

## tf.metrics.sparse\_average\_precision\_at\_k

```
sparse_average_precision_at_k(
    labels,
    predictions,
    k,
    weights=None,
    metrics_collections=None,
    updates_collections=None,
    name=None
)
```

Defined in [tensorflow/python/ops/metrics\\_impl.py](#).

Computes average precision@k of predictions with respect to sparse labels.

`sparse_average_precision_at_k` creates two local variables, `average_precision_at_<k>/total` and `average_precision_at_<k>/max`, that are used to compute the frequency. This frequency is ultimately returned as `average_precision_at_<k>`: an idempotent operation that simply divides `average_precision_at_<k>/total` by `average_precision_at_<k>/max`.

For estimation of the metric over a stream of data, the function creates an `update_op` operation that updates these variables and returns the `precision_at_<k>`. Internally, a `top_k` operation computes a `Tensor` indicating the top `k` `predictions`. Set operations applied to `top_k` and `labels` calculate the true positives and false positives weighted by `weights`. Then `update_op` increments `true_positive_at_<k>` and `false_positive_at_<k>` using these values.

If `weights` is `None`, weights default to 1. Use weights of 0 to mask values.

## Args:

- `labels`: `int64 Tensor` or `SparseTensor` with shape `[D1, ... DN, num_labels]` or `[D1, ... DN]`, where the latter implies `num_labels=1`. `N >= 1` and `num_labels` is the number of target classes for the associated prediction. Commonly, `N=1` and `labels` has shape `[batch_size, num_labels]`. `[D1, ... DN]` must match `predictions`. Values should be in range `[0, num_classes)`, where `num_classes` is the last dimension of `predictions`. Values outside this range are ignored.
- `predictions`: `Float Tensor` with shape `[D1, ... DN, num_classes]` where `N >= 1`. Commonly, `N=1` and `predictions` has shape `[batch size, num_classes]`. The final dimension contains the logit values for each class. `[D1, ... DN]` must match `labels`.
- `k`: Integer, `k` for @`k` metric. This will calculate an average precision for range `[1, k]`, as documented above.
- `weights`: `Tensor` whose rank is either 0, or `n-1`, where `n` is the rank of `labels`. If the latter, it must be broadcastable to `labels` (i.e., all dimensions must be either `1`, or the same as the corresponding `labels` dimension).
- `metrics_collections`: An optional list of collections that values should be added to.
- `updates_collections`: An optional list of collections that updates should be added to.
- `name`: Name of new update operation, and namespace for other dependent ops.

## Returns:

- `mean_average_precision`: Scalar `float64 Tensor` with the mean average precision values.
- `update`: `Operation` that increments variables appropriately, and whose value matches `metric`.

Raises:

- `ValueError` : if k is invalid.

Except as otherwise noted, the content of this page is licensed under the [Creative Commons Attribution 3.0 License](#), and code samples are licensed under the [Apache 2.0 License](#). For details, see our [Site Policies](#). Java is a registered trademark of Oracle and/or its affiliates.

Last updated November 2, 2017.

Stay Connected

- Blog
- GitHub
- Twitter

Support

- Issue Tracker
- Release Notes
- Stack Overflow