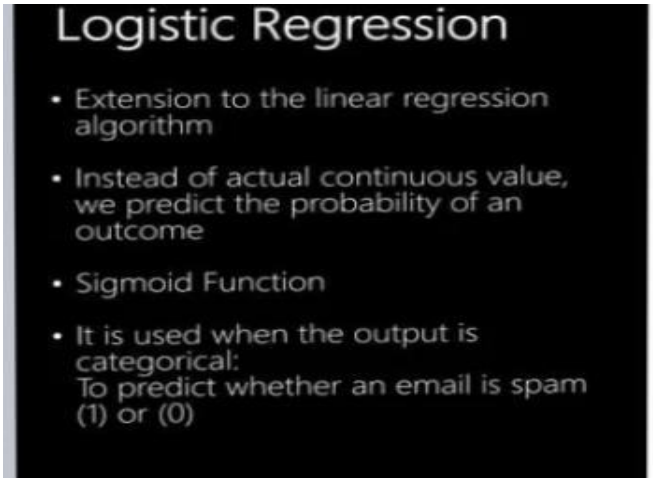
Build & Deploy Your Machine Learning Models Effortlessly

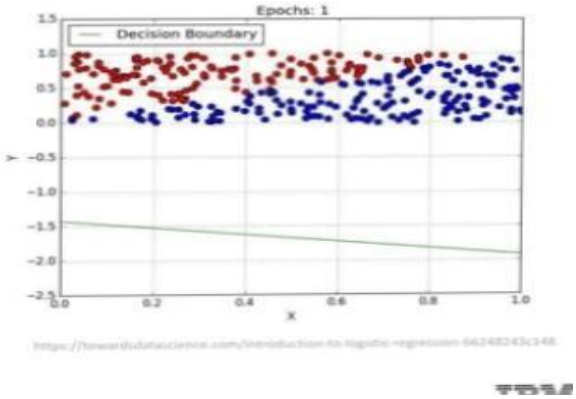
The goal was to show how everyone can easily use Jupyter Notebooks in IBM Watson Studio to run small pieces of code that process your data and immediately show you the results of your computation in an interactive environment and quickly build machine learning models.

It was divided into two sections. The first half of the workshop was explained to them about Data Science, Artificial Intelligence, Machine Learning, and Deep Learning.

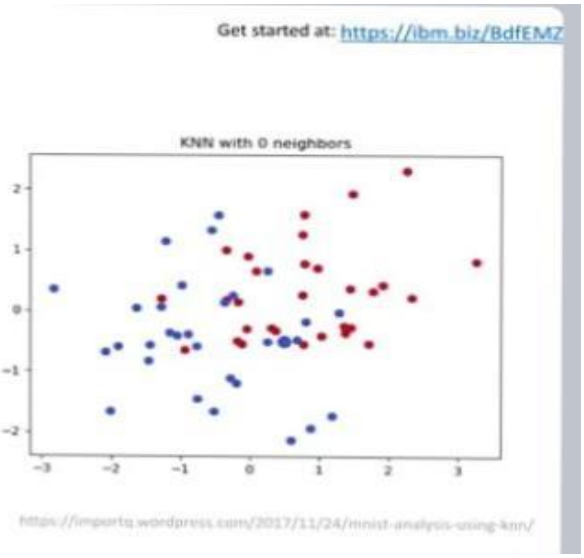
The classical machine learning algorithms, where she explained supervised learning and unsupervised learning. She then talked in detail about the Classification Algorithms like,

