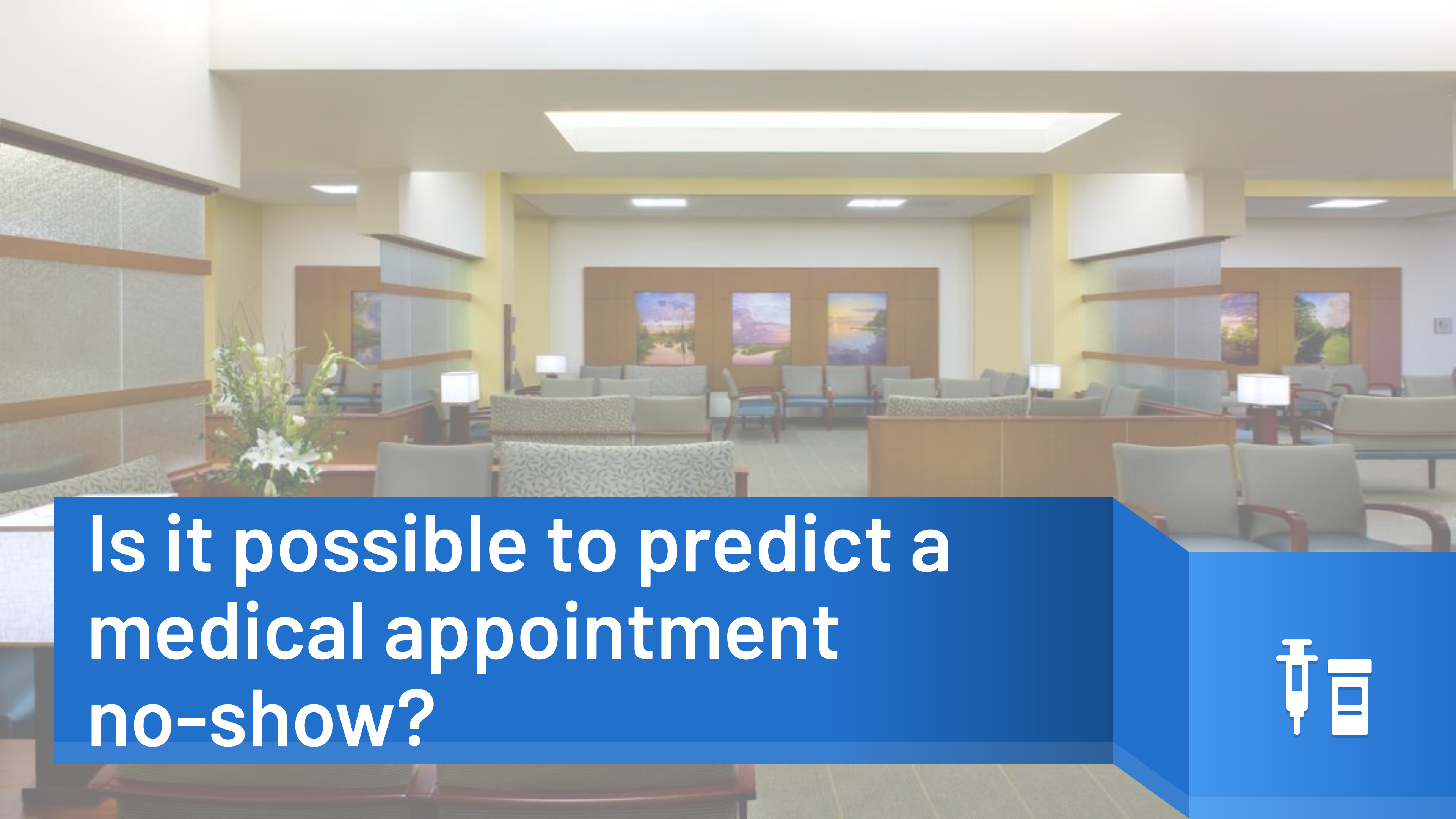


Is it possible to predict a medical appointment no-show?

Objectives

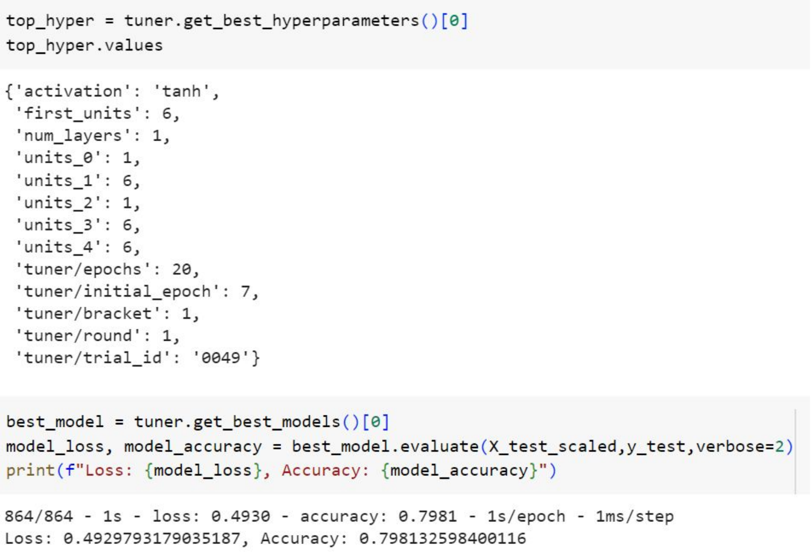
1. Create a model trained on Kaggle data of over 110,000 appointments to help us predict if any given patient will not make it to their appointment (No-show).
2. Create secondary models removing individual categories to see if removing a single factor improves accuracy in predicting No-shows
3. Create charts to help visualize the individual categories and to see if any one group disproportionately No-shows more than the others.

