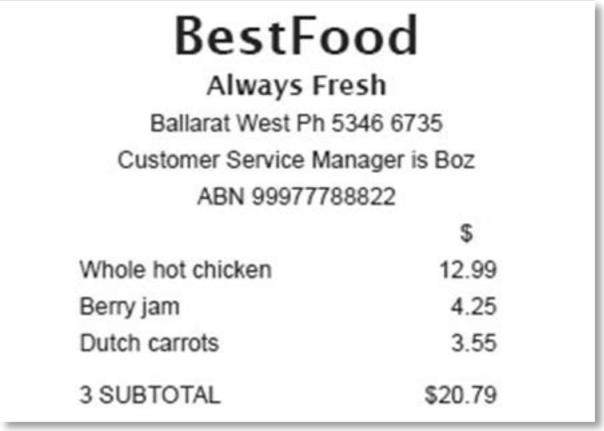
Receipt Categorizer – Machine Learning Model

Problem Statement :

Given an image of a receipt, identify and categorize the following elements

1. Vendor Name
2. Vendor Address
3. Item Name
4. Item Price
5. Total Label
6. Total Price
7. Other



BestFood

Always Fresh

Ballarat West Ph 5346 6735

Customer Service Manager is Boz

ABN 99977788822

	$
Whole hot chicken	12.99
Berry jam	4.25
Dutch carrots	3.55
3 SUBTOTAL	$20.79

Activate TensorFlow Virtual Environment

For MAC :

```
cd <TensorFlow path>
source bin\activate
```

For Windows :

```
cd <TensorFlow path>
.\tensorflow\Scripts\activate
```

Data Modelling

We are going to create a supervised machine learning model to help identify and categorize the receipt elements.

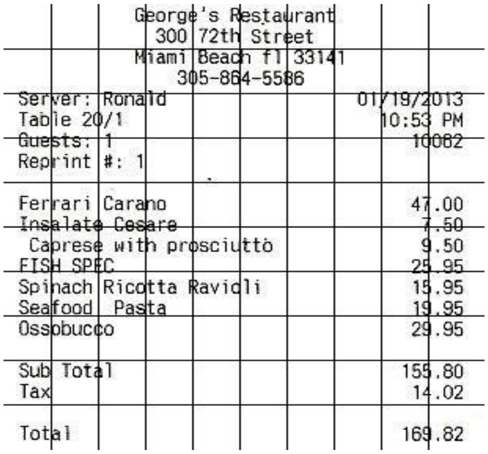
Model - A model defines the relationship between features and label, it is the thing that is doing the predicting. We're going to create a model in this module of the workshop.

The inputs to a Model are called **Features**.
The outputs of a Model are called **Labels**.
A label is the thing we're predicting, in our case it is the classification of receipt elements

EXERCISE 1 - Feature Engineering:

Features - A feature is an input variable describing our data, it is the way that we represent our data it to the machine learning system — the x variable in simple linear regression. A simple machine learning project might use a single feature, while a more sophisticated machine learning project could use millions of features

You must create a **representation** of the data to provide the model with a useful vantage point into the data's key qualities. That is, in order to train a model, you must choose the set of features that best represent the data.

