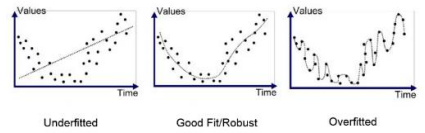

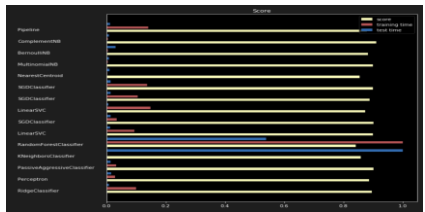


Analytics skills	Project highlights	Project description	Project
<b>Machine learning</b> <b>Train Convolutional Neural Network on dataset.</b> <b>Technology:</b> Python <b>Model training</b> Trained a simple Convolutional Neural Network on dataset.	<ul style="list-style-type: none"> <li>PyTorch on a single node</li> <li>Data processing</li> <li>Build a CNN model</li> <li>Training the model</li> </ul>	This notebook uses PyTorch on a single node. Plus, data processing on MNIST dataset, build a CNN model and train model at the end.	<u><a href="#">Handwritten Digit Recognition</a></u>  Handwritten Digit Recognition
<b>Machine learning</b> <b>Fit a neural network on the dataset.</b> <b>Technology:</b> Python <b>Model training</b> Using TensorFlow on a Spark driver node to fit neural network on the dataset.	<ul style="list-style-type: none"> <li>Install tensorflow</li> <li>Data processing</li> <li>Build a model.</li> <li>Define loss and optimizer.</li> <li>Monitor training progress by inline TensorBoard</li> <li>Train model in batches</li> <li>Test the model which trained.</li> </ul>	This notebook is about using TensorFlow on a Spark driver node for fit a neural network on MNIST to recognize handwritten digit on data.	<u><a href="#">Tensorflow on single node</a></u>  
<b>Machine learning</b> <b>Deep learning: end-to-end by using TensorFlow for house price prediction.</b> <b>Technology:</b> Python <b>Model training for prediction</b> Use TensorFlow Keras, Hyperopt, and MLflow to develop a deep learning model on the dataset for predict data.	<ul style="list-style-type: none"> <li>Data loading and preprocess</li> <li>Build a neural network model by TensorFlow Keras plus view training by inline TensorBoard</li> <li>Perform automated hyperparameter tuning with Hyperopt, MLflow and use autologging to save results.</li> <li>Use best hyperparameters set to create the final model.</li> <li>Register model in MLflow plus use the model for make predictions.</li> </ul>	Use TensorFlow Keras, Hyperopt, and MLflow to develop a deep learning model on Fetch_california_housing dataset for predict houses price in future.	<u><a href="#">Deep learning: end-to-end by using TensorFlow Keras, Hyperopt and MLflow</a></u>  
<b>Machine learning</b> <b>Classify documents by topics using a bag-of-words approach.</b> <b>Technology:</b> Python <b>Model training</b> Using scikit-learn to classify documents by topics, plus using a bag-of-words approach. Using scipy.sparse matrix for storing features and demonstrates various classifiers which efficiently handle sparse matrices.	<ul style="list-style-type: none"> <li>Load data</li> <li>split a training set and a test set.</li> <li>Mapping from integer feature name to original token string</li> <li>Benchmark classifiers</li> <li>Train SGD with Elastic Net penalty</li> <li>Train NearestCentroid without threshold</li> <li>Train sparse Naive Bayes classifiers</li> <li>Add plots</li> </ul>	This notebook using scikit-learn to classify documents by topics using a bag-of-words approach. With using a scipy.sparse matrix for storing the features and demonstrates various classifiers which efficiently handle sparse matrices.	<u><a href="#">Text documents classification by sparse features-plot</a></u>  

**Machine learning**  
**Pipeline for extract and evaluate text- Machine.**  
**Technology:** Python  
**Model training**  
Document classification in 20 categories.

- Illustrate progress logs on stdout
- Load couple of categories from training set
- Analysis categories
- Create pipeline combining the text feature extractor with the classifier.
- Try more parameters to give better exploring power.
- Find best parameters for the feature extraction and the classifier.

20 newsgroups dataset is automatically download, catch and reuse for document classification.  
Automatically get 20 categories or user can giving category name to dataset for adjusting number of them.

## Pipeline for extract and evaluate text- Machine

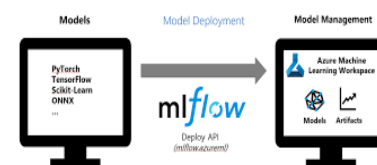


**Machine learning**  
**Random Forest model on a simple dataset plus MLflow Tracking API to log the model.**  
**Technology:** Python  
log MLflow runs to a workspace experiment.

- Create a Random Forest model on a simple dataset.
- Uses the MLflow Tracking API to log the model.
- Selected model parameters and metrics

Random Forest model on a simple dataset plus MLflow Tracking API to log the model.

## Track machine learning training runs Log runs to a notebook or workspace experiment

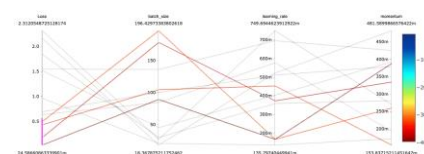


**Machine learning**  
**Models on Databricks.**  
**Technology:** Python  
**Model training**  
Train models (Hyperopt and SparkTrials), Parallel training with eachother, obtain the best.

- Train simple classification model by MLflow tracking
- Hyperparameter tuning to obtain the best performing model by Hyperopt

This notebook is an example of machine learning, for train models on Databricks. Which used scikit-learn libraries to be preinstalled on the Databricks Runtime for Machine Learning. Plus, using MLflow to track our trained models, moreover, use Hyperopt with SparkTrials for scale hyperparameter tuning.

