FACULTAD DE INGENIERIA
Universidad de Buenos Aires

TP parte 2 - Informe

75.06/95.58 - Organizacion de datos
Curso Collinet
Primer cuatrimestre de 2021

Alumno	Número de padrón	Email
Maria Sol Fontenla	103870	msfontenla@fi.uba.ar
Agustina Segura	104222	asegura@fi.uba.ar

Índice

1. Introduccion

Luego de la presentación del informe y el baseline FiuFip quiere profundizar su campaña de recaudación. Gracias al éxito logrado en la primera campaña la organización tiene más confianza en ustedes y sus "algoritmos" y está ansiosa por probar las avanzadas técnicas de inteligencia artificial.

2. Modelos Realizados

Cada modelo realizado tiene su correspondiente notebook

- ArbolDeDecision - ArbolDeDecision.ipynb
- KNN - Knn.ipynb
- Naive Bayes - NaiveBayes.ipynb
- Boosting - Boosting.ipynb
- Redes Neuronales - redesNeuronales.ipynb
- Regresion Logistica - regresionLogistica.ipynb
- Svm - svm.ipynb

3. Tabla de preprocesamientos

Para los distintos algortimos se realizaron distintos preprocesamientos a los set de datos

Nombre preprocesamiento	Funcionalidad	Nombre funcion python
IDF Arboles1	Se seleccionaron los features relevantes al algortimos, luego se aplico one hot encoding y se eliminaron los features irrelevantes.	ingenieriaDeFeaturesArboles1
IDF Arboles2	primero agrupa valores de algunas columnas de alta cardinalidad y luego a las variables seleccionadas se les aplica one hot encoding.	ingenieriaDeFeaturesArboles2
IFD Variables Normalizadas	A los features seleccionados, le aplica one hot encoding y luego los normaliza.	ingenieriaDeFeaturesVariablesNormalizadas
IDF SVM	selecciona los features relevantes y luego le aplica one hot encoding y los normaliza. Tambien borra los features irrelevantes.	ingenieriaDeFeaturesSVM
IDF Boosting	determina los features relevantes y luego le aplica mean Encoding.	ingenieriaDeFeaturesBoosting
IDF categoricalNB	selecciona las variables categoricas relevantes y luego le aplica codificacion ordinal.	ingenieriaDeFeaturesCategoricalNB
IDF CategoricalNB2	selecciona las variables categoricas relevantes y luego aplica mean encoding .	ingenieriaDeFeaturesCategoricalNB2
IDF GaussianNB	se queda con las variables continuas	ingenieriaDeFeaturesGaussianNB
Variables Normalizadas Mean Encoding	cuando selecciona las variables categoricas releventas aplica mean Encdoing y luego lo normaliza.	ingenieriaDeFeauturesVariablesNormalizadasME
IDF Redes	selecciona los features correspondientes y luego aplica one hot encoding y lo normaliza.	ingenieriaDeFeaturesRedes
IDF Reds2	selleciona los fecatures correspondientes y luego aplica MeanEncoding Normalizado.	ingenieriaDeFeaturesRedes2
preparar set	completa los campos nulos por "no responde ", dejandolo valido al set	prepararSet

4. Auxiliares

- preprocessing.py contiene todos los preprocesamientos realizados en los distintos modelos y en el set de datos
- funcionesAuxiliares.py contiene distintas funciones auxiliares usadas en todo el tp
- predicciones es la carpeta donde se encuentran los archivos csv de las predicciones realizadas sobre el set de holdout
- requirements.txt contiene todos los requisitos para correr el tp.

5. Tabla de metricas

Realizamos una tabla que contiene a cada modelo implementado con su preprocesamientos y el valor obtenido de cada metrica(Auc roc, Accurray, Precision, Recall y F1 score)

Modelo	Nombre Preprocesamiento	Auc Roc	Accuracy	Precision	Recall	F1 score
Arbol de decision	IDFArboles1	0.89	0.85	0.74	0.64	0.66
Knn	IDFVariablesNormalizadasME	0.88	0.85	0.72	0.60	0.66
Naive bayes categorico	IDFCategoricalNB	0.84	0.80	0.58	0.57	0.57
Naive bayes gaussiano	IDFGaussianNB	0.82	0.79	0.77	0.21	0.33
boosting	IDFArboles1	0.91	0.87	0.77	0.60	0.67
Redes Neuronales	IDFRedes	0.89	0.84	0.72	0.52	0.62
Regresion Logistica	IDFVariablesNormalizadas	0.89	0.84	0.73	0.56	0.63
Svm	IDFSVM	0.88	0.83	0.70	0.55	0.62

6. Conclusion

Podemos concluir que el modelo que recomendamos es el boosting, ya que es el que mejor roc score obtuvo, ademas de que, si lo comparamos con las redes neuronales, el cual es otro modelo que suele obtener buenos resultados en las predicciones, el boosting tiene un mayor recall. Esta metrica es importante para nuestro problema ya que nos interesa que predecir la mayor cantidad de positivos correctamente, y no nos importa que hayan falsos positivos.

Con respecto al baseline de la primera parte del trabajo practico, en este caso el modelo es capaz de aprender y mejorarse a partir de sus errores, a diferencia del baseline que es deterministico. Por otro lado, el baseline no corria riesgo de overfittear ya que es un modelo simple que no corre riesgo de memorizar los datos.

Si necesitaramos obtener la menor cantidad de falsos positivos, deberiamos elegir al modelo con mejor precision. En este caso eligiriamos naive bayes o boosting, ya que ambos obtuvieron las mejores precisiones.

Por ultimo, si quisieramos obtener una lista de todos los que potencialmente son de valor adquisitivo, sin preocuparnos demasiado si hay falsos positivos, observariamos la metrica recall. En este caso elegiriamos el arbol de decision, ya que es el que obtuvo el mejor recall.