ejemplo de casificacion con tensorflow

In [1]:

```
import pandas as pd
import tensorflow as tf
```

In [2]:

```
pwd
```

Out[2]:

'C:\\Users\\SARA'

In [3]:

```
cd Downloads
```

C:\Users\SARA\Downloads

In [9]:

```
ingresos = pd.read_csv('original.csv')
```

In [10]:

```
ingresos
```

Out[10]:

	age	workclass	fnlwgt	education	educational-num	marital-status	occupation	relationship	race
0	25	Private	226802	11th	7	Never-married	Machine-op-inspct	Own-child	Black
1	38	Private	89814	HS-grad	9	Married-civ-spouse	Farming-fishing	Husband	White
2	28	Local-gov	336951	Assoc-acdm	12	Married-civ-spouse	Protective-serv	Husband	White
3	44	Private	160323	Some-college	10	Married-civ-spouse	Machine-op-inspct	Husband	Black
4	18	?	103497	Some-college	10	Never-married	?	Own-child	White
...	...	...	...	...	...	...	...	...	...
48837	27	Private	257302	Assoc-acdm	12	Married-civ-spouse	Tech-support	Wife	White
48838	40	Private	154374	HS-grad	9	Married-civ-spouse	Machine-op-inspct	Husband	White
48839	58	Private	151910	HS-grad	9	Widowed	Adm-clerical	Unmarried	White
48840	22	Private	201490	HS-grad	9	Never-	Adm-clerical	Own-child	White

48840	22	Private	201490	HS-grad	9	Married-civ-spouse	Adm-clerical	Own-child	White	
	age	**workclass**	**fnlwgt**	**education**	**educational-num**	**marital-status**	**occupation**	**relationship**	**race**	
48841	52	Self-emp-inc	287927	HS-grad	9	Married-civ-spouse	Exec-managerial	Wife	White	

48842 rows × 15 columns

In [14]:

```
ingresos['income'].unique()
```

Out[14]:

```
array(['<=50K', '>50K'], dtype=object)
```

In [15]:

```
def cambio_valor(valor):
    if valor == '<=50k':
        return 0
    else:
        return 1
```

In [16]:

```
ingresos['income'] = ingresos['income'].apply(cambio_valor)
```

In [17]:

```
ingresos.head()
```

Out[17]:

	age	workclass	fnlwgt	education	educational-num	marital-status	occupation	relationship	race	ge
0	25	Private	226802	11th	7	Never-married	Machine-op-inspct	Own-child	Black	
1	38	Private	89814	HS-grad	9	Married-civ-spouse	Farming-fishing	Husband	White	
2	28	Local-gov	336951	Assoc-acdm	12	Married-civ-spouse	Protective-serv	Husband	White	
3	44	Private	160323	Some-college	10	Married-civ-spouse	Machine-op-inspct	Husband	Black	
4	18	?	103497	Some-college	10	Never-married	?	Own-child	White	Fe

In [18]:

```
from sklearn.model_selection import train_test_split
```

In [19]:

```
datos_x = ingresos.drop('income', axis=1)
```

In [20]:

```
datos_x.head()
```

Out[20]:

	age	workclass	fnlwgt	education	educational-num	marital-status	occupation	relationship	race	ge
0	25	Private	226802	11th	7	Never-married	Machine-op-inspct	Own-child	Black	
1	38	Private	89814	HS-grad	9	Married-civ-spouse	Farming-fishing	Husband	White	
2	28	Local-gov	336951	Assoc-acdm	12	Married-civ-spouse	Protective-serv	Husband	White	
3	44	Private	160323	Some-college	10	Married-civ-spouse	Machine-op-inspct	Husband	Black	
4	18	?	103497	Some-college	10	Never-married	?	Own-child	White	Fe

In [21]:

```
datos_y = ingresos['income']
```

In [22]:

```
datos_y
```

Out[22]:

```
0        1
1        1
2        1
3        1
4        1
      ..
48837    1
48838    1
48839    1
48840    1
48841    1
Name: income, Length: 48842, dtype: int64
```

In [23]:

```
x_train, x_test, y_train, y_test = train_test_split(datos_x, datos_y, test_size=0.3)
```

