tf.feature_column.linear_model

```
linear_model(
    features,
    feature_columns,
    units=1,
    sparse_combiner='sum',
    weight_collections=None,
    trainable=True
)
```

Defined in `tensorflow/python/feature_column/feature_column.py`.

Returns a linear prediction `Tensor` based on given `feature_columns`.

This function generates a weighted sum based on output dimension `units`. Weighted sum refers to logits in classification problems. It refers to the prediction itself for linear regression problems.

Note on supported columns: `linear_model` treats categorical columns as `indicator_column`s while `input_layer` explicitly requires wrapping each of them with an `embedding_column` or an `indicator_column`.

Example:

```
price = numeric_column('price')
price_buckets = bucketized_column(price, boundaries=[0., 10., 100., 1000.])
keywords = categorical_column_with_hash_bucket("keywords", 10K)
keywords_price = crossed_column('keywords', price_buckets, ...)
columns = [price_buckets, keywords, keywords_price ...]
features = tf.parse_example(..., features=make_parse_example_spec(columns))
prediction = linear_model(features, columns)
```

Args:

- `features`: A mapping from key to tensors. `_FeatureColumn`s look up via these keys. For example `numeric_column('price')` will look at 'price' key in this dict. Values are `Tensor` or `SparseTensor` depending on corresponding `_FeatureColumn`.
- `feature_columns`: An iterable containing the FeatureColumns to use as inputs to your model. All items should be instances of classes derived from `_FeatureColumn`s.
- `units`: An integer, dimensionality of the output space. Default value is 1.
- `sparse_combiner`: A string specifying how to reduce if a sparse column is multivalent. Currently "mean", "sqrtn" and "sum" are supported, with "sum" the default. "sqrtn" often achieves good accuracy, in particular with bag-of-words columns. It combines each sparse columns independently.
  - "sum": do not normalize features in the column
  - "mean": do l1 normalization on features in the column
  - "sqrtn": do l2 normalization on features in the column
- `weight_collections`: A list of collection names to which the Variable will be added. Note that, variables will also be added to collections `tf.GraphKeys.GLOBAL_VARIABLES` and `ops.GraphKeys.MODEL_VARIABLES`.
- `trainable`: If `True` also add the variable to the graph collection `GraphKeys.TRAINABLE_VARIABLES` (see `tf.Variable`).

Returns:

A `Tensor` which represents predictions/logits of a linear model. Its shape is (batch_size, units) and its dtype is `float32`.

Raises:

- `ValueError`: if an item in `feature_columns` is neither a `_DenseColumn` nor `_CategoricalColumn`.