

In [1]:

```
import tensorflow as tf
```

In [2]:

```
from tensorflow import estimator
from sklearn.datasets import load_wine
```

In [3]:

```
vino = load_wine()
```

In [4]:

```
print(vino['DESCR'])
```

.. _wine_dataset:

Wine recognition dataset

****Data Set Characteristics:****

- :Number of Instances: 178 (50 in each of three classes)
- :Number of Attributes: 13 numeric, predictive attributes and the class
- :Attribute Information:
 - Alcohol
 - Malic acid
 - Ash
 - Alcalinity of ash
 - Magnesium
 - Total phenols
 - Flavanoids
 - Nonflavanoid phenols
 - Proanthocyanins
 - Color intensity
 - Hue
 - OD280/OD315 of diluted wines
 - Proline
- class:
 - class_0
 - class_1
 - class_2

:Summary Statistics:

	Min	Max	Mean	SD
Alcohol:	11.0	14.8	13.0	0.8
Malic Acid:	0.74	5.80	2.34	1.12
Ash:	1.36	3.23	2.36	0.27
Alcalinity of Ash:	10.6	30.0	19.5	3.3
Magnesium:	70.0	162.0	99.7	14.3
Total Phenols:	0.98	3.88	2.29	0.63

Flavanoids:	0.34	5.08	2.03	1.00
Nonflavanoid Phenols:	0.13	0.66	0.36	0.12
Proanthocyanins:	0.41	3.58	1.59	0.57
Colour Intensity:	1.3	13.0	5.1	2.3
Hue:	0.48	1.71	0.96	0.23
OD280/OD315 of diluted wines:	1.27	4.00	2.61	0.71
Proline:	278	1680	746	315

=====

```
:Missing Attribute Values: None
:Class Distribution: class_0 (59), class_1 (71), class_2 (48)
:Creator: R.A. Fisher
:Donor: Michael Marshall (MARSHALL%PLU@io.arc.nasa.gov)
:Date: July, 1988
```

This is a copy of UCI ML Wine recognition datasets.
<https://archive.ics.uci.edu/ml/machine-learning-databases/wine/wine.data>

The data is the results of a chemical analysis of wines grown in the same region in Italy by three different cultivators. There are thirteen different measurements taken for different constituents found in the three types of wine.

Original Owners:

Forina, M. et al, PARVUS -
 An Extendible Package for Data Exploration, Classification and Correlation.
 Institute of Pharmaceutical and Food Analysis and Technologies,
 Via Brigata Salerno, 16147 Genoa, Italy.

Citation:
 Lichman, M. (2013). UCI Machine Learning Repository
 [<https://archive.ics.uci.edu/ml>]. Irvine, CA: University of California,
 School of Information and Computer Science.

.. topic:: References

(1) S. Aeberhard, D. Coomans and O. de Vel,
 Comparison of Classifiers in High Dimensional Settings,
 Tech. Rep. no. 92-02, (1992), Dept. of Computer Science and Dept. of
 Mathematics and Statistics, James Cook University of North Queensland.
 (Also submitted to Technometrics).

The data was used with many others for comparing various
 classifiers. The classes are separable, though only RDA
 has achieved 100% correct classification.
 (RDA : 100%, QDA 99.4%, LDA 98.9%, 1NN 96.1% (z-transformed data))
 (All results using the leave-one-out technique)

(2) S. Aeberhard, D. Coomans and O. de Vel,
 "THE CLASSIFICATION PERFORMANCE OF RDA"
 Tech. Rep. no. 92-01, (1992), Dept. of Computer Science and Dept. of
 Mathematics and Statistics, James Cook University of North Queensland.
 (Also submitted to Journal of Chemometrics).

In [5]:

```
caracteristicas = vino['data']
objetivo = vino['target']
```



In [6]:

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

In [7]:

```
X_train, X_test, y_train, y_test = train_test_split(caracteristicas, objetivo, test_size=0.3)
```

In [8]:

```
from sklearn.preprocessing import MinMaxScaler
```

In [9]:

```
normalizador = MinMaxScaler()
```

In [10]:

```
x_train_normalizado = normalizador.fit_transform(X_train)
x_test_normalizado = normalizador.transform(X_test)
```

In [11]:

```
X_train
```

Out[11]:

```
array([[1.383e+01, 1.570e+00, 2.620e+00, ..., 1.130e+00, 2.570e+00,
        1.130e+03],
       [1.406e+01, 2.150e+00, 2.610e+00, ..., 1.060e+00, 3.580e+00,
        1.295e+03],
       [1.311e+01, 1.010e+00, 1.700e+00, ..., 1.120e+00, 3.180e+00,
        5.020e+02],
       ...,
       [1.252e+01, 2.430e+00, 2.170e+00, ..., 9.000e-01, 2.780e+00,
        3.250e+02],
       [1.348e+01, 1.810e+00, 2.410e+00, ..., 1.040e+00, 3.470e+00,
        9.200e+02],
       [1.208e+01, 1.390e+00, 2.500e+00, ..., 9.300e-01, 3.190e+00,
        3.850e+02]])
```