Project 4 by Group5 = Scott Mclean,Nick Leisenring,Cam Pulling,Trevor Moore

Original Model

Model Factors

- Age
- Gender
- Scholarship
- Hypertension
- Diabetes
- Alcoholism
- Handicap
- SMS Received



```
top_hyper = tuner.get_best_hyperparameters()[0]
top_hyper.values
```

```
{'activation': 'tanh',
 'first_units': 6,
 'num_layers': 1,
 'units_0': 1,
 'units_1': 6,
 'units_2': 1,
 'units_3': 6,
 'units_4': 6,
 'tuner/epochs': 20,
 'tuner/initial_epoch': 7,
 'tuner/bracket': 1,
 'tuner/round': 1,
 'tuner/trial_id': '0049'}
```

```
best_model = tuner.get_best_models()[0]
model_loss, model_accuracy = best_model.evaluate(X_test_scaled,y_test,verbose=2)
print(f"Loss: {model_loss}, Accuracy: {model_accuracy}")
```

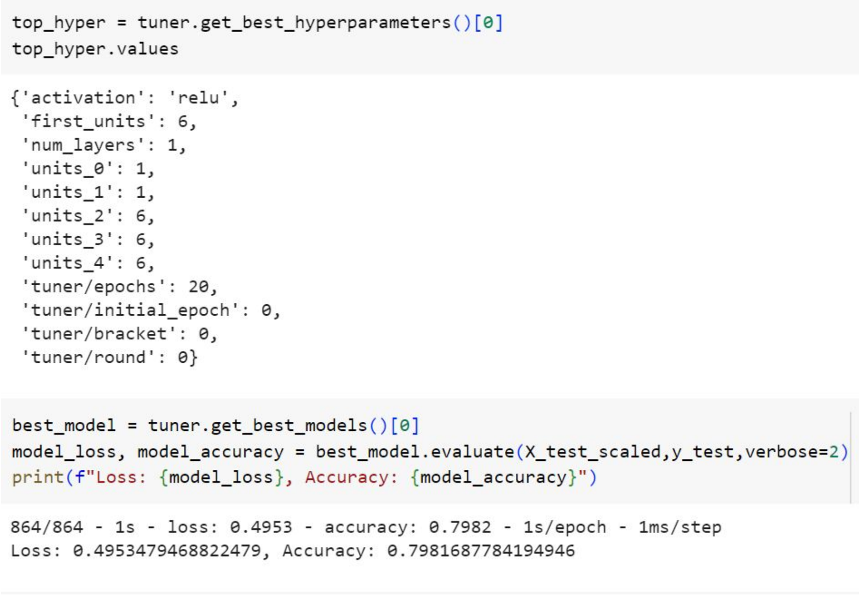
```
864/864 - 1s - loss: 0.4930 - accuracy: 0.7981 - 1s/epoch - 1ms/step
Loss: 0.4929793179035187, Accuracy: 0.798132598400116
```

Information Removed

- Patient ID
- Appointment ID
- Scheduled Day
- Appointment Day
- Neighborhood

No Age Data

- Age range goes from pre-born to 115 years old
- Model with age dropped is just as accurate as the original model at 79.8%



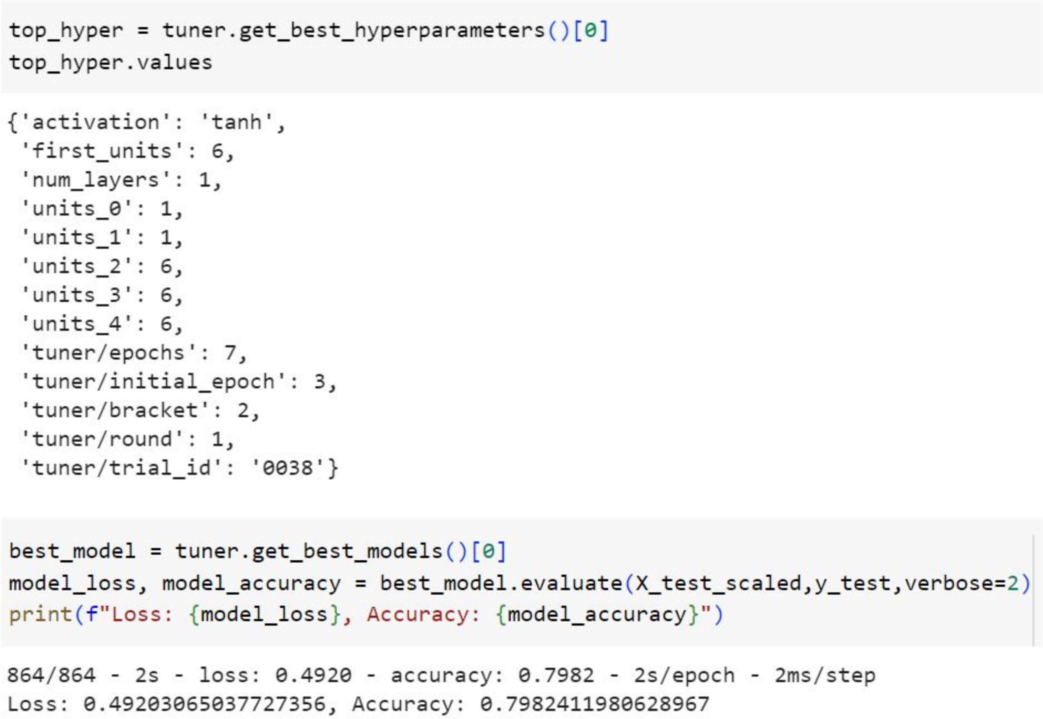
```
top_hyper = tuner.get_best_hyperparameters()[0]
top_hyper.values
```

```
{'activation': 'relu',
 'first_units': 6,
 'num_layers': 1,
 'units_0': 1,
 'units_1': 1,
 'units_2': 6,
 'units_3': 6,
 'units_4': 6,
 'tuner/epochs': 20,
 'tuner/initial_epoch': 0,
 'tuner/bracket': 0,
 'tuner/round': 0}
```

```
best_model = tuner.get_best_models()[0]
model_loss, model_accuracy = best_model.evaluate(X_test_scaled,y_test,verbose=2)
print(f"Loss: {model_loss}, Accuracy: {model_accuracy}")
```

```
864/864 - 1s - loss: 0.4953 - accuracy: 0.7982 - 1s/epoch - 1ms/step
Loss: 0.4953479468822479, Accuracy: 0.7981687784194946
```

