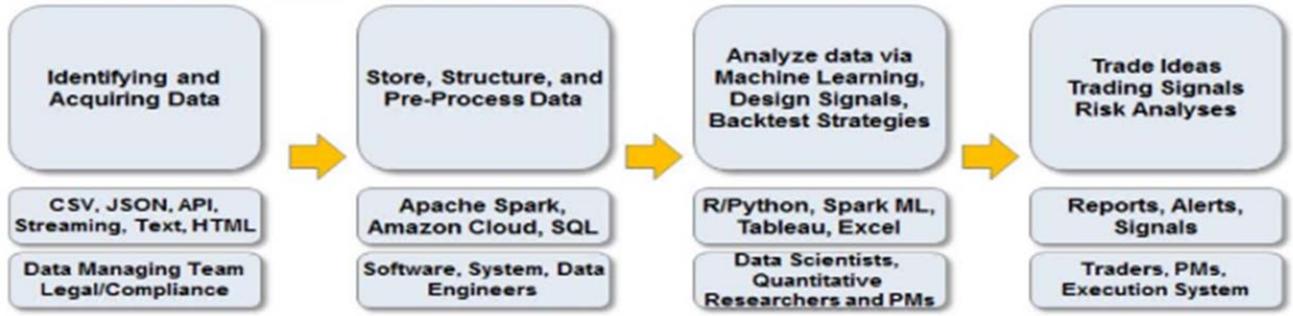
https://www.datasciencecentral.com/profiles/blogs/j-p-morgan-s-comprehensive-guide-on-machine-learning

J.P. Morgan's Comprehensive Guide on Machine Learning

- Posted by Vincent Granville on November 21, 2017 at 8:38pm
- View Blog

At 280 pages, the report is too long to cover in detail, but we've pulled out the most salient points for you below.


Identifying and Acquiring Data
CSV, JSON, API, Streaming, Text, HTML
Data Managing Team Legal/Compliance
Store, Structure, and Pre-Process Data
Apache Spark, Amazon Cloud, SQL
Software, System, Data Engineers
Analyze data via Machine Learning, Design Signals, Backtest Strategies
R/Python, Spark ML, Tableau, Excel
Data Scientists, Quantitative Researchers and PMs
Trade Ideas Trading Signals Risk Analyses
Reports, Alerts, Signals
Traders, PMs, Execution System

Main Points

- Banks will need to hire excellent data scientists who also understand how markets work
- Machines are best equipped to make trading decisions in the short and medium term
- An army of people will be needed to acquire, clean, and assess the data
- There are different kinds of machine learning. And they are used for different purposes
- Supervised learning will be used to make trend-based predictions using sample data
- Unsupervised learning will be used to identify relationships between a large number of variables
- Deep learning systems will undertake tasks that are hard for people to define but easy to perform
- Reinforcement learning will be used to choose a successive course of actions to maximize the final reward
- You won't need to be a machine learning expert, you will need to be an excellent quant and an excellent programmer
- These are the coding languages and data analysis packages you'll need to know
- And these are some examples of popular machine learning codes using Python
- Support functions are going to need to understand big data too

Read more here.

https://news.efinancialcareers.com/uk-en/285249/machine-learning-and-big-data-j-p-morgan

J.P.Morgan's massive guide to machine learning and big data jobs in finance

by Sarah Butcher 26 December 2017

Financial services jobs go in and out of fashion. In 2001 equity research for internet companies was all the rage. In 2006, structuring collateralised debt obligations (CDOs) was the thing. In 2010, credit traders were popular. In 2014, compliance professionals were it. In 2017, it was all about machine learning and big data.

In May, J.P. Morgan's quantitative investing and derivatives strategy team, led Marko Kolanovic and Rajesh T. Krishnamachari, issued the most comprehensive report *ever* on big data and machine learning in financial services.

Titled, 'Big Data and AI Strategies' and subheaded, 'Machine Learning and Alternative Data Approach to Investing', the report says that machine learning will become crucial to the future functioning of markets. Analysts, portfolio managers, traders and chief investment officers all need to become familiar with machine learning techniques. If they don't they'll be left behind: traditional data sources like quarterly earnings and GDP figures will become increasingly irrelevant as managers using newer datasets and methods will be able to predict them in advance and to trade ahead of their release.

At 280 pages, the report is too long to cover in detail, but we've pulled out the most salient points for you below.

1. Banks will need to hire excellent data scientists who also understand how markets work

J.P. Morgan cautions against the fashion for banks and finance firms to prioritize data analysis skills over market knowledge. Doing so is dangerous. Understanding the economics behind the data and the signals is more important than developing complex technical solutions.

2. Machines are best equipped to make trading decisions in the short and medium term

J.P. Morgan notes that human beings are already all but excluded from high frequency trading. In future, they say machines will become increasingly prevalent over the medium term too: "Machines have the ability to quickly analyze news feeds and tweets, process earnings statements, scrape websites, and trade on these instantaneously." This will help erode demand for fundamental analysts, equity long-short managers and macro investors.

In the long term, however, humans will retain an advantage: "Machines will likely not do well in assessing regime changes (market turning points) and forecasts which involve interpreting more complicated human responses such as those of politicians and central bankers, understanding client positioning, or anticipating crowding," says J.P. Morgan. If you want to survive as a human investor, this is where you will need to make your niche,

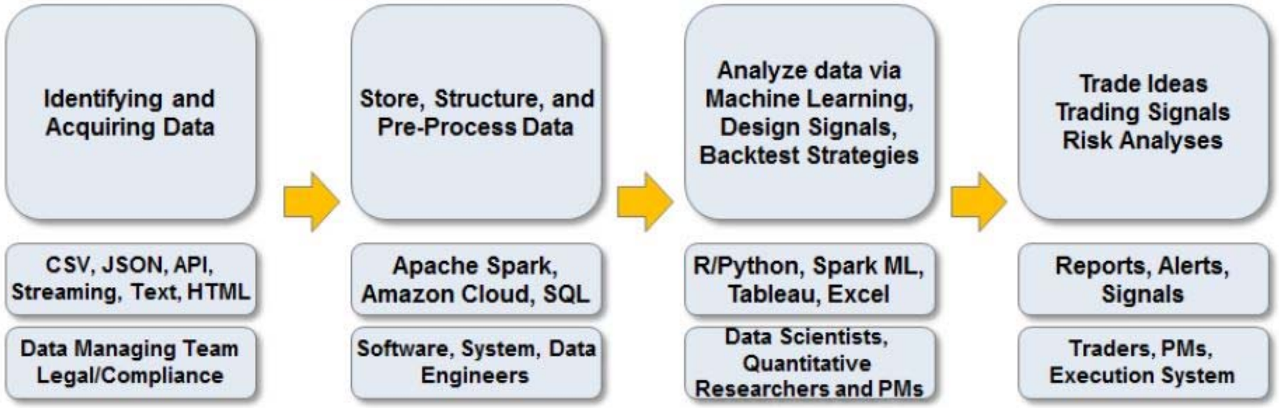
4. An army of people will be needed to acquire, clean, and assess the data

Before machine learning strategies can be implemented, data scientists and quantitative researchers need to acquire and analyze the data with the aim of deriving tradable signals and insights.

J.P. Morgan notes that data analysis is complex. Today's datasets are often bigger than yesterday's. They can include anything from data generated by individuals (social media posts, product reviews, search trends, etc.), to data generated by business processes (company exhaust data, commercial transaction, credit card data, etc.) and data generated by sensors (satellite image data, foot and car traffic, ship locations, etc.). These new forms of data need to be analyzed before they can be used in a trading strategy. They also need to be assessed for 'alpha content' - their ability to generate alpha.

Alpha content will be partially dependent upon the cost of the data, the amount of processing required and how well-used the dataset is already.

Figure 9: Big Data workflow for investment managers


Identifying and Acquiring Data
Store, Structure, and Pre-Process Data
Analyze data via Machine Learning, Design Signals, Backtest Strategies
Trade Ideas Trading Signals Risk Analyses
CSV, JSON, API, Streaming, Text, HTML
Apache Spark, Amazon Cloud, SQL
R/Python, Spark ML, Tableau, Excel
Reports, Alerts, Signals
Data Managing Team Legal/Compliance
Software, System, Data Engineers
Data Scientists, Quantitative Researchers and PMs
Traders, PMs, Execution System

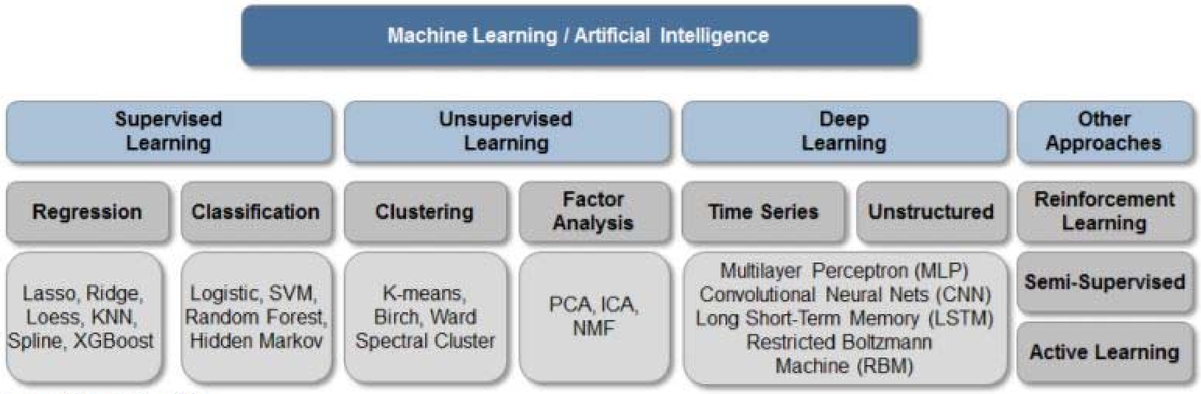
Source: J.P.Morgan Macro QDS

5. There are different kinds of machine learning. And they are used for different purposes

Machine learning has various iterations, including supervised learning, unsupervised learning and deep and reinforcement learning.

The purpose of supervised learning is to establish a relationship between two datasets and to use one dataset to forecast the other. The purpose of unsupervised learning is to try to understand the structure of data and to identify the main drivers behind it. The purpose of deep learning is to use multi-layered neural networks to analyze a trend, while reinforcement learning encourages algorithms to explore and find the most profitable trading strategies.

Figure 39: Classification of Machine Learning techniques


Machine Learning / Artificial Intelligence
Supervised Learning
Unsupervised Learning
Deep Learning
Other Approaches
Regression
Classification
Clustering
Factor Analysis
Time Series
Unstructured
Reinforcement Learning
Lasso, Ridge, Loess, KNN, Spline, XGBoost
Logistic, SVM, Random Forest, Hidden Markov
K-means, Birch, Ward Spectral Cluster
PCA, ICA, NMF
Multilayer Perceptron (MLP) Convolutional Neural Nets (CNN) Long Short-Term Memory (LSTM) Restricted Boltzmann Machine (RBM)
Semi-Supervised
Active Learning

Source: J.P.Morgan Macro QDS

6. Supervised learning will be used to make trend-based predictions using sample data

In a finance context, J.P. Morgan says supervised learning algorithms are provided with provided historical data and asked to find the relationship that has the best predictive power. Supervised learning algorithms come in two varieties: regression and classification methods.

Regression-based supervised learning methods try to predict outputs based on input variables. For example, they might look at how the market will move if inflation spikes.

Classification methods work backwards and try to identify which category a set of classifications belong to.

7. Unsupervised learning will be used to identify relationships between a large number of variables

In unsupervised learning, a machine is given an entire set of returns from assets and doesn't know which are the dependent and the independent variables. At a high level, unsupervised learning methods are categorized as clustering or factor analyses.

Clustering involves splitting a dataset into smaller groups based on some notion of similarity. For example, it cant involve identifying historical regimes with high and low volatility, rising and failing rates, or rising and falling inflation.

Factor analyses aim to identify the main drivers of the data or to identify best representation of the data. For example, yield curve movements can be described by the parallel shift of yields, steepening of the curve, and convexity of the curve. In a multi-asset portfolio, factor analysis will identify the main drivers such as momentum, value, carry, volatility, or liquidity.

8. Deep learning systems will undertake tasks that are hard for people to define but easy to perform

Deep learning is effectively an attempt to artificially recreate human intelligence. J.P. Morgan says deep learning is particularly well suited to the pre-processing of unstructured big data sets (for instance, it can be used to count cars in satellite images, or to identify sentiment in a press release.). A deep learning model could use a hypothetical financial data series to estimate the probability of a market correction.

Deep Learning methods are based on neural networks which are loosely inspired by the workings of the human brain. In a network, each neuron receives inputs from other neurons, and ‘computes’ a weighted average of these inputs. The relative weighting of different inputs is guided by the past experience.