In [12]:

```
x_train_normalizado
```

Out[12]:

```
array([[0.75268817, 0.16403162, 0.67741935, ..., 0.52845528, 0.48669202,
        0.59795322],
       [0.81451613, 0.27865613, 0.67204301, ..., 0.47154472, 0.87072243,
        0.71856725],
       [0.55913978, 0.05335968, 0.1827957 , ..., 0.5203252 , 0.71863118,
        0.13888889],
       ...,
       [0.40053763, 0.33399209, 0.43548387, ..., 0.34146341, 0.56653992,
        0.00950292],
       [0.65860215, 0.21146245, 0.56451613, ..., 0.45528455, 0.82889734,
        0.44444444],
       [0.28225806, 0.1284585 , 0.61290323, ..., 0.36585366, 0.72243346,
        0.05336257]])
```

In [13]:

```
x_train_normalizado.shape
```

Out[13]:

```
(124, 13)
```

In [14]:

```
columnas_caracteristicas = (tf.feature_column.numeric_column('x', shape=[13]))
```

In [16]:

```
modelo = estimator.DNNClassifier(hidden_units=[20,20,20], feature_columns = columnas_caracteristicas, n_classes=3, optimizer=tf.train.GradientDescentOptimizer(learning_rate=0.01))
```

INFO:tensorflow:Using default config.

WARNING:tensorflow:Using temporary folder as model directory: C:\Users\SARA\AppData\Local\Temp\tmp8spj_nvm

INFO:tensorflow:Using config: {'_model_dir': 'C:\\Users\\SARA\\AppData\\Local\\Temp\\tmp8spj_nvm', '_tf_random_seed': None, '_save_summary_steps': 100, '_save_checkpoints_steps': None, '_save_checkpoints_secs': 600, '_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': None, '_experimental_max_worker_delay_secs': None, '_session_creation_timeout_secs': 7200, '_service': None, '_cluster_spec': <tensorflow.python.training.server_lib.ClusterSpec object at 0x0000020B540BAD08>, '_task_type': 'worker', '_task_id': 0, '_global_id_in_cluster': 0, '_master': '', '_evaluation_master': '', '_is_chief': True, '_num_ps_replicas': 0, '_num_worker_replicas': 1}

In [17]:

```
funcion_entrada = estimator.inputs.numpy_input_fn(x={'x':x_train_normalizado}, y=y_train, shuffle=True, batch_size=10, num_epochs=10)
```

In [21]:

```
modelo.train(input_fn=funcion_entrada, steps=600)
```

INFO:tensorflow:Calling model_fn.

ValueError Traceback (most recent call last)

<ipython-input-21-854820b4a2ee> in <module>

----> 1 modelo.train(input_fn=funcion_entrada, steps=600)

~\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\estima
tor.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\estima
tor.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\canne
d\dnn.py in _model_fn(features, labels, mode, config)
809     input_layer_partitioner=input_layer_partitioner,
810     config=config,
--> 811     batch_norm=batch_norm)
812
813     super(DNNClassifier, self).__init__(

~\anaconda3\envs\pruebasTensorflow\lib\site-packages\tensorflow_estimator\python\estimator\canne
d\dnn.py in _dnn_model_fn(features, labels, mode, head, hidden_units, feature_columns, optimizer, acti
vation_fn, dropout, input_layer_partitioner, config, use_tpu, batch_norm)
461     input_layer_partitioner=input_layer_partitioner,
462     batch_norm=batch_norm)
--> 463     logits = logit_fn(features=features, mode=mode)
464
465     return _get_dnn_estimator_spec(use_tpu, head, features, labels, mode,

~\anaconda3\envs\pruebasTensorflow\lib\site-packages\tensorflow_estimator\python\estimator\canne
d\dnn.py in dnn_logit_fn(features, mode)
107     batch_norm,
108     name='dnn')
--> 109     return dnn_model(features, mode)
110
111     return dnn_logit_fn

~\anaconda3\envs\pruebasTensorflow\lib\site-packages\tensorflow_core\python\keras\engine\base_la
yer.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_estimator\python\estimator\canned\dnn.py:252 call *  
    net = self._input_layer(features)  
tensorflow_core\python\feature_column\feature_column.py:338 __call__  
    from_template=True)  
tensorflow_core\python\ops\template.py:393 __call__  
    return self._call_func(args, kwargs)  
tensorflow_core\python\ops\template.py:355 _call_func  
    result = self._func(*args, **kwargs)  
tensorflow_core\python\feature_column\feature_column.py:182 _internal_input_layer  
    feature_columns = _normalize_feature_columns(feature_columns)  
tensorflow_core\python\feature_column\feature_column.py:2300 _normalize_feature_columns  
    'Given (type {}): {}'.format(type(column), column))
```

ValueError: Items of feature_columns must be a _FeatureColumn. Given (type <class 'str'>): x,.

originally defined at:

```
File "C:\Users\SARA\anaconda3\envs\pruebasTensorflow\lib\site-packages\tensorflow_estimator\python\estimator\canned\dnn.py", line 108, in dnn_logit_fn  
    name='dnn')  
File "C:\Users\SARA\anaconda3\envs\pruebasTensorflow\lib\site-packages\tensorflow_estimator\python\estimator\canned\dnn.py", line 191, in __init__  
    create_scope_now=False)  
File "C:\Users\SARA\anaconda3\envs\pruebasTensorflow\lib\site-packages\tensorflow_core\python\feature_column\feature_column.py", line 328, in __init__  
    self._name, _internal_input_layer, create_scope_now_=create_scope_now)  
File "C:\Users\SARA\anaconda3\envs\pruebasTensorflow\lib\site-packages\tensorflow_core\python\ops\template.py", line 161, in make_template  
    **kwargs)
```

In []:

In [19]:

```
funcion_evaluacion = estimator.inputs.numpy_input_fn(x={'x':x_test_normalizado}, shuffle=False)
```

In [20]:

```
predicciones = list(modelo.predict(input_fn=funcion_evaluacion))
```

INFO:tensorflow:Could not find trained model in model_dir: C:\Users\SARA\AppData\Local\Temp\tmp8spj_nvm, running initialization to predict.
INFO:tensorflow:Calling model_fn.

ValueError Traceback (most recent call last)

<ipython-input-20-c3d73d8ed804> in <module>

----> 1 predicciones = list(modelo.predict(input_fn=funcion_evaluacion))

~\anaconda3\envs\pruebasTensorflow\lib\site-packages\tensorflow_estimator\python\estimator\estimator.py in predict(self, input_fn, predict_keys, hooks, checkpoint_path, yield_single_examples)

620 input_fn, ModeKeys.PREDICT)

621 estimator_spec = self._call_model_fn(
--> 622 features, None, ModeKeys.PREDICT, self.config)

623

624 # Call to warm_start has to be after model_fn is called.

```

~\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\dnn.py in _model_fn(features, labels, mode, config)
    809     input_layer_partitioner=input_layer_partitioner,
    810     config=config,
-> 811     batch_norm=batch_norm)
    812
    813     super(DNNClassifier, self).__init__(

```

```

~\anaconda3\envs\pruebasTensorflow\lib\site-packages\tensorflow_estimator\python\estimator\canned\dnn.py in _dnn_model_fn(features, labels, mode, head, hidden_units, feature_columns, optimizer, activation_fn, dropout, input_layer_partitioner, config, use_tpu, batch_norm)
    461     input_layer_partitioner=input_layer_partitioner,
    462     batch_norm=batch_norm)
-> 463     logits = logit_fn(features=features, mode=mode)
    464
    465     return _get_dnn_estimator_spec(use_tpu, head, features, labels, mode,

```

```

~\anaconda3\envs\pruebasTensorflow\lib\site-packages\tensorflow_estimator\python\estimator\canned\dnn.py in dnn_logit_fn(features, mode)
    107     batch_norm,
    108     name='dnn')
-> 109     return dnn_model(features, mode)
    110
    111     return dnn_logit_fn

```

```

~\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_estimator\python\estimator\canned\dnn.py:252 call *
    net = self._input_layer(features)
tensorflow_core\python\feature_column\feature_column.py:338 __call__
    from_template=True)
tensorflow_core\python\ops\template.py:393 __call__
    return self._call_func(args, kwargs)
tensorflow_core\python\ops\template.py:355 _call_func

```

```
result = self._func(*args, **kwargs)
tensorflow_core\python\feature_column\feature_column.py:182 _internal_input_layer
feature_columns = _normalize_feature_columns(feature_columns)
tensorflow_core\python\feature_column\feature_column.py:2300 _normalize_feature_columns
'Given (type {}): {}'.format(type(column), column))
```

ValueError: Items of feature_columns must be a _FeatureColumn. Given (type <class 'str'>): x,.

originally defined at:

```
File "C:\Users\SARA\anaconda3\envs\pruebasTensorflow\lib\site-packages\tensorflow_estimator\python\estimator\canned\dnn.py", line 108, in dnn_logit_fn
    name='dnn')
File "C:\Users\SARA\anaconda3\envs\pruebasTensorflow\lib\site-packages\tensorflow_estimator\python\estimator\canned\dnn.py", line 191, in __init__
    create_scope_now=False)
File "C:\Users\SARA\anaconda3\envs\pruebasTensorflow\lib\site-packages\tensorflow_core\python\feature_column\feature_column.py", line 328, in __init__
    self._name, _internal_input_layer, create_scope_now_=create_scope_now)
File "C:\Users\SARA\anaconda3\envs\pruebasTensorflow\lib\site-packages\tensorflow_core\python\ops\template.py", line 161, in make_template
    **kwargs)
```

In []:

In []: