.UBAfiuba
FACULTAD DE INGENIERÍA

Trabajo Práctico: parte 2

Organización de datos
Primer cuatrimestre 2021

Grupo: 10

Apellido y Nombre	Padrón	Email
Buzzone, Mauricio	103783	mbuzzone@fi.uba.ar
Valdez, Santiago	103785	svaldez@fi.uba.ar

Índice

1. Introducción

En el presente trabajo se busca ampliar el análisis realizado en la parte 1 del trabajo practico utilizando los modelo estudiados en la materia con el fin de mejorar la predicción.

2. Objetivos

El trabajo practico tiene se plantea los siguiente objetivos.

- Entender el funcionamiento de los diferentes modelos.
- Saber medir los modelo con las diferentes métricas conocidas.
- Utilizar diferentes preprocesamientos para los datos.

3. Archivos entregados

A continuación se detalla una lista de archivos presentados

3.1. Notebooks

- 1-ArbolesDeDecision.ipynb
- 2-KNN.ipynb
- 3-SVM.ipynb
- 4-Bagging.ipynb
- 5-RandomForest.ipynb
- 6-RedesNeuronales.ipynb

3.2. Misceláneas

- preprocesamiento.py - Contienen las funciones de preprocesamiento utilizadas, ademas de algunas funciones auxiliares
- requirements.txt - Contiene la lista de requerimientos para la correcta funcionalidad de los notebooks
- Carpeta PrediccionesHoldout - En ella se encuentran los CSV de las predicciones hechas por el mejor modelo de los de su mismo tipo
- Exploracion - Carpeta utilizada para la exploracion del data set asi como la utilizacion de modelos no supervisados, no forma parte de la entrega formal
- Enunciado.md - Enunciado del Trabajo Practico Parte 2

4. Tablas

4.1. Tabla 1: Preprocesamientos

Nombre	Explicacion	Nombre de la función
Feature Engineering	Aplica el feature engineering utilizado en el TP1	feature_engineering
Preprocesamiento Basico	Aplica el feature engineering del TP1 y divide el dataset en (X, y)	preprocesar_data_frame
Dummificar	Agarra las variables categoricas y las dummifica	prepros_dummies
Numericas	Aplica la tanh para transformar los valores numericos al intervalo [-1, 1]	preprocesar_variables_numericas
Min Max Scaler	Agarra las variables categoricas, las dummifica y luego escala los features en base a su valor minimo y maximo	preprocesar_df_min_max_scaler
PCA	Buscar la proyeccion que mantiene la mayor varianza en la dimension pedida	preprocesar_df_pca
Standard Scaler	Escala los fetures restandoles su media y dividiendo su varianza	preprocesar_standar_scaler
Normalize Scaler	Agarra las variables categoricas, las dummifica y luego escala las instancias de forma tal de que tengan norma uno	preprocesar_normalize_scaler

4.2. Tabla 2: Métricas de los modelos

Presentamos la tabla ordenada por orden de realización de modelo. Para la toma de valores de Precision, Recall y F1 Score se tomaron los valores para los unos y los ceros.

Todos los modelos se pasaron primero por Preprocesamiento Basico, debido a esto se decidio no incluirlo en la tabla, además de que en algunos preprocesamientos se realizo una previa selección de features, tampoco figura esta selección en la tabla.

Nombre	Preprocesamiento	AUC-ROC	Accuracy	Precision	Recall	F1 Score
1 - Arboles de Decisión	Min Max Scaler y PCA	0.87	0.83	0: 0.88 1: 0.64	0: 0.89 1: 0.62	0: 0.89 1: 0.63
2 - KNN	Min Max Scaler y PCA	0.88	0.84	0: 0.86 1: 0.73	0: 0.94 1: 0.52	0: 0.90 1: 0.61
3 - SVM	Dummificar y Numericas	0.88	0.82	0: 0.88 1: 0.63	0: 0.89 1: 0.61	0: 0.88 1: 0.62
4 - Bagging	Normalize Scaler y Selección Var	0.91	0.85	0: 0.88 1: 0.75	0: 0.93 1: 0.58	0: 0.90 1: 0.64
5 - Random Forest	Dummificar y Min Max Scaler	0.91	0.85	0: 0.88 1: 0.73	0: 0.93 1: 0.61	0: 0.91 1: 0.66
6 - Redes Neuronales	Dummificar y Normalize Scaler	0.89	0.84	0: 0.88 1: 0.70	0: 0.92 1: 0.58	0: 0.90 1: 0.63

