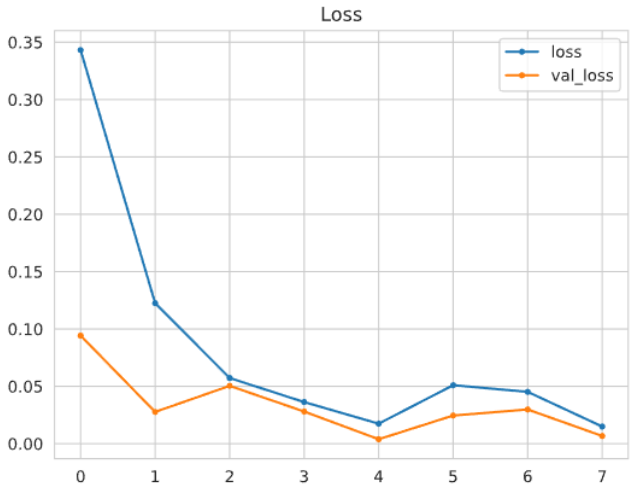
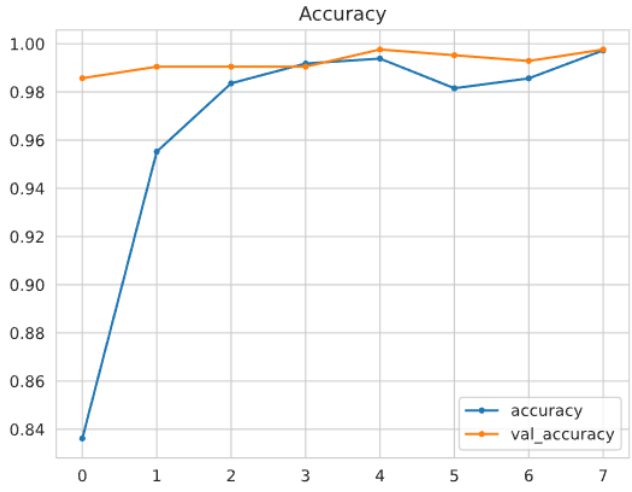
v2:

```
def create_tf_model():
    model = Sequential()

    model.add(Conv2D(filters=32, kernel_size=(3,3),input_shape=image_shape, activation='relu',))
    model.add(MaxPooling2D(pool_size=(2, 2)))

    model.add(Conv2D(filters=64, kernel_size=(3,3),input_shape=image_shape, activation='relu',))
    model.add(MaxPooling2D(pool_size=(2, 2)))

    model.add(Conv2D(filters=64, kernel_size=(3,3),input_shape=image_shape, activation='relu',))
    model.add(MaxPooling2D(pool_size=(2, 2)))

    model.add(Flatten())
    model.add(Dense(128, activation = 'relu'))

    model.add(Dropout(0.5))
    model.add(Dense(1, activation = 'sigmoid'))

    model.compile(loss='binary_crossentropy',
                  optimizer='adam',
                  metrics=['accuracy'])

    return model
```

```
early_stop = EarlyStopping(monitor='val_loss',patience=3)
```






```
    evaluation = model.evaluate(test_set)
```

```
43/43 [==============================] - 16s 364ms/step - loss: 0.0039 - accuracy: 0.9976
```

