ESTUDIANTES

Camilo Hernández Guerrero

Samy Felipe Cuestas Merchán

PROFESOR

Randy Darrell Lancheros Molano

Tecnologías Digitales Emergentes

PROYECTO I

PONTIFICIA UNIVERSIDAD JAVERIANA

BOGOTA D.C.

SEPTIEMBRE 4

2022

1. *Descripción del problema*

Se tiene un conjunto de datos tomados de una base de datos de Kaggle[1] donde lo que se busca con estos datos es poder hacer una clasificación de los datos presentes y una predicción de estos. Para esto se consideró el algoritmo de Support Vector Machine (SVM) debido a que lo que se buscaba hacer con la clasificación, la predicción y que es supervisado ya que los datos vienen clasificados con etiquetas, esto hace que encajen en este. Además se va a comparar entre los kernels sigmoide, lineal y RBF.

2. *Descripción del conjunto de datos*

El dataset seleccionado corresponde a datos sobre cultivos donde lo que se busca obtener de estos es una predicción de un posible cultivo en un suelo y un ambiente característico, para esto se tienen los datos de porcentaje de nitrógeno (N), fósforo (P), potasio (K) y ph del suelo (ph), además se tienen los datos de la temperatura en grados celsius (temperature), humedad en porcentaje (humidity) y la cantidad de lluvia en milímetros (rainfall). Los datos están clasificados por etiquetas y cada una corresponde a los datos que fueron tomados lo que permite que el algoritmo usado sea supervisado.

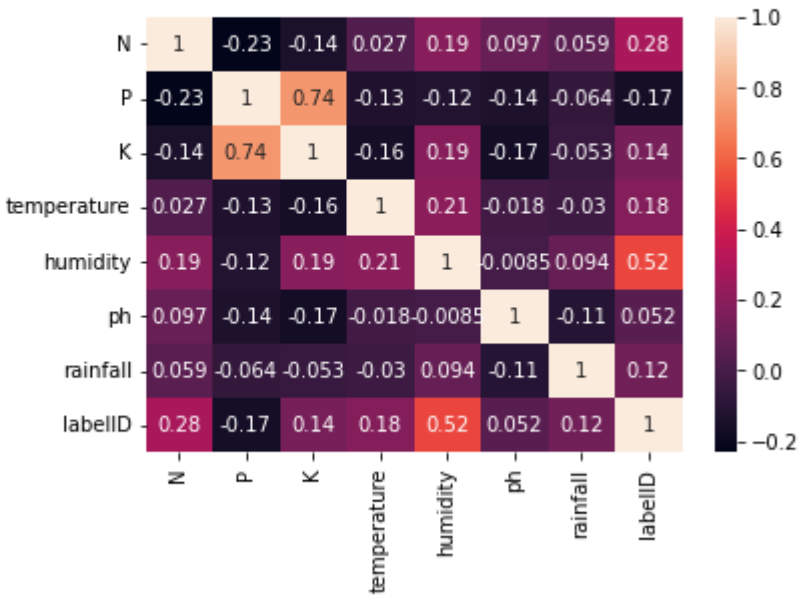
Los datos del dataset pertenecen a datos de ambiente de la India.

3. *Selección y justificación de las variables a utilizar en el desarrollo*

Se van a utilizar como variables independientes las siguientes variables:

- P: La proporción del contenido de fósforo en el suelo.
- temperature: Temperatura en grados celsius.
- humidity: Humedad relativa en porcentaje.
- ph: Valor del ph en el suelo.
- rainfall: Lluvia en milímetros.

```
In [40]: sns.heatmap(df_merge.corr(), annot = True)
Out[40]: <AxesSubplot:>
```



Dichas variables independientes fueron seleccionadas porque al realizar un mapa de calor con la correlación entre las variables vemos que la correlación entre P y K es muy alta, con un 74% por lo que se evitó elegir ambas para el modelo. Por otro lado se puede observar que las demás correlaciones entre variables son muy bajas, por lo que se pueden elegir sin temor a afectar negativamente el modelo. Tampoco se eligió la variable “N” porque el modelo habría caído en overfitting fácilmente.

Como variable dependiente utilizaremos el label, que es el alimento al que corresponden las demás columnas, como variable dependiente fue elegida porque es la variable más interesante para predecir, además de que es la única que no es cuantitativa.

4. Análisis de los resultados obtenidos

Para la experimentación se miraron los diferentes tipos de kernels vistos en clase, los cuales eran RBF, sigmoide y lineal.

La solución con kernel lineal obtuvo los siguientes resultados:

```
In [42]: SVM = SVC(kernel = 'linear', random_state = 0)
         SVM.fit(Xtrain, Ytrain)

         predicted_values = SVM.predict(Xtest)

         accuracy = accuracy_score(Ytest, predicted_values)

         print("SVM's with linear kernel accuracy: ", accuracy, "\n")

         SVM's with linear kernel accuracy:  0.9295454545454546
```

La solución con kernel RBF obtuvo los siguientes resultados:

```
In [44]: SVM = SVC(kernel = 'rbf', random_state = 0)
         SVM.fit(Xtrain, Ytrain)

         predicted_values = SVM.predict(Xtest)

         accuracy = accuracy_score(Ytest, predicted_values)

         print("SVM's with rbf kernel accuracy: ", accuracy, "\n")

         SVM's with rbf kernel accuracy:  0.9431818181818182
```

La solución con kernel sigmoide obtuvo los siguientes resultados:

```
In [43]: SVM = SVC(kernel = 'sigmoid', random_state = 0)
         SVM.fit(Xtrain, Ytrain)

         predicted_values = SVM.predict(Xtest)

         accuracy = accuracy_score(Ytest, predicted_values)

         print("SVM's with sigmoid kernel accuracy: ", accuracy, "\n")

         SVM's with sigmoid kernel accuracy:  0.6386363636363637
```

5. *Análisis del desempeño de la solución*

Se puede apreciar que Support Vector Machine en general, sin importar el kernel que le pongamos, arroja resultados al menos aceptables, con un mínimo de 63.86% según el índice de precisión en el caso del kernel sigmoide. Sin embargo, los mejores resultados se evidencian utilizando los otros dos kernel, el lineal y el RBF, donde el RBF toma la delantera con apenas aproximadamente 1.37% mejor precisión para un total de 94.32%.

Dado que el índice de precisión es alto pero todavía está lejos de la perfección, se puede decir con seguridad que el modelo planteado no cae en overfitting y mucho menos en underfitting, por lo que el modelo debería ser capaz de predecir con muy buen desempeño algún otro dataset que se ajuste.

```
cm = confusion_matrix(Ytest, predicted_values)
print(cm)
```

```
[[13  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0]
 [ 0 17  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0]
 [ 0  0 14  0  0  0  0  0  0  0  2  0  0  0  0  0  0  0  0  0  0]
 [ 0  0  0 21  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0]
 [ 0  0  0  0 21  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0]
 [ 0  0  0  0  0 21  0  0  1  0  0  0  0  0  0  0  0  0  0  0  0]
 [ 0  0  0  0  0  0 20  0  0  0  0  0  0  0  0  0  0  0  0  0  0]
 [ 0  0  0  0  0  0  0 18  0  0  0  0  0  0  0  0  0  0  0  0  0]
 [ 0  0  0  0  0  0  0  0 17  0  0  0  0  0  1  0  0  0  0  0 10  0]
 [ 0  0  0  0  0  0  0  0  0 14  0  0  0  0  0  0  0  0  0  0  0  0]
 [ 0  0  1  0  0  0  0  0  0  0 18  3  0  0  0  0  0  0  1  0  0  0]
 [ 0  0  0  0  0  1  1  0  1  0  2 16  0  0  0  0  0  0  0  0  0  0]
 [ 0  0  0  0  0  1  0  0  0  0  0  0 25  0  0  0  0  0  0  0  0  0]
 [ 0  0  1  0  0  1  0  0  0  0  0  0  2 12  0  0  0  0  3  0  0  0]
 [ 0  0  0  0  0  0  0  0  0  0  0  0  0  0 24  0  0  0  0  0  0  0]
 [ 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 22  0  0  0  0  0  1]
 [ 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 29  0  0  0  0  0]
 [ 0  0  0  0  0  0  0  0  1  0  0  0  0  0  0  0  0 18  0  0  0  0]
 [ 0  0  0  0  0  0  0  0  0  0  0  0  0  1  0  0  0  0 17  0  0  0]
 [ 0  0  0  0  1  0  0  0  0  0  0  0  0  0  0  0  0  0 15  0  1]
 [ 0  0  0  0  0  0  0  0  5  0  0  0  0  0  0  0  0  0  0 11  0]
 [ 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 15]]
```

También se evidencia de la matriz de confusión que los casos que son falsos positivos (no pertenecientes a la diagonal) son muy pocos comparados a los reales en la diagonal. Los verdaderos positivos son un total de 392, mientras que los falsos positivos no son más de 30.

6. Conclusiones

Tras la realización del ejercicio se pudo apreciar que para encontrar un mejor rendimiento no solo es necesario elegir un algoritmo de aprendizaje de máquina si no que también es igual de importante revisar los distintos kernels que se pueden aplicar ya que esto puede ayudar a encontrar un mejor resultado el cual permite tener una mejor predicción o clasificación de los datos.

7. Bibliografía

[1] https://www.kaggle.com/datasets/atharvaingle/crop-recommendation-dataset/code