No Gender Data



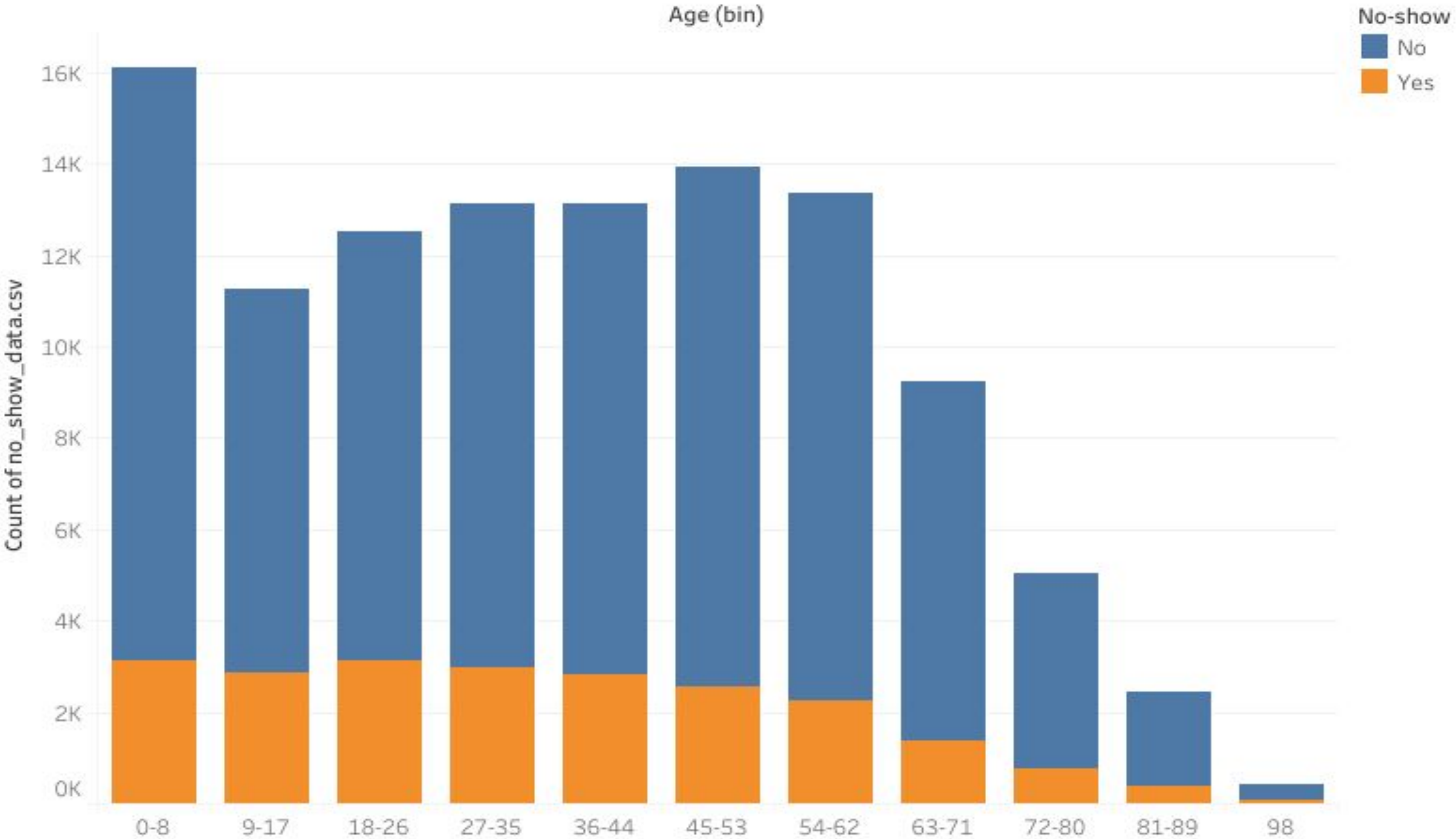
```
top_hyper = tuner.get_best_hyperparameters()[0]
top_hyper.values
```

```
{'activation': 'tanh',
 'first_units': 6,
 'num_layers': 1,
 'units_0': 1,
 'units_1': 1,
 'units_2': 6,
 'units_3': 6,
 'units_4': 6,
 'tuner/epochs': 7,
 'tuner/initial_epoch': 3,
 'tuner/bracket': 2,
 'tuner/round': 1,
 'tuner/trial_id': '0038'}
```

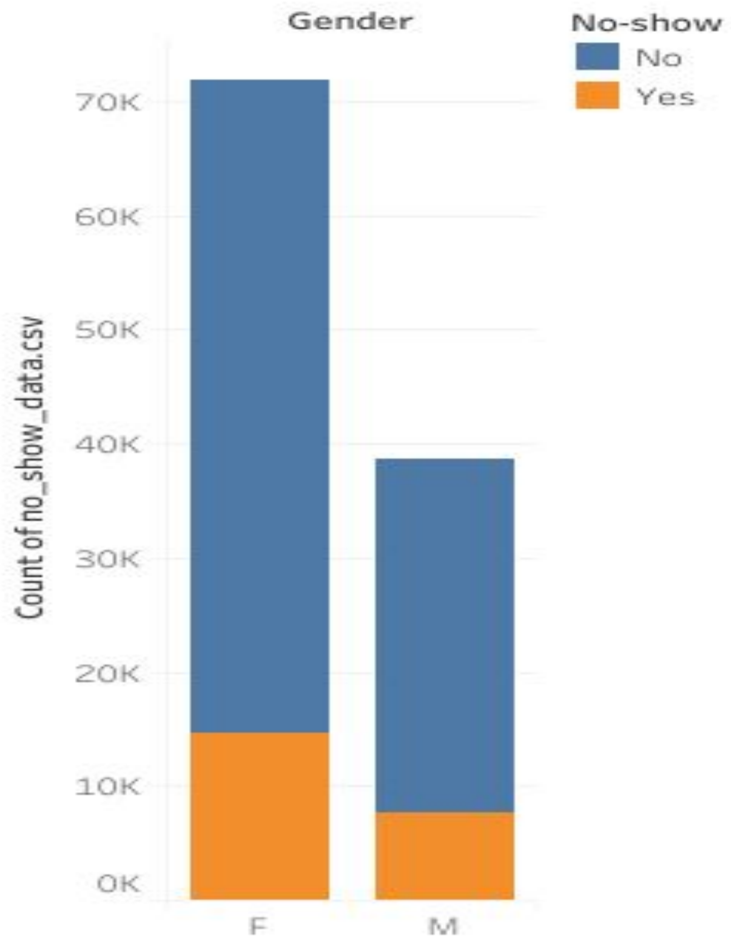
```
best_model = tuner.get_best_models()[0]
model_loss, model_accuracy = best_model.evaluate(X_test_scaled,y_test,verbose=2)
print(f"Loss: {model_loss}, Accuracy: {model_accuracy}")
```

```
864/864 - 2s - loss: 0.4920 - accuracy: 0.7982 - 2s/epoch - 2ms/step
Loss: 0.49203065037727356, Accuracy: 0.7982411980628967
```