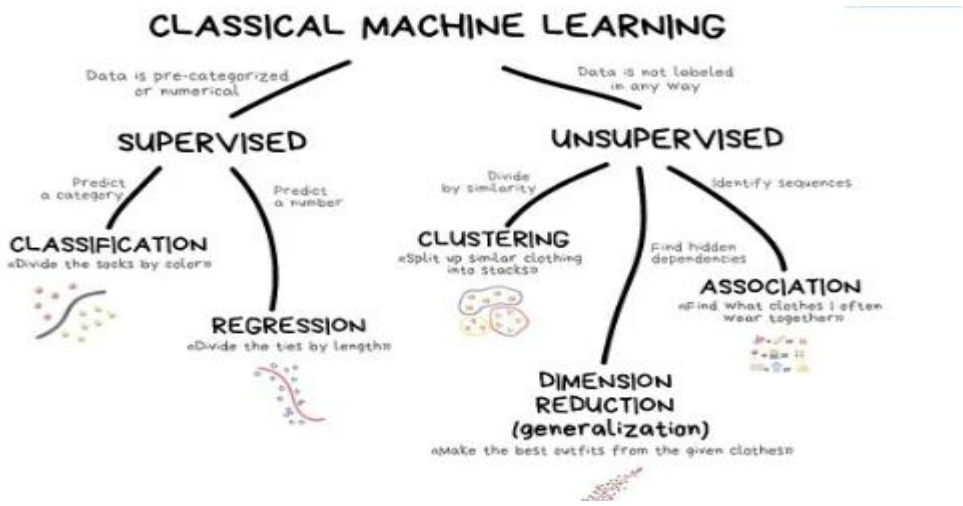
Logistic Regression





K-nearest neighbors(KNN)





The second half of this part consisted of a hands-on demo of how to build and deploy your machine learning models easily on Jupyter notebooks using Watson Studio on IBM Cloud.

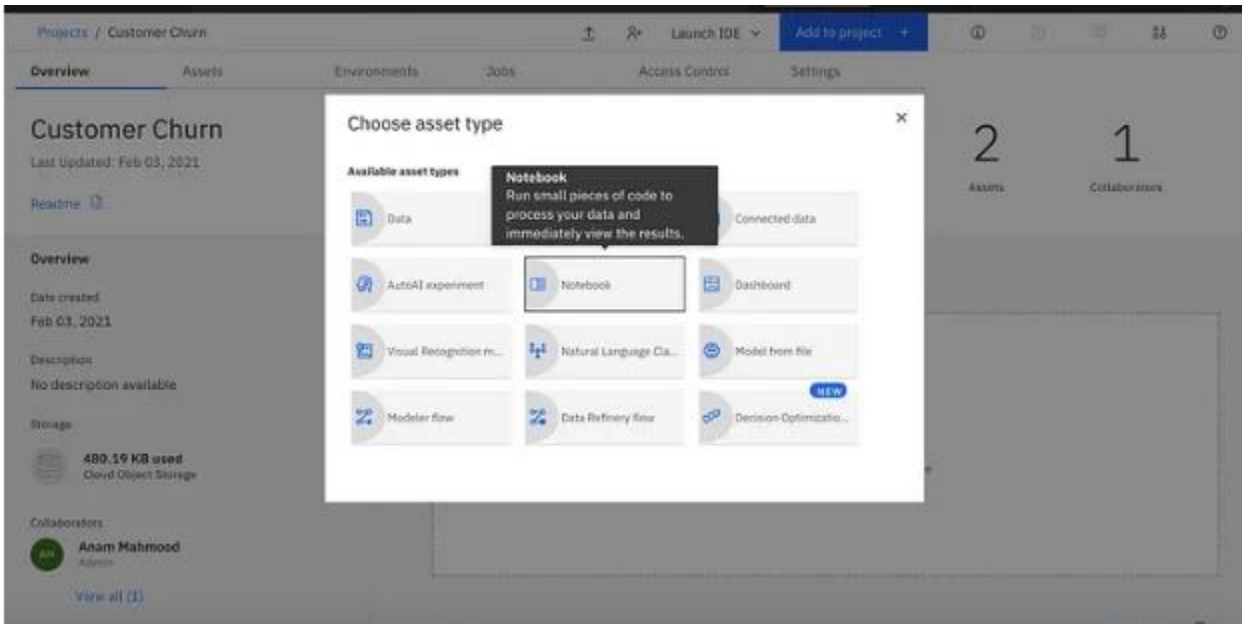
Churn refers to customers no longer using a company's services or products.

- Churn refers to customers no longer using a company's services or products.
- Churn analysis is the evaluation of a company's customer loss rate in order to reduce it.