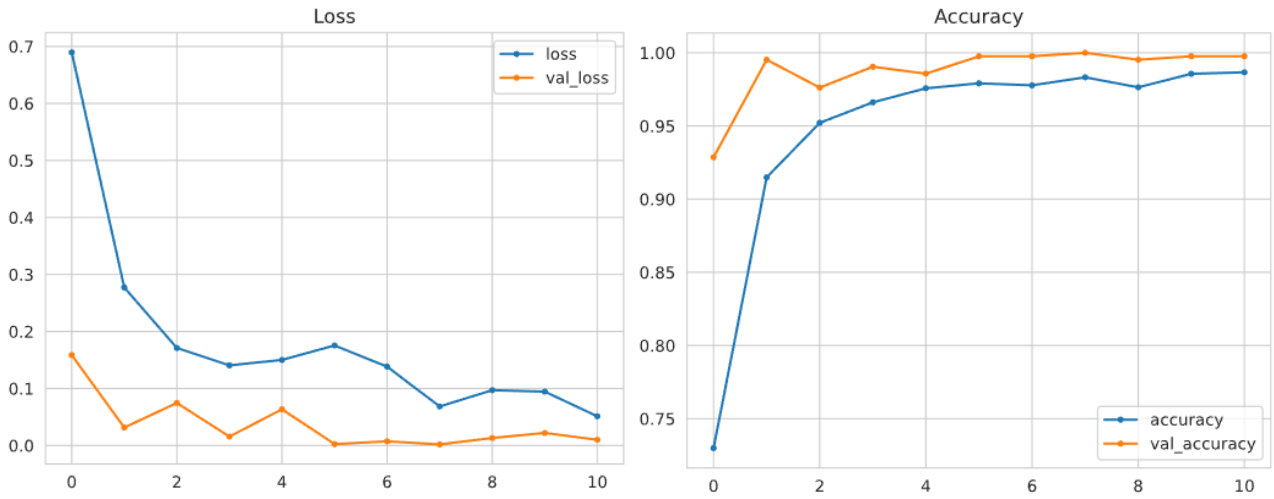
```
0.9999896963781794
healthy
```

Model accuracy plot suggests slight overfitting may occur if further epochs were run as the accuracy and val_accuracy are the same at the end of the curve.

v3:

```
def create_tf_model():
    model = Sequential()

    model.add(Conv2D(filters=32, kernel_size=(3,3),input_shape=image_shape, activation='relu',))
    model.add(MaxPooling2D(pool_size=(2, 2)))

    model.add(Conv2D(filters=64, kernel_size=(3,3),input_shape=image_shape, activation='relu',))
    model.add(MaxPooling2D(pool_size=(2, 2)))

    model.add(Conv2D(filters=64, kernel_size=(3,3),input_shape=image_shape, activation='relu',))
    model.add(MaxPooling2D(pool_size=(2, 2)))

    model.add(Flatten())
    model.add(Dense(128, activation = 'relu'))

    model.add(Dropout(0.5))
    model.add(Dense(1, activation = 'sigmoid'))

    model.compile(loss='binary_crossentropy',
                  optimizer='rmsprop',
                  metrics=['accuracy'])

    return model
```

```
early_stop = EarlyStopping(monitor='val_loss',patience=3)
```




```
evaluation = model.evaluate(test_set)
```

```
43/43 [==============================] - 17s 382ms/step - loss: 0.0441 - accuracy: 0.9953
```

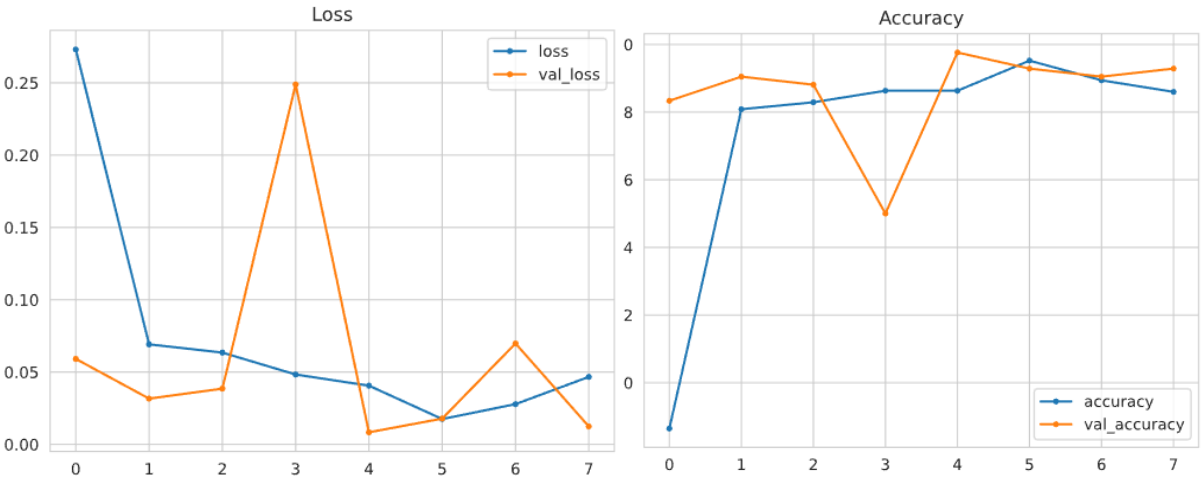
```
0.9999999817082621
healthy
```

Model accuracy plot suggests a normal fit. The loss models plot suggests that better learning is achievable.