No-shows by age ranges



No-Shows by Gender

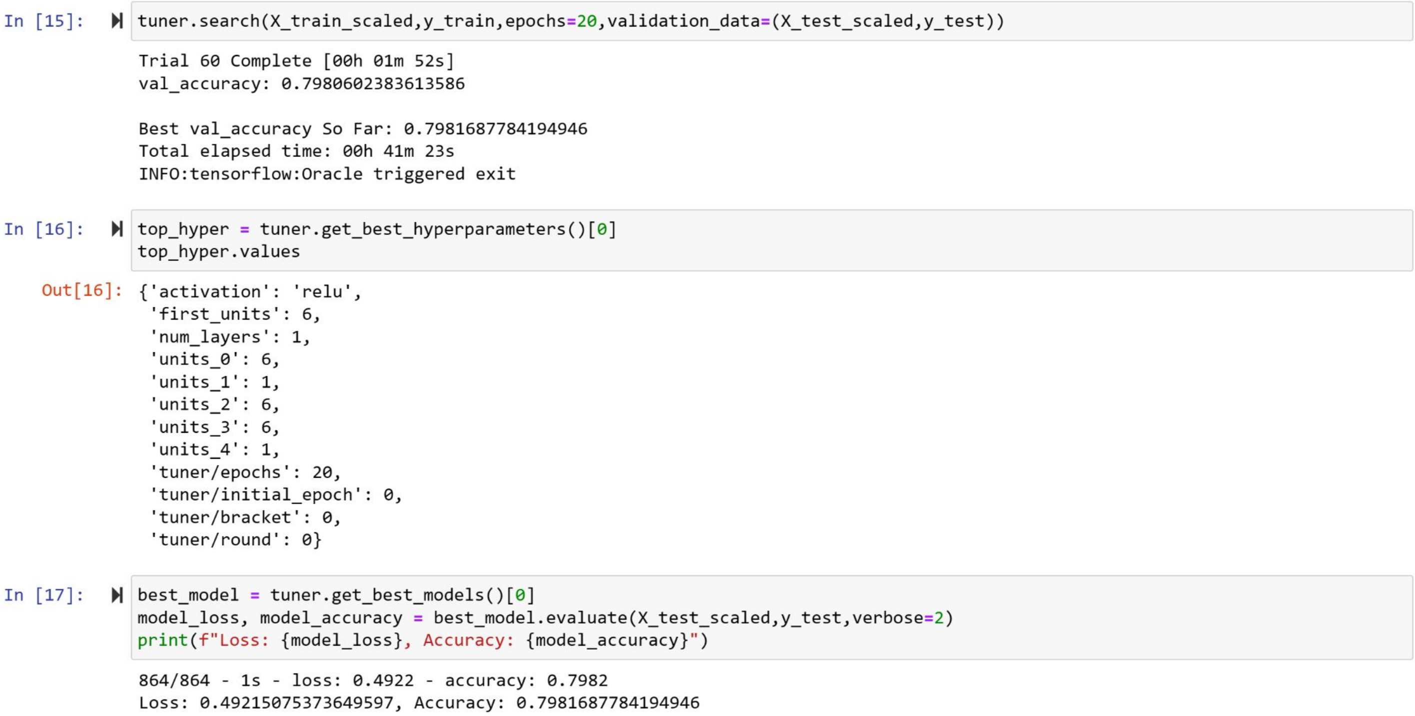


Women had roughly twice the amount of appointments as men however they No-showed at nearly identical rates.

Females No-showed at a 20.3% rate while Males were at 20.0%.

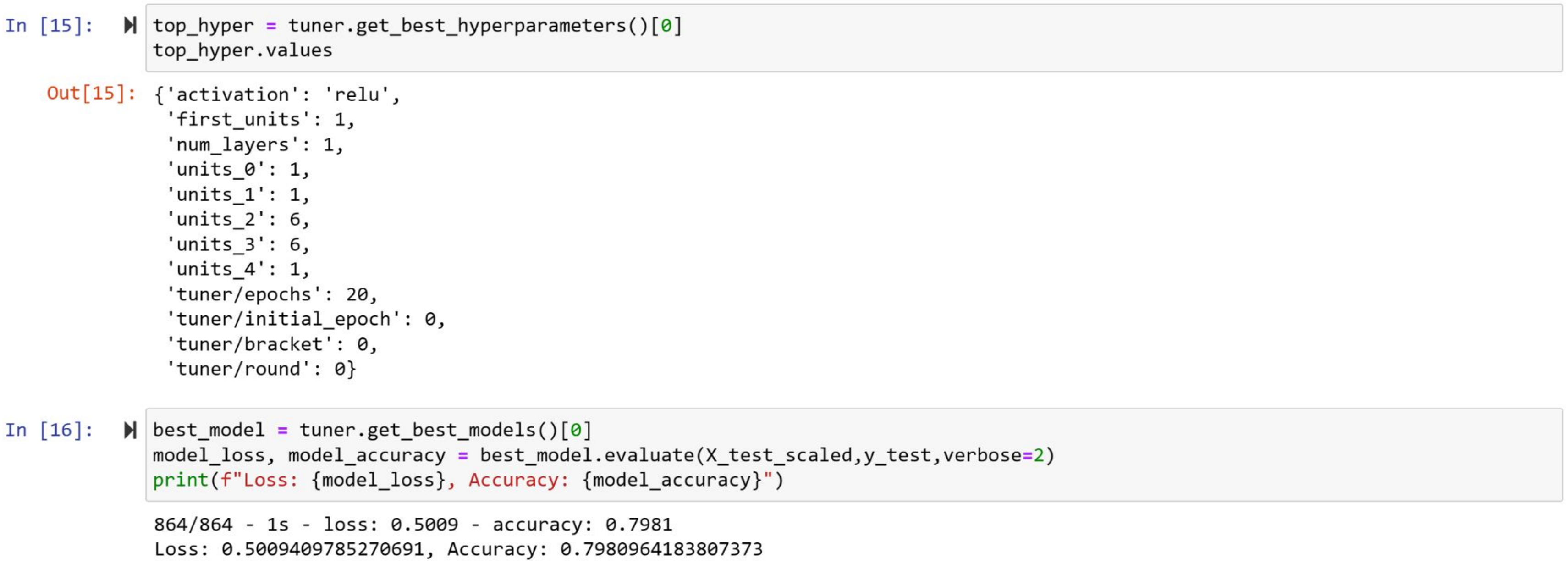
Gender is not a very determinative factor in a patient No-showing