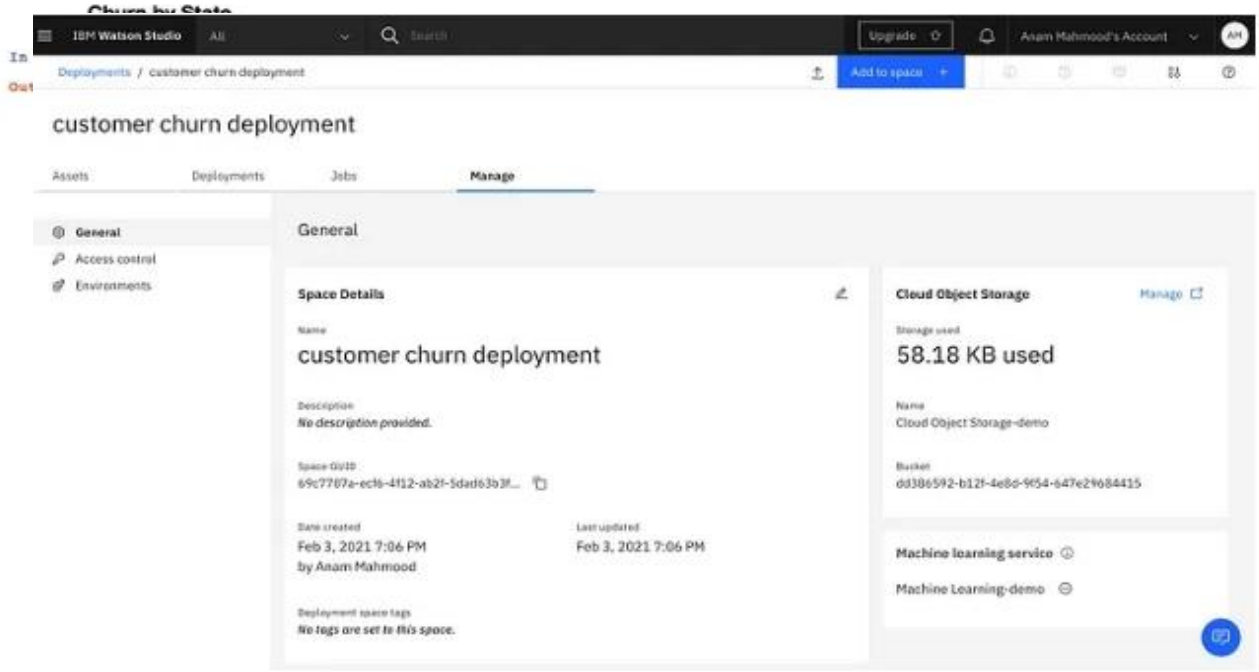
Then Showed them how to create services.

1. Create an IBM Cloud Object Storage service.
2. Create an IBM Watson Studio project.
3. Provision IBM Cloud services.
4. Upload the data set.
5. Create API Key

Then showed them how to create a deployment space and add a notebook from a URL. Once all was set, we then ran the notebook and imported the data as a data frame using pandas. She then walked them through the notebook, explaining to the audience the different algorithms and visualizations.



Add a notebook to your Watson Studio Project



In the end, we showed them how to deploy their notebook to IBM Cloud and test their model easily.

Getting the deployment space Space ID to deploy your notebook

```
# use our WML client to score our model
# add some test data
scoring_payload = {"input_data": [
    {'fields': ['state', 'account length', 'area code', 'international plan', 'voice mail plan', 'number vmail messages', 'total day minutes',
                'total day calls', 'total day charge', 'total eve minutes', 'total eve calls', 'total eve charge', 'total night minutes',
                'total night calls', 'total night charge', 'total intl minutes', 'total intl calls', 'total intl charge', 'customer service calls' ],
     'values': [[ '2','162','415','0','0','0','70.7','108','12.02','157.5','87','13.39','154.8','82','6.97','9.1','3','2.46','4' ]]
    }]}
```

```
# score the model
predictions = client.deployments.score(deployment_uid, scoring_payload)
print('prediction',json.dumps(predictions, indent=2))
```


```
prediction {
  "predictions": [
    {
      "fields": [
        "prediction",
        "probability"
      ],
      "values": [
        [
          1,
          [
            0.002986012165854701,
            0.9970139878341453
          ]
        ]
      ]
    }
```


Conclusion:

we have focused on how to prepare and use ML algorithms in Go. This included the preparation of data setting Up the Development Environment, and the use of data to build models.Supervised Learning, and Unsupervised Learning. We also looked at how to integrate an existing ML model into a Go application Using Pretrained Models. Finally, we covered how to integrate ML into production systems in Deploying Machine Learning Applications. To conclude, we will take a look at the different stages in a typical project, and how to manage the end-to-end process of developing and deploying a successful ML system.