

**Clasificación del desempeño académico utilizando modelos de Machine Learning y
Deep Learning**

Rebeca Cuan Corral

Universidad Autónoma del Estado de Morelos

Proyecto final del diplomado en ciencia de datos con Python

Dr. Mauricio Toledo

03 de agosto de 2024

Clasificación del desempeño académico utilizando modelos de Machine Learning y Deep Learning

Identificación del problema:

¿Cómo predecir el desempeño académico de los estudiantes de preparatoria en base con los datos demográficos, los hábitos de estudio, la participación de los padres y las actividades extracurriculares?

Objetivo:

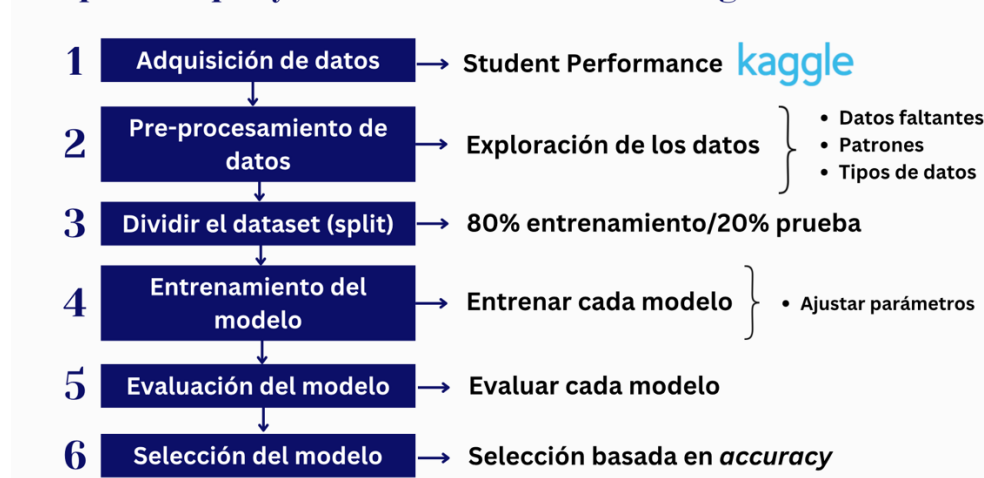
Seleccionar el modelo de *Machine Learning* más preciso (*accuracy*) para clasificar el desempeño estudiantil dentro de los siguientes algoritmos:

- Logistic Regression.
- Random Forest.
- Support Vector Machine (SVM).
- Decision Tree.
- Multilayer Perceptron(MLP).

Metodología:

A continuación, se describen las etapas del proyecto de Machine Learning.

Etapas del proyecto de Machine Learning



Los datos utilizados se tomaron de Kaggle (Rabie El Kharoua, 2024).

En la etapa 3 se dividen los datos en una proporción 80% (entrenamiento) / 20% (prueba).

Los datos se escalaron utilizando el módulo *StandardScaler*.

Resultados:

A continuación, se describirán los resultados de cada uno de los modelos de Machine Learning.

Regresión Logística

Se entrenaron los siguientes escenarios:

1. Regresión Logística con los parametros por defecto
2. Regresión Logística con *Polynomial Features*
3. Regresión Logística con los mejores parametros resultantes de la busqueda utilizando *GridsearchCV*
4. Regresión Logística con los mejores parametros resultantes de la busqueda utilizando *GridsearchCV*, después de aplicar Feature Selection usando *SelectKBest*

Los resultados son los siguientes:

#Modelo	Parametros	Accuracy
1	Default	71.82%
2	PolynomialFeatures(2,include_bias=False) LogisticRegression(max_iter=100000, random_state=42, solver='saga')	70.56%
3	param_grid = {'C': [0.1, 1, 10], 'penalty': ['l1', 'l2'], 'max_iter': [3000], 'solver': ['saga']} LogisticRegression(max_iter=3000, random_state=42, solver='saga', C=1, penalty='l1')	72.23%
4	SelectKBest(chi2, k=14) LogisticRegression(max_iter=3000, random_state=42, solver='saga', C=1, penalty='l1')	73.7%