In [24]:

```
x_train
```

Out[24]:

	age	workclass	fnlwgt	education	educational-num	marital-status	occupation	relationship	ra
48396	29	Private	119359	HS-grad	9	Married-civ-spouse	Prof-specialty	Wife	Asi P Islan

	age	workclass	fnlwgt	education	educational-num	marital-status	occupation	relationship	ra
29891	40	Self-emp-not-inc	179533	HS-grad	9	civ-spouse	Transport-moving	Husband	Wh
9684	27	Private	416946	HS-grad	9	Married-civ-spouse	Craft-repair	Husband	Bla
8978	22	Private	203518	Bachelors	13	Never-married	Sales	Not-in-family	Wh
5572	34	Private	205152	HS-grad	9	Married-civ-spouse	Handlers-cleaners	Husband	Wh
...	...	...	...	...	...	...	...	...	
27451	44	Private	300528	11th	7	Married-civ-spouse	Adm-clerical	Husband	Wh
3207	22	Private	272591	10th	6	Never-married	Machine-op-inspct	Not-in-family	Wh
38397	42	Private	194710	Some-college	10	Never-married	Exec-managerial	Not-in-family	Wh
45465	29	Local-gov	205262	Some-college	10	Never-married	Adm-clerical	Not-in-family	Otl
23143	51	?	69328	Assoc-voc	11	Married-civ-spouse	?	Husband	Wh

34189 rows × 14 columns

In [27]:

```
x_test.head()
```

Out[27]:

	age	workclass	fnlwgt	education	educational-num	marital-status	occupation	relationship	ra
5814	48	Private	323798	Assoc-acdm	12	Married-civ-spouse	Prof-specialty	Husband	Whi
10714	55	Private	199212	Some-college	10	Married-civ-spouse	Adm-clerical	Wife	Whi
28718	21	Private	166517	HS-grad	9	Never-married	Craft-repair	Own-child	Whi
6625	39	Private	140169	10th	6	Separated	Other-service	Unmarried	Whi
10598	39	Self-emp-inc	128715	HS-grad	9	Married-civ-spouse	Exec-managerial	Husband	Whi

In [28]:

```
ingresos.columns
```

Out[28]:

```
Index(['age', 'workclass', 'fnlwgt', 'education', 'educational-num',
       'marital-status', 'occupation', 'relationship', 'race', 'gender',
       'capital-gain', 'capital-loss', 'hours-per-week', 'native-country',
       'income'],
      dtype='object')
```

In [29]:

```
gender = tf.feature_column.categorical_column_with_vocabulary_list('gender',['Female, Male'])
```

In [31]:

```
occupation = tf.feature_column.categorical_column_with_hash_bucket('occupation', hash_bucket_size=1000
)
```

In [32]:

```
occupation = tf.feature_column.categorical_column_with_hash_bucket('occupation', hash_bucket_size=1000
)
marital_status = tf.feature_column.categorical_column_with_hash_bucket('marital-status', hash_bucket_size
=1000)
relationship = tf.feature_column.categorical_column_with_hash_bucket('relationship', hash_bucket_size=100
0)
education= tf.feature_column.categorical_column_with_hash_bucket('education', hash_bucket_size=1000)
native_country = tf.feature_column.categorical_column_with_hash_bucket('native-country', hash_bucket_siz
e=1000)
workclass = tf.feature_column.categorical_column_with_hash_bucket('workclass', hash_bucket_size=1000)
```

In [33]:

```
age = tf.feature_column.numeric_column('age')
```

In [34]:

```
age = tf.feature_column.numeric_column('age')
fnlwgt = tf.feature_column.numeric_column('fnlwgt')
educational_num = tf.feature_column.numeric_column('educational_num')
capital_gain = tf.feature_column.numeric_column('capital_gain')
capital_loss = tf.feature_column.numeric_column('capital_loss')
hours_per_week = tf.feature_column.numeric_column('hours_per_week')
```

In [37]:

```
columnas_categorias = [gender, occupation, marital_status, relationship, education, native_country, workcla
ss, age, fnlwgt, educational_num, capital_gain, capital_loss, hours_per_week]
```

In [41]:

```
funcion_entrada = tf.estimator.inputs.pandas_input_fn(x=x_train, y=y_train, batch_size=100, num_epochs=N
one, shuffle=True)
```