v4:

```
def create_tf_model():
    model = Sequential()

    model.add(Conv2D(filters=32, kernel_size=(3,3),input_shape=image_shape, activation='relu',))
    model.add(MaxPooling2D(pool_size=(2, 2)))

    model.add(Conv2D(filters=64, kernel_size=(3,3),input_shape=image_shape, activation='relu',))
    model.add(MaxPooling2D(pool_size=(2, 2)))

    model.add(Conv2D(filters=64, kernel_size=(3,3),input_shape=image_shape, activation='relu',))
    model.add(MaxPooling2D(pool_size=(2, 2)))

    model.add(Flatten())
    model.add(Dense(128, activation = 'relu'))

    model.add(Dropout(0.5))
    model.add(Dense(2, activation = 'softmax'))

    model.compile(loss='categorical_crossentropy',
                  optimizer='adam',
                  metrics=['accuracy'])

    return model
```

```
early_stop = EarlyStopping(monitor='val_loss',patience=3)
```




```
    evaluation = model.evaluate(test_set)
```

```
43/43 [==============================] - 15s 348ms/step - loss: 0.0116 - accuracy: 0.9964
```

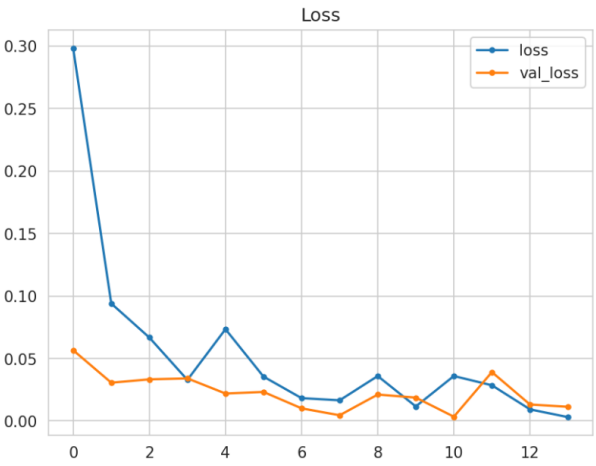
```
0.99985063
powdery_mildew
```

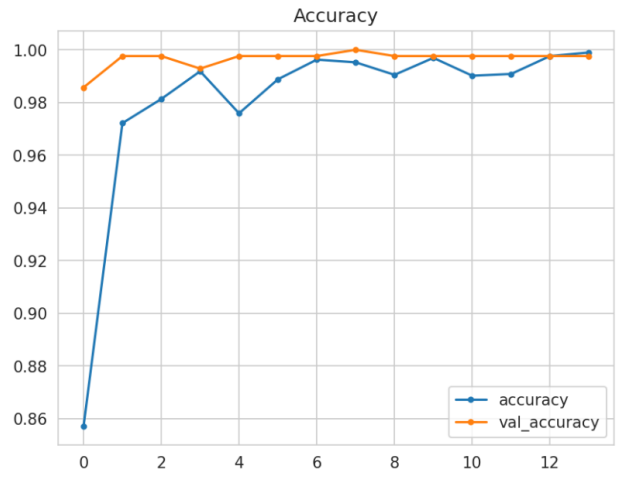
The model plots suggest poor learning. The prediction for the image was reversed, indicating an issue with the code for evaluating the model.

V5:

```
def create_tf_model():
    model = Sequential()

    model.add(Conv2D(filters=16, kernel_size=(5,5),input_shape=image_shape, activation='relu',))
    model.add(MaxPooling2D(pool_size=(2, 2)))

    model.add(Conv2D(filters=32, kernel_size=(4,4),input_shape=image_shape, activation='relu',))
    model.add(MaxPooling2D(pool_size=(2, 2)))

    model.add(Conv2D(filters=64, kernel_size=(3,3),input_shape=image_shape, activation='relu',))
    model.add(MaxPooling2D(pool_size=(2, 2)))

    model.add(Conv2D(filters=32, kernel_size=(3,3),input_shape=image_shape, activation='relu',))
    model.add(MaxPooling2D(pool_size=(2, 2)))

    model.add(Flatten())
    model.add(Dense(64, activation = 'relu'))

    model.add(Dropout(0.3))
    model.add(Dense(1, activation = 'sigmoid'))

    model.compile(loss='binary_crossentropy',
                  optimizer='adam',
                  metrics=['accuracy'])

    return model
```

```
early_stop = EarlyStopping(monitor='val_loss',patience=3)
```






```
evaluation = model.evaluate(test_set)
```

```
27/27 [==============================] - 13s 472ms/step - loss: 1.8114e-05 - accuracy: 1.0000
```

```
0.9999972517478
healthy
```

Model accuracy plot suggests slight overfitting as the loss and accuracy lines have crossed the corresponding val_loss and val_accuracy lines.

V6:

The first v6 created gave the following error during model evaluation:

Load saved model

In [3]:
```
from keras.models import load_model
model = load_model('outputs/v6/mildew_detector_model.h5')
```

```
---------------------------------------------------------------------------
OSError                                   Traceback (most recent call last)
Cell In[3], line 2
      1 from keras.models import load_model
----> 2 model = load_model('outputs/v6/mildew_detector_model.h5')

File ~/.local/lib/python3.8/site-packages/keras/saving/save.py:205, in load_model(filepath, custom_objects, compile, options)
    203       filepath = path_to_string(filepath)
    204       if isinstance(filepath, str):
--> 205         return saved_model_load.load(filepath, compile, options)
    207 raise IOError(
    208     'Unable to load model. Filepath is not an hdf5 file (or h5py is not '
    209     'available) or SavedModel.')

File ~/.local/lib/python3.8/site-packages/keras/saving/saved_model/load.py:108, in load(path, compile, options)
    103 # TODO(kathywu): Add saving/loading of optimizer, compiled losses and metrics.
    104 # TODO(kathywu): Add code to load from objects that contain all endpoints
    105
    106 # Look for metadata file or parse the SavedModel
    107 metadata = saved_metadata_pb2.SavedMetadata()
--> 108 meta_graph_def = tf.__internal__.saved_model.parse_saved_model(path).meta_graphs[0]
    109 object_graph_def = meta_graph_def.object_graph_def
    110 path_to_metadata_pb = os.path.join(path, constants.SAVED_METADATA_PATH)

File ~/.local/lib/python3.8/site-packages/tensorflow/python/saved_model/loader_impl.py:118, in parse_saved_model(export_dir)
    116     raise IOError("Cannot parse file %s: %s." % (path_to_pbtxt, str(e)))
    117 else:
--> 118   raise IOError(
    119       "SavedModel file does not exist at: %s%s{%s|%s}" %
    120       (export_dir, os.path.sep, constants.SAVED_MODEL_FILENAME_PBTXT,
    121        constants.SAVED_MODEL_FILENAME_PB))

OSError: SavedModel file does not exist at: outputs/v6/mildew_detector_model.h5/{saved_model.pbtxt|saved_model.pb}
```

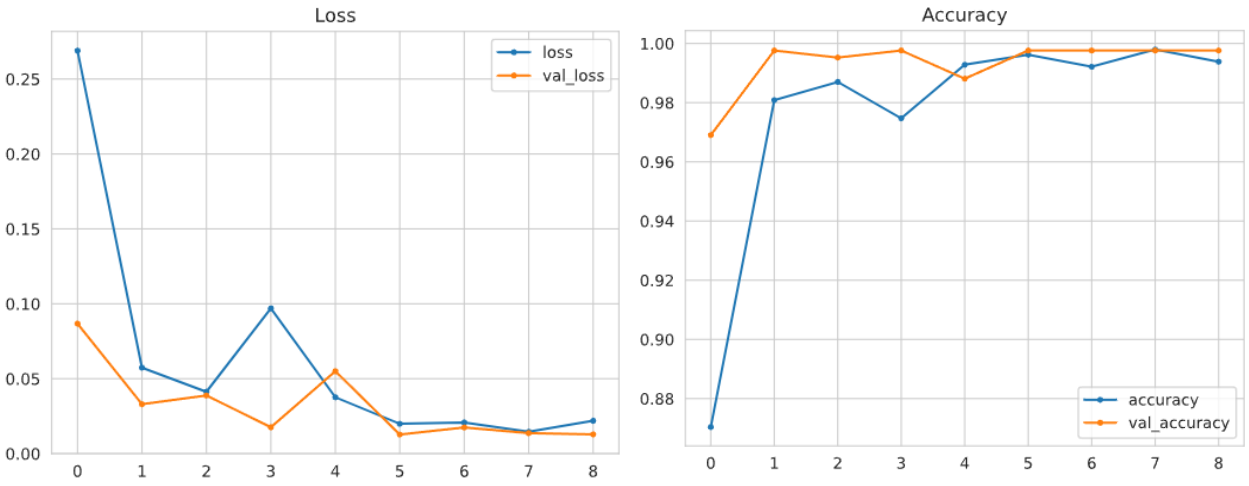
To test if the model was created satisfactorily, the model was tested on the dashboard which also gave an error.