Without Scholarship



```
In [15]:  tuner.search(X_train_scaled,y_train,epochs=20,validation_data=(X_test_scaled,y_test))

          Trial 60 Complete [00h 01m 52s]
          val_accuracy: 0.7980602383613586

          Best val_accuracy So Far: 0.7981687784194946
          Total elapsed time: 00h 41m 23s
          INFO:tensorflow:Oracle triggered exit

In [16]:  top_hyper = tuner.get_best_hyperparameters()[0]
          top_hyper.values

Out[16]:  {'activation': 'relu',
           'first_units': 6,
           'num_layers': 1,
           'units_0': 6,
           'units_1': 1,
           'units_2': 6,
           'units_3': 6,
           'units_4': 1,
           'tuner/epochs': 20,
           'tuner/initial_epoch': 0,
           'tuner/bracket': 0,
           'tuner/round': 0}

In [17]:  best_model = tuner.get_best_models()[0]
          model_loss, model_accuracy = best_model.evaluate(X_test_scaled,y_test,verbose=2)
          print(f"Loss: {model_loss}, Accuracy: {model_accuracy}")

          864/864 - 1s - loss: 0.4922 - accuracy: 0.7982
          Loss: 0.49215075373649597, Accuracy: 0.7981687784194946
```

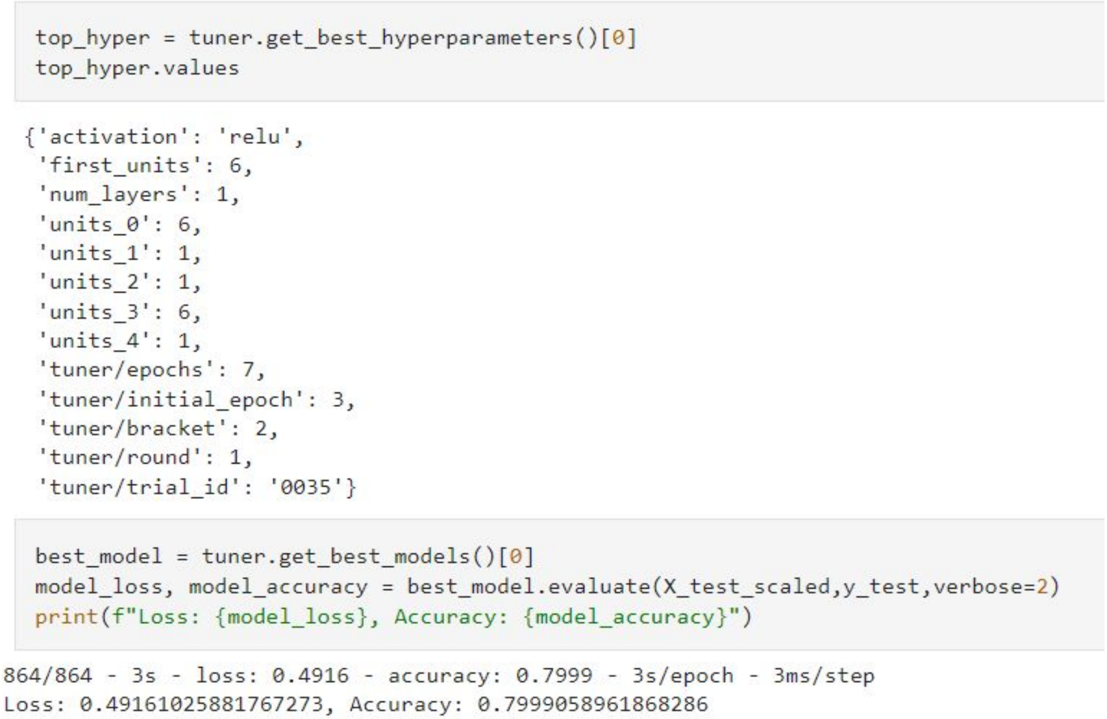
Without Hypertension



```
In [15]:  top_hyper = tuner.get_best_hyperparameters()[0]
          top_hyper.values

Out[15]: {'activation': 'relu',
          'first_units': 1,
          'num_layers': 1,
          'units_0': 1,
          'units_1': 1,
          'units_2': 6,
          'units_3': 6,
          'units_4': 1,
          'tuner/epochs': 20,
          'tuner/initial_epoch': 0,
          'tuner/bracket': 0,
          'tuner/round': 0}
```

```
In [16]:  best_model = tuner.get_best_models()[0]
          model_loss, model_accuracy = best_model.evaluate(X_test_scaled,y_test,verbose=2)
          print(f"Loss: {model_loss}, Accuracy: {model_accuracy}")

          864/864 - 1s - loss: 0.5009 - accuracy: 0.7981
          Loss: 0.5009409785270691, Accuracy: 0.7980964183807373
```