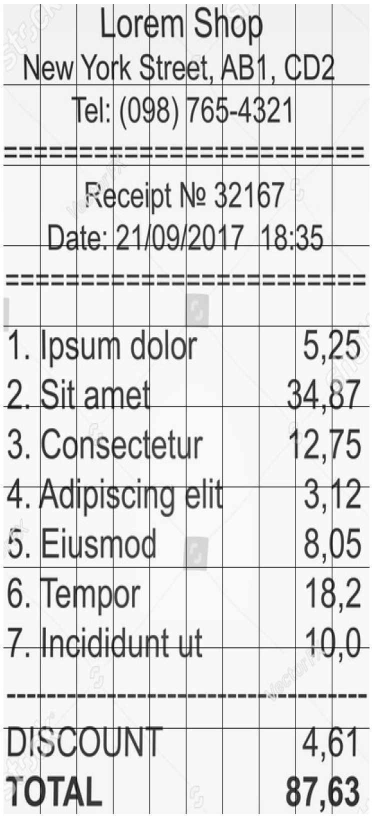


```
      George's Restaurant
        300 72th Street
      Miami Beach fl 33141
         305-864-5586
Server: Ronald           01/19/2013
Table 20/1                 10:53 PM
Guests: 1                     10082
Reprint #: 1

Ferrari Carano                47.00
Insalate Cesare                7.50
 Caprese with prosciutto       9.50
FISH SPEC                     25.95
Spinach Ricotta Ravioli       15.95
Seafood  Pasta                19.95
Ossobucco                     29.95

Sub Total                    155.80
Tax                           14.02

Total                        169.82
```



```
        Lorem Shop
  New York Street, AB1, CD2
     Tel: (098) 765-4321
========================
      Receipt № 32167
  Date: 21/09/2017  18:35
========================

1. Ipsum dolor          5,25
2. Sit amet            34,87
3. Consectetur         12,75
4. Adipiscing elit      3,12
5. Eiusmod              8,05
6. Tempor               18,2
7. Incididunt ut        10,0
----------------------------
DISCOUNT                4,61
TOTAL                  87,63
```

Consider the above receipts - which vary in dimensions. If we divide each receipt into a 10 X 10 grid the following patterns can be observed (which will hold true for majority of receipts) :

1. The Vendor Name appears in Row 1
2. The Total appears in Row 10
3. All prices are numeric
4. All the Item Names are aligned with the 'Total' label and appear around Column 1.
5. All the Item Prices are aligned with the Total Price and appear around Column 9.

Keeping in mind these observations, the following will be the features in our model

1. xLeftRel – The relative x-axis position of the top-left corner of each block of text in the receipt

   xLeftRel = (100 * TopLeft.x) / Image Width

2. yTopRel – The relative y-axis position of the top-left corner of each block of text in the receipt

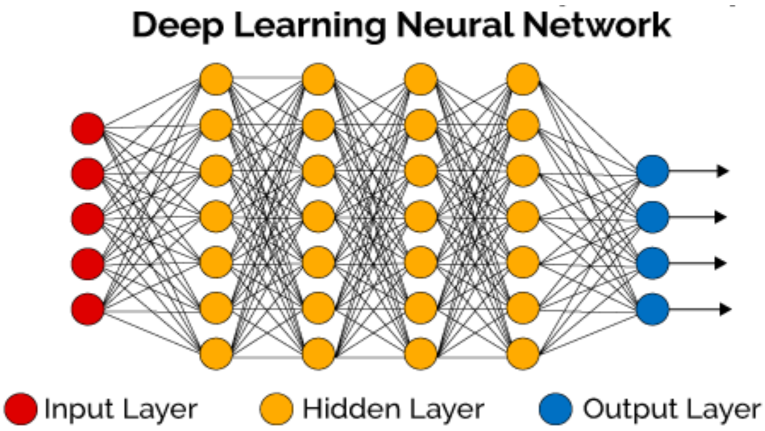
   yTopRel = (100 * TopRight.y) / Image Height

3. IsNumeric – Set to true if the block if text is Numeric
4. IsTotal – Set to true if the block of text is the 'Total' label in the receipt.

Do It Yourself: In real life Data Scientists spend a significant amount of time in Feature Engineering as it is one of the most important aspects of machine learning.
What other features can you think of to improve the accuracy of the model in the given use case?

EXERCISE 2 – Create a Model

A deep neural network (DNN) is an artificial neural network (ANN) with multiple layers between the input and output layers. The DNN finds the correct mathematical manipulation to turn the input into the output, whether it be a linear relationship or a non-linear relationship. The network moves through the layers calculating the probability of each output.



tf.estimator.DNNClassifier - A classifier for TensorFlow DNN models.