A continuación, se presentan los coeficientes para el escenario de Regresión Logística número 4. En este escenario se hizo una selección de atributos utilizando el módulo *SelectKBest*.

```
1 | Ver los coeficientes
2 odds = np.exp(clf_lr_selected.coef_[0])
3 pd.DataFrame(odds,
4               selector.get_feature_names_out(),
5               columns=['coef'])
```

	coef
StudyTimeWeekly	1.080722
Absences	0.874932
Ethnicity_2	1.538189
Ethnicity_3	0.851647
ParentalEducation_4	1.000000
Tutoring_1	2.510916
ParentalSupport_1	1.000000
ParentalSupport_2	0.990683
ParentalSupport_3	3.211383
ParentalSupport_4	5.649132
Extracurricular_1	1.589320
Sports_1	1.643693
Music_1	1.221400
Volunteering_1	1.000000

Respecto a los coeficientes de Regresión Logística que aparecen en la imagen anterior, podemos decir lo siguiente:

- La variable faltas (absences) reducen la probabilidad de pertenencia a la calificación “A” (coeficiente menor que 1).
- El ParentSupport3 y ParentSupport4 (Alto y Muy Alto, respectivamente) aumentan la probabilidad de obtener una clase “A” (coeficiente mayor que 1).

Los siguientes modelos se entrenarán con el dataset resultante de utilizar SelectKBest=14.

Decision Trees

Se entrenaron los siguientes escenarios:

1. Decision Trees con los parámetros por defecto.
2. Decision Trees con los mejores parametros resultantes de la búsqueda utilizando *GridSearchCV*.

Los resultados son los siguientes:

#Modelo	Parámetros	Accuracy
1	Default	62.21%
2	DecisionTreeClassifier(max_depth=9, criterion='gini', ccp_alpha=0.001, max_features=None)	67.43%

Random Forest

Se entrenaron los siguientes escenarios:

1. Random Forest con los parámetros por defecto.
2. Random Forest con los mejores parametros resultantes de la búsqueda utilizando *GridSearchCV*.

Los resultados son los siguientes:

# Modelo	Parametros	Accuracy
1	Default	69.7%
2	RandomForestClassifier(random_state=42, max_depth=13)	70.6%

Support Vector Machine (SVM)

Se entrenaron los siguientes escenarios:

1. SVM con los parámetros por defecto.
2. SVM con los mejores parámetros resultantes de la búsqueda utilizando *GridSearchCV*.

Los resultados son los siguientes:

# Modelo	Parámetros	Accuracy
1	Default	76.61%
2	SVC(kernel='rbf',C=10, gamma=0.01,random_state=42)	74.95%

Multi-Layer Perceptron (MLP)

Se entrenaron los siguientes escenarios:

1. MLP con 1 capa oculta.

Los resultados son los siguientes:

# Modelo	Parametros	Accuracy
1	<pre>model = Sequential() model.add(Dense(16, activation='relu', input_dim=14)) # model.add(Dense(32, activation='relu')) model.add(Dense(5, activation='softmax'))</pre>	72.23%

Resumen general

Modelo	Accuracy
Regresion Logistica	73.7%
Decision Trees	67.43%
Random Forest	70.1%
SVM	74.95% - 76.61%
MLP	72.23%

Conclusiones

- El modelo de clasificación más preciso es el de SVM con un accuracy de 74.95%.
- El modelo más interpretable es el Logistic Regression con un accuracy de 73.7%, tres puntos porcentuales menos que el mejor modelo encontrado (SVM).
- La decisión de qué modelo se va a elegir va a depender si el **usuario final** prefiere **interpretabilidad** o mejor resultado.

Referencias

Rabie El Kharoua. (2024). 📖 Students Performance Dataset 📖 [Data set]. Kaggle.
<https://doi.org/10.34740/KAGGLE/DS/5195702>