## ML Hyperopt & SparkTrials: Model

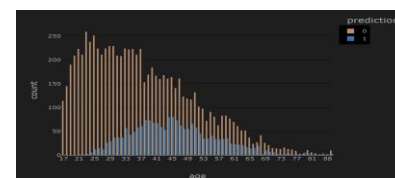


**Machine learning**  
**Is an individual's income > \$50,000?**  
**Technology:** Python  
**Model training for prediction**  
Using Apache Spark MLlib.  
Investigates the binary classification problem  
Build by some of the capabilities available in MLlib, tools for data preprocessing, machine learning pipelines, plus machine learning algorithms.

- Load the dataset.
- Feature preprocessing
- Define the model.
- Build the pipeline.
- Evaluate the model.
- Hyperparameter tuning
- Make predictions and evaluate model performance

This is an example work with Apache Spark MLlib. Which investigates the binary classification problem - to predict if an individual's income > \$50,000 based on given data.  
This project build by some of the capabilities available in MLlib, as well as tools for data preprocessing, machine learning pipelines, and several different machine learning algorithms.

## Predict if an individual's income > \$50,000



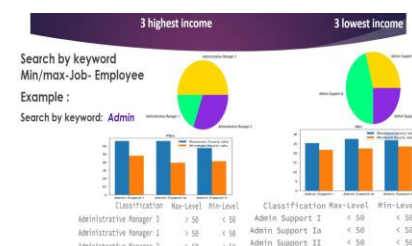
**Advance data analytic- Machine learning**  
**Recommend sorted job by user exception.**



**Technology:** Python

- Problem statement
- Data cleaning
- Exploratory analysis
- Feature engineering
- Methodology

Get user expectation (by asking question from user) and recommend some sorted job category with salary range and information.


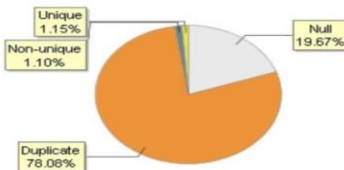
## **Job Analysis**



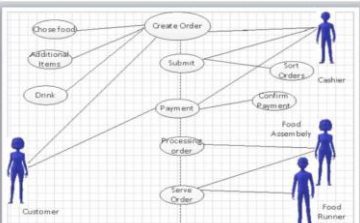
<p><b>Advance data analytic- Machine learning</b> visualize top 3 FCS game live by Plays, YPA and YPC for classified by city. <b>Technology:</b> Python Visualizing top 3 FCS game live from: <a href="http://api.collegefootballdata.com">api.collegefootballdata.com</a></p>	<ul style="list-style-type: none"> <li>• Load data from api</li> <li>• Data preprocess</li> <li>• Plays, YPA and YPC summarize by function</li> <li>• 3 Top FCS Games</li> <li>• Plot data</li> </ul>	<p>The goal of this project is visualizing top 3 FCS game live from: <a href="http://api.collegefootballdata.com">api.collegefootballdata.com</a> by Plays, YPA and YPC for classified by city.</p>	<p><b>3 Top FCS Games from 'College Football Data API'</b></p> 
<p><b>Power BI</b> <b>FIFA world cup history.</b> <b>Technology:</b> Power BI</p>	<ul style="list-style-type: none"> <li>• Contain some dashboards includes related chart from processed data to get maximum information from them.</li> </ul>	<p>This report containing a map to show countries and count of the matches played by heatmap for each. the table to show information by text and the line chart to compare countries in term of number of matches they had played on world cups from 1930 to 2014.</p>	<p><b>Introduction of FIFA world cup history</b></p> 

When I start play **FIFA 2020** I wish to have real match result for the teams I chose for the match. That would be pleasant if I had other players data and my history in game to have wide comparison between real data from real match and other players plus my history.

However, I start work on football data and making some sample works to help reach my dream 😊

<p><b>Power BI</b> <b>Analysis of California houses price.</b> <b>Technology:</b> Power BI</p>	<ul style="list-style-type: none"> <li>• Include some graphs on each dashboard, each has some related chart to filter data and get more insight from charts.</li> </ul>	<p>Analysis California houses price by house size and other factors.</p>	<p><b>PowerBI - California houses</b></p> 
<p><b>Data preparation</b> <b>Data quality improvement</b> <b>Technology:</b> Excel <b>Data quality assessment (DQA)</b> Initial assessment: SME review, research, suggestions by DQ rules.</p>	<p><b>Dataset anomalies:</b></p> <ul style="list-style-type: none"> <li>• Blank cells</li> <li>• Duplicates</li> <li>• Abbreviations</li> </ul> <p>Identify the potential anomalies to increase data quality for future processing.</p>	<p><b>Vancouver Business License dataset</b> Includes sort of relevant column. Initial screen of the dataset revealed some anomalies, that has been extensively analyzed. The graph shown dataset anomalies.</p>	<p><b>Greater Vancouver Business License</b></p> 

I love data preparation, that's my magic to understand data faster and prepare that for perform exploratory analysis and modeling.

<p><b>Business Analysis</b> <b>Food ordering process.</b> <b>Technology:</b> Microsoft Visio</p>	<p><b>ECS System Use-Cases</b></p> <ul style="list-style-type: none"> <li>• Use-Case Name</li> <li>• Actor</li> <li>• Trigger</li> <li>• Responses</li> </ul> <p><b>Check-out Equipment</b></p> <ul style="list-style-type: none"> <li>• Actor Action</li> <li>• System Response</li> </ul>	<p>Analysis of food ordering process.</p>	<p><b>Food order</b></p> 
--	---	---	--