the bias and variance). For doing so, I used the Stratified_KFold which returns the indices of the training set and test set and used a for loop to loop over these data points.

4.2)d)

C	*Accuracy*	*F1-Score*	*AUROC*	*Precision*	*Sensitivity*	*Specificity*
10^{-3}	0.65357641	0.790497935	0.5	0.6535764	1.0	0.0
10^{-2}	0.782167629	0.849684211	0.71093407	0.7736231	0.94265087	0.47921727
10^{-1}	0.807183716	0.8584263172	0.76878821	0.8264614	0.89344687	0.64412955
1	0.82497793	0.8675405697	0.80188309	0.8584381	0.87704553	0.72672064
10^{1}	0.81957281	0.8636329571	0.79531754	0.8535799 6	0.8743058	0.71632928
10^{2}	0.819572812	863632957148	0.79531754	0.8535799	0.8743058	0.71632928
Best C	1.0	1.0	1.0	1.0	0.001	1.0

How does the 5-fold CV performance vary with C and the performance metric?

Score trend for C is as follows: for small C value, the score is low. As C is increased, the score increases, and the optimal value of C is found at C=1.0. After that, increasing the C value, results in lower score and that's an indication that we have the best C value across the different C values.

4.3)a)

Describe the role of the additional hyperparameter γ for an RBF-kernel SVM. How does γ affect generalization error?

Gamma hyper-parameter is defined as the inverse of radius of influence of a training data point. To illustrate the effect, let's take 2 extremes. If the value of gamma is large, it means that the radius of influence is small and other points do not have as much effect. This can lead to overfitting. If gamma is small, then radius of influence includes the entire training set and this means that the model would not be able to capture the complex shapes and therefore under

fits (high bias and low variance and lower generations error but does poorly on the training data).

4.3)b) Explain what kind of grid you used and why.

For finding the optimal values of C and gamma, I used the brute force grid search to find the optimal values (highest score) for a given measurement metric. I used logspace values for both C and gamma in the range -10^2, up to 10^2 separated by a factor of 10, eg.

```
C_range = np.logspace(-2, 2, 5)
gamma_range = np.logspace(-2, 2, 5)
```

The reason I chose brute force grid search was to find out for different values of c and gamma from very large to very small values, how the behavior and score changes. Also, initially I tried running the algorithm with 13 values for c and gamma, but this resulted in a very slow training and so I decided to use only 5 values for C and gamma. Brute-force allows me to go over each of the combinations of c and gamma and choose the best value, and this ensures that we do find the score on each of the combinations (this results in O (N^2) computations since we have a nested loop), whereas, random search might miss the optimal value. This is why I chose brute force.

4.3)c)

Metric	*Score*	*C*	*γ*
Accuracy	0.826859318231	10.0	0.01
F1-Score	0.870676578136	10.0	0.01
AUROC	0.798475998043	10.0	0.01
Precision	0.851925647452	100.0	0.01
Sensitivity	1.0	0.01	0.01
Specificity	0.711336032389	100.0	0.01

How does the CV performance vary with the hyperparameters of the RBF-kernel SVM?

Since we are using rbf kernel of SVM and try to optimize both of the hyperparameters gamma

and C, we must consider both of these values at the same time (i.e. hyperparameter space). We saw in the CV performance of the model that for small values of gamma and C, the score is relatively low. Increasing the C and gamma values, we found that for most of the metrics, C = 10 and gamma = 0.01 provides the best score and, thus, we can use these hyperparameter values to train our model. For large C and gamma values, we saw that the CV performance (score) of the model dropped.

4.4)a) Based on the results you obtained in Section 0.2 and Section 0.3, choose a hyperparameter setting for the linear-kernel SVM and a hyperparameter setting for the RBF-kernel SVM. Explain your choice.

For linear model SVC I chose C = 1.0 and for rbf kernel SVC I chose C = 10.0 and gamma = 0.01 as we found these optimal values from the CV performance that produce the highest score. I used the C value from 4.2)d) for the linear SVM and used C and gamma for 4.3)C) for the RBF kernel.

4.4)C)

Linear SVM:

Metric	*Test Score*
Accuracy	0.8571428571428571
F1-Score	0.9
AUROC	0.8493589743589743
Precision	0.9375
Sensitivity	0.8653846153846154
Specificity	0.8333333333333334

RBF SVM:

Metric	*Test Score*
Accuracy	0.9142857142857143
F1-Score	0.9433962264150944
AUROC	0.8696581196581197
Precision	0.9259259259259259
Sensitivity	0.9615384615384616
Specificity	0.7777777777777778

How do the test performance of your two classifiers compare?

Looking at the test performance of the linear SVM and rbf SVM, one can see that RBF kernel outperforms the linear model on **accuracy**, **F1_score** and **AUROC** and **sensitivity**. On the other hand, rbf SVM performs better on **specificity** and **precision**. In general, RBF performs better

than linear model. This is can be due to using a better kernel and also using a richer feature set. So our RBF model can classify the tweets as positive (+1) and negative (-1) with 91.5% accuracy (which is reasonable). Also, RBF kernel of SVM is usually better at capturing non-linearity of the features.