In [42]:

```
modelo = tf.estimator.LinearClassifier(feature_columns=columnas_categorias)
```

```
INFO:tensorflow:Using default config.
WARNING:tensorflow:Using temporary folder as model directory: C:\Users\SARA\AppData\Local\Temp\tmpukt6
zdb6
INFO:tensorflow:Using config: {'_model_dir': 'C:\\Users\\SARA\\AppData\\Local\\Temp\\tmpukt6zdb6', '_tf_rando
m_seed': None, '_save_summary_steps': 100, '_save_checkpoints_steps': None, '_save_checkpoints_secs': 60
0, '_session_config': allow_soft_placement: true
graph_options {
  rewrite_options {
    meta_optimizer_iterations: ONE
  }
}
, '_keep_checkpoint_max': 5, '_keep_checkpoint_every_n_hours': 10000, '_log_step_count_steps': 100, '_train
_distribute': None, '_device_fn': None, '_protocol': None, '_eval_distribute': None, '_experimental_distribute': No
ne, '_experimental_max_worker_delay_secs': None, '_session_creation_timeout_secs': 7200, '_service': None,
'_cluster_spec': <tensorflow.python.training.server_lib.ClusterSpec object at 0x00000245BDD53088>, '_task_ty
```

pe': 'worker', '_task_id': 0, '_global_id_in_cluster': 0, '_master': '', '_evaluation_master': '', '_is_chief': True, '_num_ps_replicas': 0, '_num_worker_replicas': 1}