No Handicap



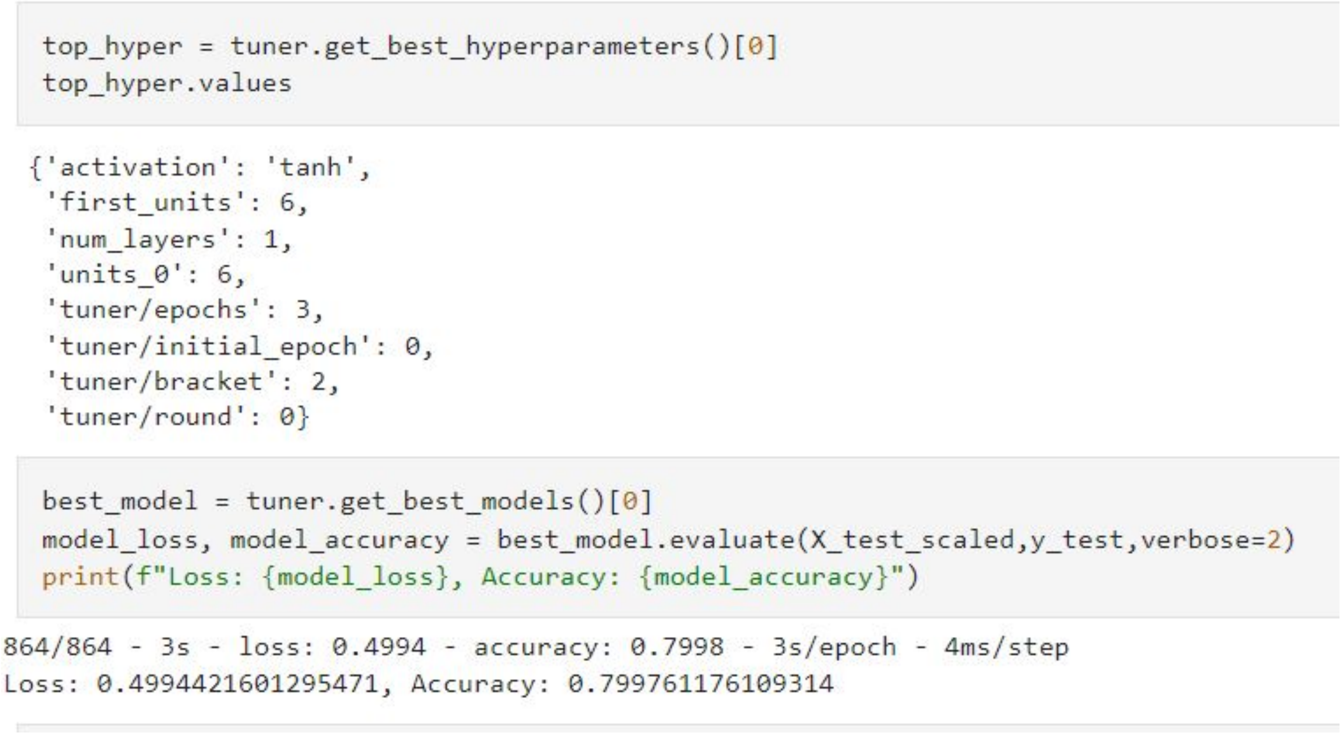
```
top_hyper = tuner.get_best_hyperparameters()[0]
top_hyper.values
```

```
{'activation': 'relu',
 'first_units': 6,
 'num_layers': 1,
 'units_0': 6,
 'units_1': 1,
 'units_2': 1,
 'units_3': 6,
 'units_4': 1,
 'tuner/epochs': 7,
 'tuner/initial_epoch': 3,
 'tuner/bracket': 2,
 'tuner/round': 1,
 'tuner/trial_id': '0035'}
```

```
best_model = tuner.get_best_models()[0]
model_loss, model_accuracy = best_model.evaluate(X_test_scaled,y_test,verbose=2)
print(f"Loss: {model_loss}, Accuracy: {model_accuracy}")
```

```
864/864 - 3s - loss: 0.4916 - accuracy: 0.7999 - 3s/epoch - 3ms/step
Loss: 0.4916102588176727 3, Accuracy: 0.7999058961868286
```

No Text Message Received Data



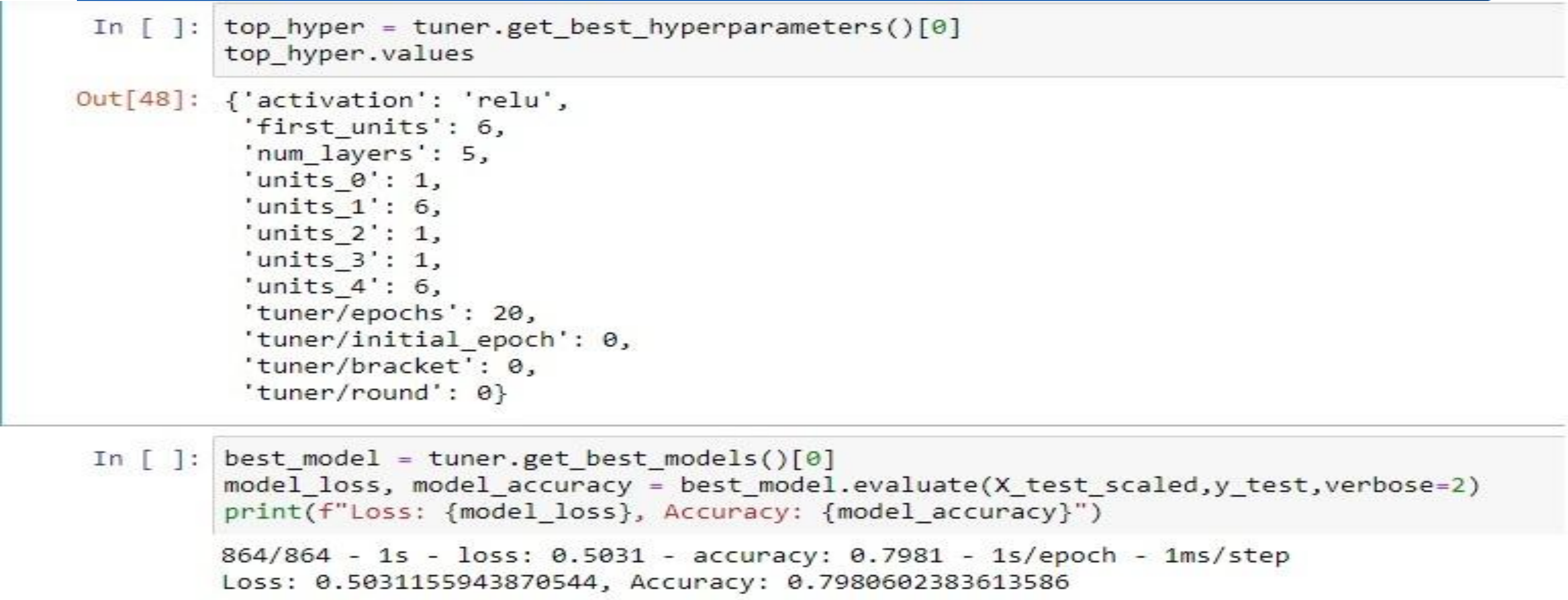
```
top_hyper = tuner.get_best_hyperparameters()[0]
top_hyper.values
```

```
{'activation': 'tanh',
 'first_units': 6,
 'num_layers': 1,
 'units_0': 6,
 'tuner/epochs': 3,
 'tuner/initial_epoch': 0,
 'tuner/bracket': 2,
 'tuner/round': 0}
```

```
best_model = tuner.get_best_models()[0]
model_loss, model_accuracy = best_model.evaluate(X_test_scaled,y_test,verbose=2)
print(f"Loss: {model_loss}, Accuracy: {model_accuracy}")
```

```
864/864 - 3s - loss: 0.4994 - accuracy: 0.7998 - 3s/epoch - 4ms/step
Loss: 0.4994421601295471, Accuracy: 0.799761176109314
```