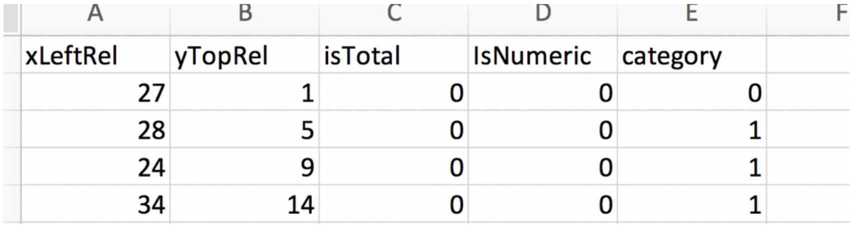
hidden_units: Iterable of number hidden units per layer. All layers are fully connected. Ex. [64, 32] means first layer has 64 nodes and second one has 32.
feature_columns: An iterable containing all the feature columns used by the model.
n_classes: Number of label classes. Defaults to 2, namely binary classification. Must be > 1.

EXERCISE 3 – Train and Evaluate the Model

Example – One piece of data

Labeled example

An example in which the feature and label are both defined. Used to train and evaluate the model.
In our case **refer Test_Dataset.csv, Training_Dataset.csv**. The csv contains features for blocks of text from receipts with the categories already defined



A	B	C	D	E	F
xLeftRel	yTopRel	isTotal	IsNumeric	category	
27	1	0	0	0	
28	5	0	0	1	
24	9	0	0	1	
34	14	0	0	1	

Training a model involves providing the model with training data to learn from. The training data must contain the correct answer. The learning algorithm finds patterns in the training data that map the input data attributes to the target. This is an iterative process.

In our case, **Training_Dataset.csv** contains the data for 10 sample receipts which have already been categorized.

TensorFlow API for training takes following parameters:

> **input_fn**: A function that provides input data for training as minibatches. The function should construct and return a tf.data.Dataset object
>
> **steps**: Number of steps for which to train the model

Evaluating a model is necessary to determine if it will do a good job of making predictions on new and future data. Because future instances have unknown label values, you need to check the accuracy of the ML model on data for which you already know the answer.

To properly evaluate a model, you should reserve some labeled sample data for the testing dataset. Once you have finished training the ML model, you run the model on the testing dataset. You then compare the predictions returned by the ML model against the known label values. Finally, you compute a summary metric that tells you how well the predicted and true values match.

In our case, **Test_Dataset.csv** contains the data for 5 sample receipts which have already been categorized and are not part of the Training_Dataset.csv

> **Do It Yourself:** Hyperparameters are the knobs that programmers tweak in machine learning algorithms. *Change the steps and batch size in out model and see the effects on accuracy. What are the optimal values?*

EXERCISE 4 - Predict using Model

Unlabeled example

An example where only the feature is defined, not the label. Once we've trained our model with labeled examples, we use that model to predict the label on unlabeled examples.

In our case, we will be running a prediction on Receipt.jpg. Receipt.csv contains the corresponding data with labels extracted by android app.

NOTE:

1. You don't have to train and evaluate your model every time you want to run a prediction. The Model is usually trained and evaluated at an earlier point, and the predictions are made real-time. We are training and evaluating the model for every prediction for the purposes of this workshop.
2. If you want to see detailed logs of the activities that happen on running your model, change the logging verbosity to DEBUG in your python script.

   *tf.logging.set_verbosity(tf.logging.**DEBUG**)*

Useful Links :

Git Repo – https://github.com/intuit-GHC-2018/Smart-Receipt
TensorFlow - https://www.tensorflow.org/tutorials/
TensorFlow DNNClassifier - https://www.tensorflow.org/api_docs/python/tf/estimator/DNNClassifier
Machine Learning Crash Course - https://developers.google.com/machine-learning/crash-course/
ML resources - https://www.kaggle.com/