Due to the error, this version was overwritten with the next model created.

V6:

```
def create_tf_model():
    model = Sequential()

    model.add(Conv2D(filters=32, kernel_size=(3,3),input_shape=image_shape, activation='relu',))
    model.add(MaxPooling2D(pool_size=(2, 2)))

    model.add(Conv2D(filters=16, kernel_size=(3,3),input_shape=image_shape, activation='relu',))
    model.add(MaxPooling2D(pool_size=(2, 2)))

    model.add(Conv2D(filters=8, kernel_size=(3,3),input_shape=image_shape, activation='relu',))
    model.add(MaxPooling2D(pool_size=(2, 2)))

    model.add(Flatten())
    model.add(Dense(64, activation = 'relu'))

    model.add(Dropout(0.3))
    model.add(Dense(2, activation = 'softmax'))

    model.compile(loss='categorical_crossentropy',
                  optimizer='adam',
                  metrics=['accuracy'])

    return model
```

```
early_stop = EarlyStopping(monitor='val_loss',patience=3)
```




```
evaluation = model.evaluate(test_set)
```

```
43/43 [==============================] - 21s 457ms/step - loss: 0.0015 - accuracy: 0.9988
```

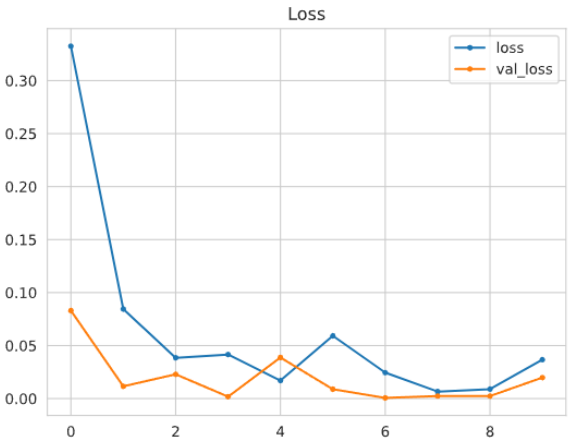
```
0.99999964
powdery_mildew
```

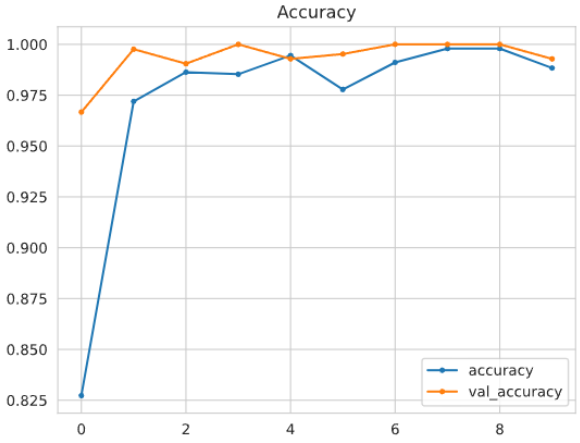
The model plots suggest a normal fit. Like the previous softmax model (v4) the prediction for the image was reversed, the code was reviewed and fixed (refer to bug section of ReadMe).

V7:

```
def create_tf_model():
    model = Sequential()

    model.add(Conv2D(filters=32, kernel_size=(3,3),input_shape=image_shape, activation='relu',))
    model.add(MaxPooling2D(pool_size=(2, 2)))

    model.add(Conv2D(filters=16, kernel_size=(3,3),input_shape=image_shape, activation='relu',))
    model.add(MaxPooling2D(pool_size=(2, 2)))

    model.add(Conv2D(filters=16, kernel_size=(3,3),input_shape=image_shape, activation='relu',))
    model.add(MaxPooling2D(pool_size=(2, 2)))

    model.add(Flatten())
    model.add(Dense(64, activation = 'relu'))

    model.add(Dropout(0.3))
    model.add(Dense(2, activation = 'softmax'))

    model.compile(loss='categorical_crossentropy',
                  optimizer='adam',
                  metrics=['accuracy'])

    return model
```

```
early_stop = EarlyStopping(monitor='val_loss',patience=3)
```






```
evaluation = model.evaluate(test_set)
```

```
43/43 [==============================] - 17s 400ms/step - loss: 0.0035 - accuracy: 1.0000
```

```
0.9999974
healthy
```

The model plots suggest slight overfitting.