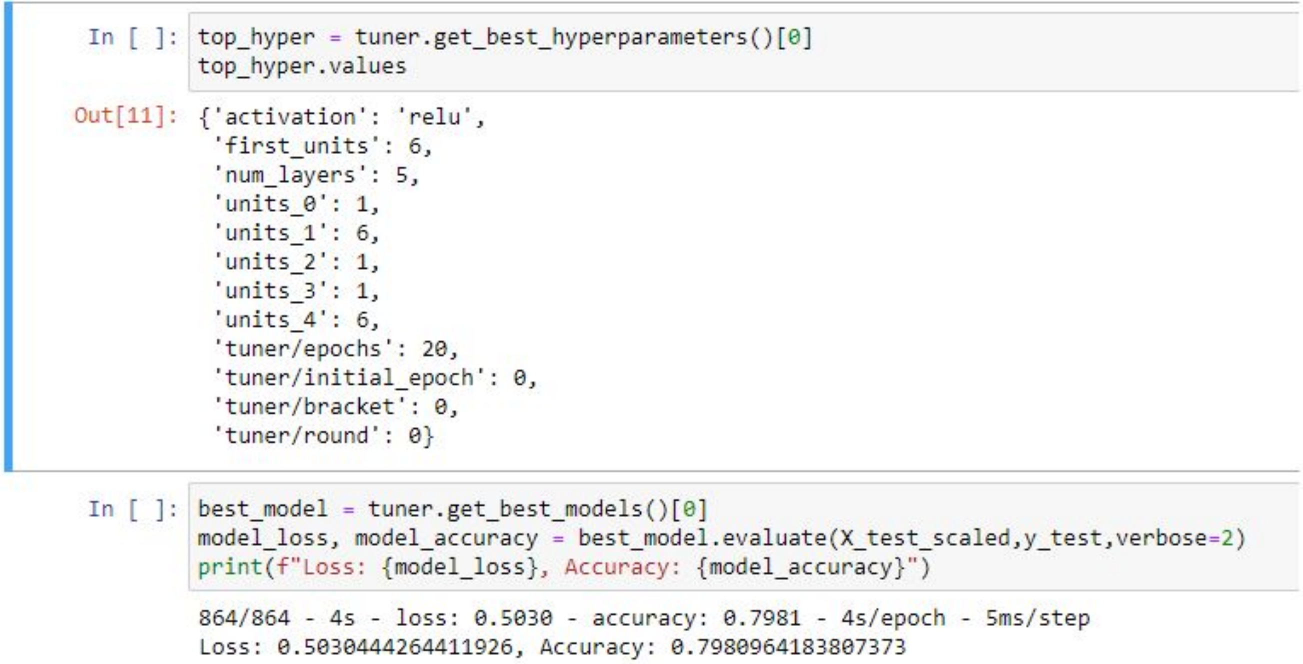
No Diabetes Data



```
In [ ]: top_hyper = tuner.get_best_hyperparameters()[0]
        top_hyper.values

Out[48]: {'activation': 'relu',
          'first_units': 6,
          'num_layers': 5,
          'units_0': 1,
          'units_1': 6,
          'units_2': 1,
          'units_3': 1,
          'units_4': 6,
          'tuner/epochs': 20,
          'tuner/initial_epoch': 0,
          'tuner/bracket': 0,
          'tuner/round': 0}
```

```
In [ ]: best_model = tuner.get_best_models()[0]
        model_loss, model_accuracy = best_model.evaluate(X_test_scaled,y_test,verbose=2)
        print(f"Loss: {model_loss}, Accuracy: {model_accuracy}")

        864/864 - 1s - loss: 0.5031 - accuracy: 0.7981 - 1s/epoch - 1ms/step
        Loss: 0.5031155943870544, Accuracy: 0.7980602383613586
```

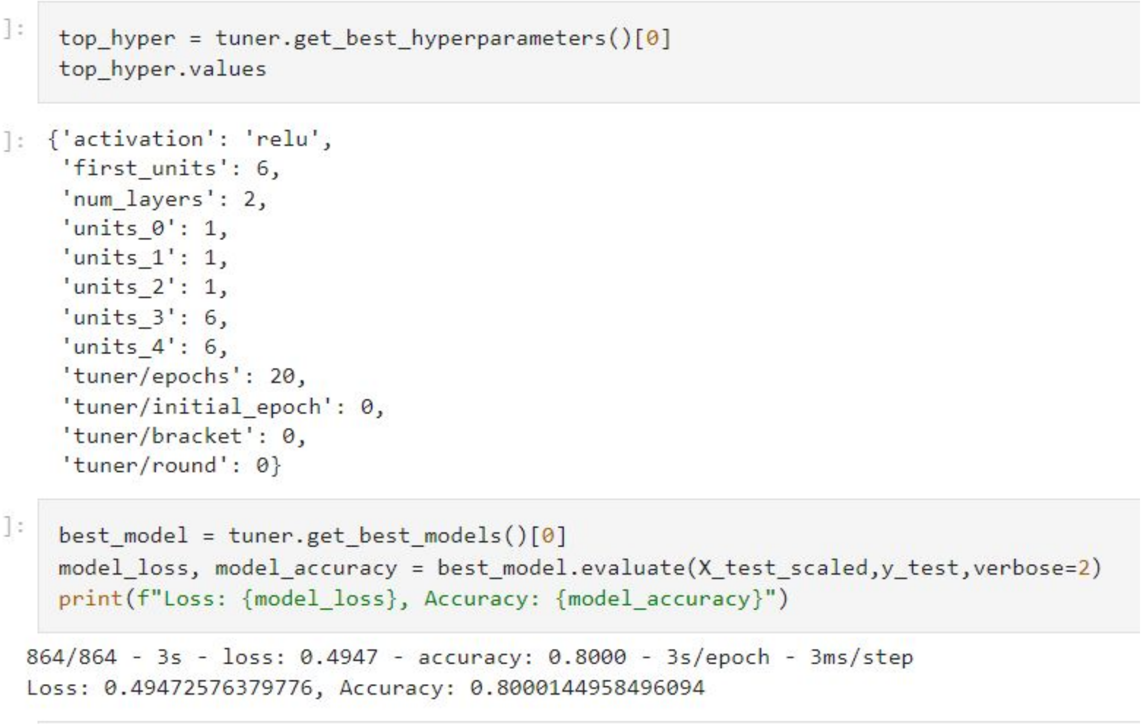
No Alcoholism Data



```
In [ ]: top_hyper = tuner.get_best_hyperparameters()[0]
        top_hyper.values

Out[11]: {'activation': 'relu',
          'first_units': 6,
          'num_layers': 5,
          'units_0': 1,
          'units_1': 6,
          'units_2': 1,
          'units_3': 1,
          'units_4': 6,
          'tuner/epochs': 20,
          'tuner/initial_epoch': 0,
          'tuner/bracket': 0,
          'tuner/round': 0}
```

```
In [ ]: best_model = tuner.get_best_models()[0]
        model_loss, model_accuracy = best_model.evaluate(X_test_scaled,y_test,verbose=2)
        print(f"Loss: {model_loss}, Accuracy: {model_accuracy}")

        864/864 - 4s - loss: 0.5030 - accuracy: 0.7981 - 4s/epoch - 5ms/step
        Loss: 0.5030444264411926, Accuracy: 0.7980964183807373
```

With Days of the Week



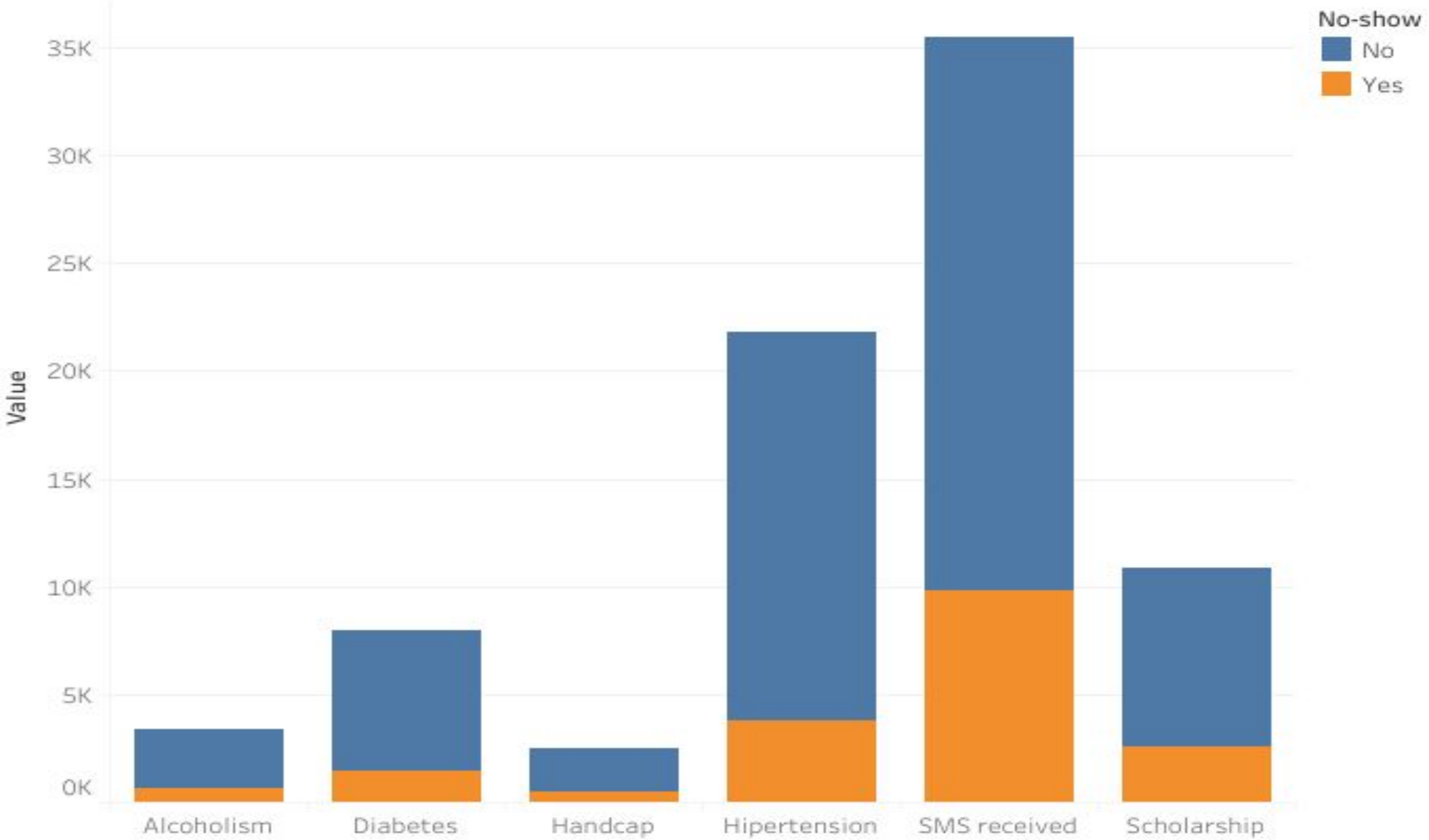
```
top_hyper = tuner.get_best_hyperparameters()[0]
top_hyper.values
```

```
{'activation': 'relu',
 'first_units': 6,
 'num_layers': 2,
 'units_0': 1,
 'units_1': 1,
 'units_2': 1,
 'units_3': 6,
 'units_4': 6,
 'tuner/epochs': 20,
 'tuner/initial_epoch': 0,
 'tuner/bracket': 0,
 'tuner/round': 0}
```

```
best_model = tuner.get_best_models()[0]
model_loss, model_accuracy = best_model.evaluate(X_test_scaled,y_test,verbose=2)
print(f"Loss: {model_loss}, Accuracy: {model_accuracy}")
```

```
864/864 - 3s - loss: 0.4947 - accuracy: 0.8000 - 3s/epoch - 3ms/step
Loss: 0.49472576379776, Accuracy: 0.8000144958496094
```

Visualization of No-shows by category



110,527 Records

22,318 No-shows

20% Of Appointments were no-shows

Conclusion

After reviewing all of our models, we can conclude that none of the categories in our dataset were directly responsible for a no-show.

We can confirm that within our dataset, Females and people who did receive a SMS Message accounted for most of the no-shows.

Thanks!

Any questions?