In [43]:

```
modelo.train(input_fn=funcion_entrada, steps=8000)
```

```
WARNING:tensorflow:From C:\Users\SARA\anaconda3\envs\pruebasTensorflow\lib\site-packages\tensorflow_core\python\training\training_util.py:236: Variable.initialized_value (from tensorflow.python.ops.variables) is deprecated and will be removed in a future version.
Instructions for updating:
Use Variable.read_value. Variables in 2.X are initialized automatically both in eager and graph (inside tf.defun) contexts.
WARNING:tensorflow:From C:\Users\SARA\anaconda3\envs\pruebasTensorflow\lib\site-packages\tensorflow_estimator\python\estimator\inputs\queues\feeding_queue_runner.py:62: QueueRunner.__init__ (from tensorflow.python.training.queue_runner_impl) is deprecated and will be removed in a future version.
Instructions for updating:
To construct input pipelines, use the `tf.data` module.
WARNING:tensorflow:From C:\Users\SARA\anaconda3\envs\pruebasTensorflow\lib\site-packages\tensorflow_estimator\python\estimator\inputs\queues\feeding_functions.py:500: add_queue_runner (from tensorflow.python.training.queue_runner_impl) is deprecated and will be removed in a future version.
Instructions for updating:
To construct input pipelines, use the `tf.data` module.
INFO:tensorflow:Calling model_fn.
WARNING:tensorflow:From C:\Users\SARA\anaconda3\envs\pruebasTensorflow\lib\site-packages\tensorflow_core\python\feature_column\feature_column_v2.py:305: Layer.add_variable (from tensorflow.python.keras.engine.base_layer) is deprecated and will be removed in a future version.
Instructions for updating:
Please use `layer.add_weight` method instead.
WARNING:tensorflow:From C:\Users\SARA\anaconda3\envs\pruebasTensorflow\lib\site-packages\tensorflow_core\python\ops\resource_variable_ops.py:1630: calling BaseResourceVariable.__init__ (from tensorflow.python.ops.resource_variable_ops) with constraint is deprecated and will be removed in a future version.
Instructions for updating:
If using Keras pass *_constraint arguments to layers.

---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
<ipython-input-43-fe147f8aff12> in <module>
----> 1 modelo.train(input_fn=funcion_entrada, steps=8000)

~\anaconda3\envs\pruebasTensorflow\lib\site-packages\tensorflow_estimator\python\estimator\estimator.py in train(self, input_fn, hooks, steps, max_steps, saving_listeners)
    368
    369       saving_listeners = _check_listeners_type(saving_listeners)
--> 370       loss = self._train_model(input_fn, hooks, saving_listeners)
    371       logging.info('Loss for final step: %s.', loss)
    372       return self

~\anaconda3\envs\pruebasTensorflow\lib\site-packages\tensorflow_estimator\python\estimator\estimator.py in _train_model(self, input_fn, hooks, saving_listeners)
   1159       return self._train_model_distributed(input_fn, hooks, saving_listeners)
   1160     else:
-> 1161       return self._train_model_default(input_fn, hooks, saving_listeners)
   1162
   1163   def _train_model_default(self, input_fn, hooks, saving_listeners):

~\anaconda3\envs\pruebasTensorflow\lib\site-packages\tensorflow_estimator\python\estimator\estimator.py in _train_model_default(self, input_fn, hooks, saving_listeners)
   1189       worker_hooks.extend(input_hooks)
   1190       estimator_spec = self._call_model_fn(
-> 1191           features, labels, ModeKeys.TRAIN, self.config)
```

```
1192       global_step_tensor = training_util.get_global_step(g)
1193       return self._train_with_estimator_spec(estimator_spec, worker_hooks,

~\anaconda3\envs\pruebasTensorflow\lib\site-packages\tensorflow_estimator\python\estimator\estimator.py in _call_model_fn(self, features, labels, mode, config)
1147
1148     logging.info('Calling model_fn.')
-> 1149     model_fn_results = self._model_fn(features=features, **kwargs)
1150     logging.info('Done calling model_fn.')
1151

~\anaconda3\envs\pruebasTensorflow\lib\site-packages\tensorflow_estimator\python\estimator\canned\linear.py in _model_fn(features, labels, mode, config)
989           partitioner=partitioner,
990           config=config,
--> 991           sparse_combiner=sparse_combiner)
992
993     super(LinearClassifier, self).__init__(

~\anaconda3\envs\pruebasTensorflow\lib\site-packages\tensorflow_estimator\python\estimator\canned\linear.py in _linear_model_fn(features, labels, mode, head, feature_columns, optimizer, partitioner, config, sparse_combiner)
742           sparse_combiner=sparse_combiner,
743           )
--> 744       logits = logit_fn(features=features)
745
746       optimizer = optimizers.get_optimizer_instance(

~\anaconda3\envs\pruebasTensorflow\lib\site-packages\tensorflow_estimator\python\estimator\canned\linear.py in linear_logit_fn(features)
421           sparse_combiner=sparse_combiner,
422           name='linear_model')
--> 423       logits = linear_model(features)
424       bias = linear_model.bias
425

~\anaconda3\envs\pruebasTensorflow\lib\site-packages\tensorflow_core\python\keras\engine\base_layer.py in __call__(self, inputs, *args, **kwargs)
852                     outputs = base_layer_utils.mark_as_return(outputs, acd)
853                 else:
--> 854                   outputs = call_fn(cast_inputs, *args, **kwargs)
855
856           except errors.OperatorNotAllowedInGraphError as e:

~\anaconda3\envs\pruebasTensorflow\lib\site-packages\tensorflow_core\python\autograph\impl\api.py in wrapper(*args, **kwargs)
235       except Exception as e:  # pylint:disable=broad-except
236         if hasattr(e, 'ag_error_metadata'):
--> 237           raise e.ag_error_metadata.to_exception(e)
238         else:
239           raise

ValueError: in converted code:
    relative to C:\Users\SARA\anaconda3\envs\pruebasTensorflow\lib\site-packages\tensorflow_core\python\feature_column:

    feature_column_v2.py:696 call
        return self.layer(features)
    feature_column_v2.py:530 call
        weight_var=weight_var)
```

```
feature_column_v2.py:2401 _create_weighted_sum
    weight_var=weight_var)
feature_column_v2.py:2407 _create_dense_column_weighted_sum
    tensor = column.get_dense_tensor(transformation_cache, state_manager)
feature_column_v2.py:2835 get_dense_tensor
    return transformation_cache.get(self, state_manager)
feature_column_v2.py:2598 get
    transformed = column.transform_feature(self, state_manager)
feature_column_v2.py:2807 transform_feature
    input_tensor = transformation_cache.get(self.key, state_manager)
feature_column_v2.py:2590 get
    raise ValueError('Feature {} is not in features dictionary.'.format(key))

ValueError: Feature capital_gain is not in features dictionary.
```

In [47]:

```
funcion_prediccion = tf.estimator.inputs.pandas_input_fn(x=x_test, batch_size=len(x_test), shuffle=False)
```

In [48]:

```
generador_predicciones=modelo.predict(input_fn=funcion_prediccion)
```

In [50]:

```
predicciones = list(generador_predicciones)
```

In [51]:

```
predicciones
```

Out[51]:

```
[]
```

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]: