UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

Facultad de Estudios Superiores
Acatlán

Practica 4:

Bosques, regresión logística y KNN

Aprendizaje de maquina

Integrantes:

Gustavo Adolfo Alvarez Hernández

Profesor:

Eduardo Eloy Loza Pacheco

FES Acatlán, 26 de septiembre 2022

Objetivo

Que el alumno aprenda a utilizar algoritmos de aprendizaje conjunto y note la mejoría en desempeño respecto a los modelos simples; por otro lado, que aprenda a manejar problemas de clasificación lineales y no lineales.

Materiales

- Google colab: Utilizando Sklearn, pandas y matplotlib.

Resumen

En la presente práctica resolveremos un problema de predicción por medio de bosques de regresión sobre los datos de salarios y así obtener una estimación del salario con respecto a la experiencia del individuo.

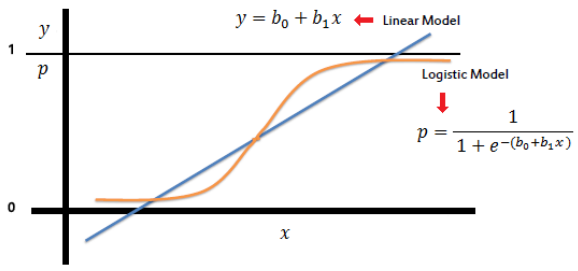
Por otra parte, se aborda el problema de clasificar si los usuarios compran o no cierto producto que se oferta en una campaña por redes sociales, tomando como variables la edad y el salario del usuario; por lo cual se implementa una regresión lineal y KNN para dividir las dos clases y contrarrestar el desempeño de los dos modelos.

Antecedentes

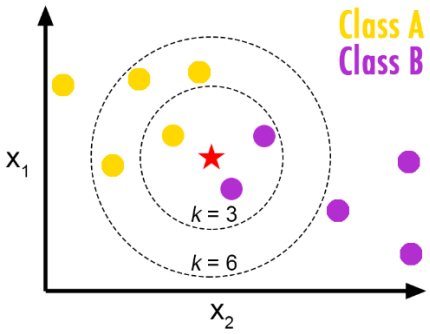
Aquí introducimos el concepto de aprendizaje conjunto (Ensemble learning), en el cual se entrena más de un modelo sobre los mismos datos, se genera un aglomerado con ellos y al momento de realizar la tarea, cada uno de nuestros modelos realiza la predicción, para después en conjunto dar una mejor aproximación; esto se suma a un proceso de arranque (Bootstrapping) que consiste en generar divisiones de los datos, para alimentar a los distintos modelos y generar modelos distintos.

En nuestro caso en particular, es un bosque de regresión aleatorio que consiste en muchos arboles de regresión y al momento de pedir una estimación se le pide a los n arboles su predicción y se promedia para dar el resultado final.

Por su parte, cuando nos enfrentamos a problemas de clasificación (esto es, nuestra variable a predecir es categórica) podemos emplear varios modelos, entre los cuales tenemos a la regresión logística; la cual consiste en una regresión que alimenta al modelo dando sus estimaciones, posteriormente se pasan por una función de activación (usualmente la función sigmoide) arrojando una probabilidad, por ultimo se establece un umbral para saber a qué categoría cae dada la probabilidad que obtuvimos.



Por último, el algoritmo de KNN, se basa en la idea de que los datos de la misma clase se distribuyen de la misma forma, por lo cual los datos sin clasificar se pueden catalogar por los datos que están cercanos a ellos; de esta forma trabaja KNN, dado un nuevo dato busca a los k datos más cercanos a él (por medio de alguna métrica) y se ve a que clase corresponde la mayoría de los k vecinos, siendo el resultado la clase la cual se asignar a este nuevo dato.



Desarrollo

1. Cargamos los módulos correspondientes para el bosque de regresión.

```
[ ]  import pandas as pd
     import numpy as np
     import matplotlib.pyplot as plt
     from sklearn.ensemble import RandomForestRegressor
```

2. Procedemos a cargar nuestros datos de los salarios.

```
data= pd.read_csv("Salarios.csv")
X= data.iloc[:,1:2].values
y= data.iloc[:,2].values
```

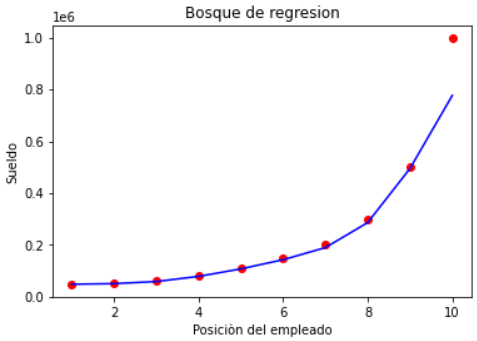
3. Y procedemos a instanciar al bosque de regresión.

```
[ ]  regresion=RandomForestRegressor(n_estimators=100,random_state=0)
     regresion.fit(X,y)

     RandomForestRegressor(random_state=0)
```

4. Generamos una estimación para ver el salario de alguien con 6.5 años de experiencia.

```
y_pred= regresion.predict([[6.5]])
y_pred

array([158300.])
```

5. Y visualizamos la predicción en los años que tenemos.



6. Ahora añadimos los módulos para la regresión logística y KNN

```
[ ]  import pandas as pd
     import numpy as np
     import matplotlib.pyplot as plt
     from sklearn.model_selection import train_test_split
     from sklearn.linear_model import LogisticRegression
     from sklearn.preprocessing import StandardScaler
     from sklearn.metrics import plot_confusion_matrix
     from matplotlib.colors import ListedColormap
```

7. Cargamos los datos sobre anuncios en redes sociales

```
[ ]  data= pd.read_csv("AnunciosRedesSociales.csv")
```

```
[ ]  data.head()
```

	User ID	Gender	Age	EstimatedSalary	Purchased
0	15624510	Male	19.0	19000.0	0
1	15810944	Male	35.0	20000.0	0
2	15668575	Female	26.0	43000.0	0
3	15603246	Female	27.0	57000.0	0
4	15804002	Male	19.0	76000.0	0

8. Dividimos nuestros datos en entrenamiento y test para generalizar nuestros modelos

```
[ ]  x_train,x_test,y_train,y_test= train_test_split(X,y,test_size= .25, random_state=0)
```

9. Procedemos a escalar para reducir la distancia entre los datos.

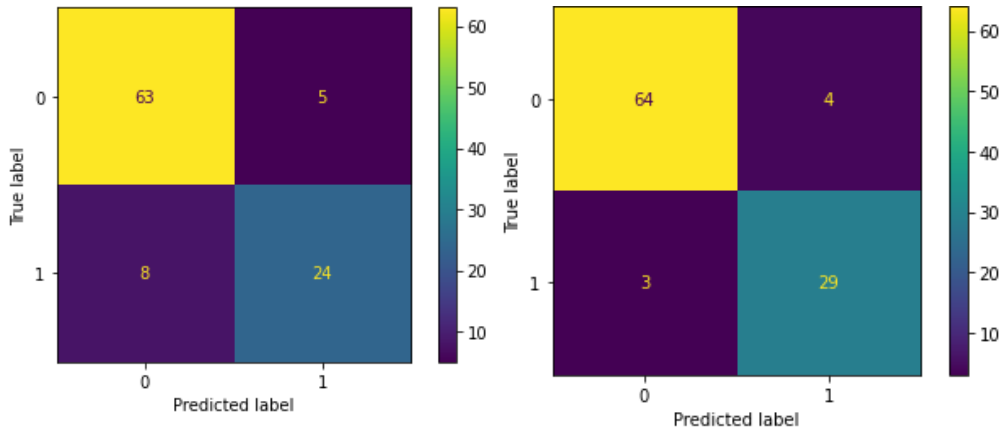
```
[ ]  scx= StandardScaler()
     x_train=scx.fit_transform(x_train)
     x_test= scx.fit_transform(x_test)
```

10. Instanciamos ambos modelos (regresión logística y KNN)

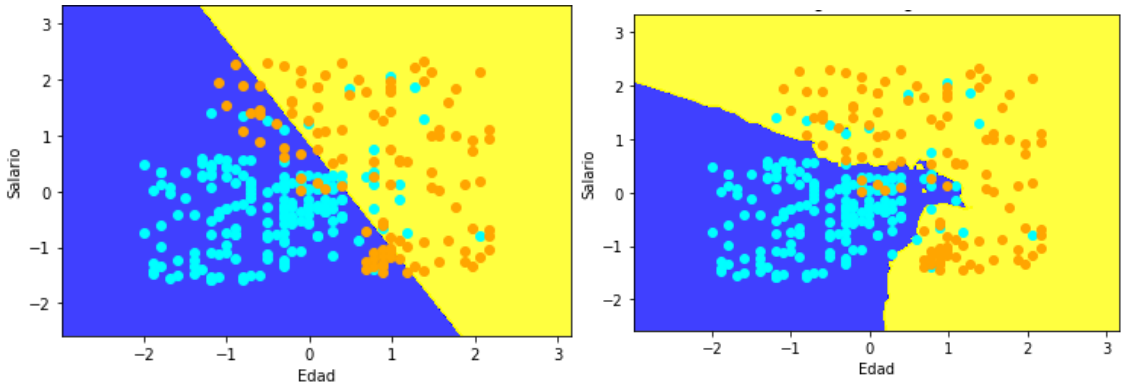
```
regLog= LogisticRegression(random_state=0)
regLog.fit(x_train,y_train)
```

```
knn= KNeighborsClassifier(n_neighbors=5, metric= "minkowski", p=2)
knn.fit(x_train,y_train)
```

11. Predecimos los datos de prueba y analizamos que tal la predicción por medio de su matriz de confusión. (logística a la izq., KNN derecha)



12. Por último, graficamos para ver como clasifican.



Conclusiones

En cuanto a los algoritmos de ensamblaje, podemos notar que hay mejor aproximación a un modelo simple, como era de esperarse al ser un aglomerado de modelos simples; aunque hay que analizar si el mejor performance es significativo en relación con el aumento en la carga computacional.

Ahor en cuanto a los problemas de clasificación se puede decir que ambos modelos tienen un desempeño bastante bueno, el hecho que KNN logre un mejor desempeño se puede ver debido a que el problema no es linealmente separable, por lo cual la regresión no tendría un performance mayor, al estar limitado por una división por una recta; aún así se tiene un resultado bastante apropiado.

Por último, hay que ser hincapié en que estos fallos también entran en lo apropiado, puesto que de no haber fallos habría un sobreajuste y nuestro modelo no sería bueno generalizado. En cuanto a los resultados son congruentes a lo esperado, esto es que se requiere una edad mayor y salario arriba de la media para tener la posibilidad de adquirir un choche y los casos que no se clasifican bien suelen ser datos anómalos.

Referencias

[1] N. Beheshti, «Towards Data Science,» 2 Marzo 2021. [En línea]. Available: https://towardsdatascience.com/random-forest-regression-5f605132d19d. [Último acceso: 26 Septiembre 2022].

[2] S. Swaminathan, «Towards Data Science,» 15 Marzo 2018. [En línea]. Available: https://towardsdatascience.com/logistic-regression-detailed-overview-46c4da4303bc. [Último acceso: 26 Septiembre 2022].