5. Conclusión

Como conclusiones del trabajo práctico, destacamos que casi todos los modelos implementados tuvieron un buen rendimiento, estando este en el rango de 0.85 a 0.9 según la métrica AUCROC. Debido a esto todos son candidatos validos a la hora de elegir el modelo que se utilizará para predecir si una persona tiene altos ingresos o bajos ingresos a partir de los mismos.

Habiendo visto los resultados de los diferentes modelos presentados y considerando que debemos elegir solo un modelo, llegamos a la conclusión que recomendamos utilizar el modelo de Random Forest, esto se debe a que presenta el mejor resultado de AUC-ROC, junto a un muy buen rendimiento en Accuracy, Precision, Recall, y F1 Score en comparación a los demás.

Que Random Forest sea el modelo recomendado resulta esperable, debido a que es un ensamble. Si comparamos este modelo con el baseline implementado anteriormente, podemos observar que Random Forest tiene un mejor resultado para Accuracy, la métrica presentada en el baseline. (0.85 vs 0.81). Sin embargo, cabe recordar que el accuracy del baseline fue obtenido a partir de entrenar y evaluar sobre todo el set de datos.

Para obtener los modelos recién mencionados se utilizo grid search cross validation, con esto obtuvimos las mejores combinaciones de hiperparámetros para cada uno de ellos (probando en diferentes casos distintos preprocesamientos). Hicimos una división del set de datos en train y test (test del tamaño 0.25, aproximadamente 8100 casos de evaluación). Con el set de datos, entrenamos el modelo y con el set de test evaluamos la predicciones realizadas y las métricas para el modelo. Cabe destacar que la partición de los datos es pseudo-aleatoria, manejada por un parámetro de random state (igualado a $19 \cdot 103785$ a lo largo del trabajo).

Hay que recordar que no todos los modelos sirven para todas las situaciones, existen casos en que es mas importante no tener falsos positivos que un alto accuracy. O puede ser mas importante captar a todos los potenciales targets sin preocuparse por clasificar incorrectamente a los demás. Debido a esto también decidimos analizar cuales son los mejores modelo para estos casos.

- En el caso de no querer tener falso positivos lo que se busca es una gran precisión de unos, que si el modelo predice un uno tenga pocas chances de estar equivocado. Existen varios posibles candidatos: KNN, Random Forest y Bagging con uno 0.73/0.72 de precisión.
- En el caso de querer captar a la mayor cantidad de unos sin tener en cuenta los ceros mal clasificados, debemos priorizar el recall. Para esta tarea se podría utilizar: Arboles de decisión, SVM y Random Forest con un 0.62/0.61 de recall. Lo recomendado seria utilizar Arboles de decisión ya que sacrificando un poco de precisión en las demás métricas se obtiene una representación gráfica que puede ser muy útil.

Algo importante que también se puede destacar es la flexibilidad de las redes neuronales, ya que ellas nos devuelven como output un scoring de "que dan 1 o 0 es la instancia"de modo que podemos fijar el threshold con el cual clasificamos en alto y bajo poder adquisitivo. Por ejemplo si fijamos que si el output es mayor 0.5 se clasifica como 1 y en caso contrario se clasifica como 0. Este numero mágico "0.5"lo podemos subir y bajar en función de si queremos priorizar recall o precisión. En el caso de ser menor a 0.5 se busca tener un buen recall, clasificamos como 1 también a la que las red los clasifica como confusos. Y si el threshold es mayor a 0.5 priorizamos la precisión ya que solo clasificamos como 1 a los que la red esta segura que deben ser 1.