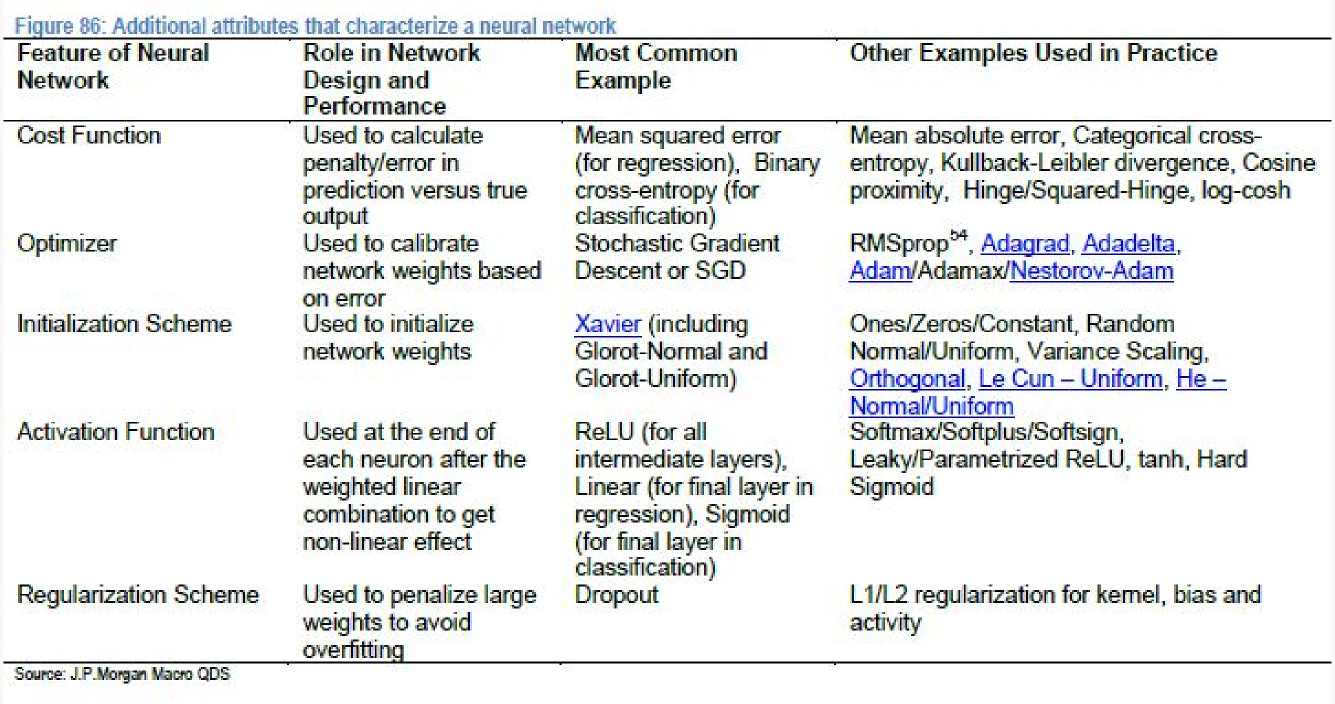


Figure 86: Additional attributes that characterize a neural network

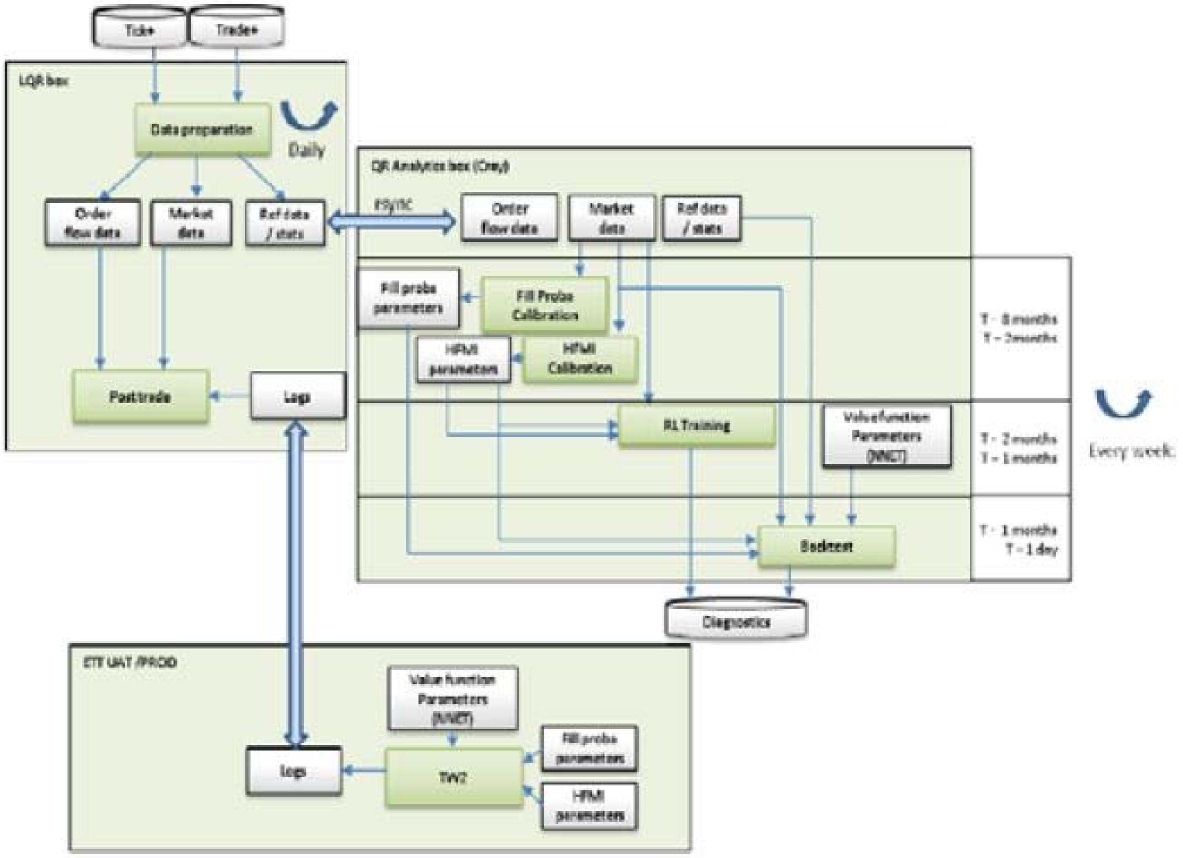
Feature of Neural Network	Role in Network Design and Performance	Most Common Example	Other Examples Used in Practice
Cost Function	Used to calculate penalty/error in prediction versus true output	Mean squared error (for regression), Binary cross-entropy (for classification)	Mean absolute error, Categorical cross-entropy, Kullback-Leibler divergence, Cosine proximity, Hinge/Squared-Hinge, log-cosh
Optimizer	Used to calibrate network weights based on error	Stochastic Gradient Descent or SGD	RMSprop[54], Adagrad, Adadelta, Adam/Adamax/Nestorov-Adam
Initialization Scheme	Used to initialize network weights	Xavier (including Glorot-Normal and Glorot-Uniform)	Ones/Zeros/Constant, Random Normal/Uniform, Variance Scaling, Orthogonal, Le Cun – Uniform, He – Normal/Uniform
Activation Function	Used at the end of each neuron after the weighted linear combination to get non-linear effect	ReLU (for all intermediate layers), Linear (for final layer in regression), Sigmoid (for final layer in classification)	Softmax/Softplus/Softsign, Leaky/Parametrized ReLU, tanh, Hard Sigmoid
Regularization Scheme	Used to penalize large weights to avoid overfitting	Dropout	L1/L2 regularization for kernel, bias and activity

Source: J.P.Morgan Macro QDS

9. Reinforcement learning will be used to choose a successive course of actions to maximize the final reward

The goal of reinforcement learning is to choose a course of successive actions in order to maximize the final (or cumulative) reward. Unlike supervised learning (which is typically a one step process), the reinforcement learning model doesn’t know the correct action at each step.

J.P. Morgan's electronic trading group has already developed algorithms using reinforcement learning. The diagram below shows the bank's machine learning model (we suspect it's blurry on purpose).



Source: JPM Linear Quantitative Research Team.

10. You *won't* need to be a machine learning expert, you *will* need to be an excellent quant and an excellent programmer

J.P. Morgan says the skillset for the role of data scientists is virtually the same as for any other quantitative researchers. Existing buy side and sell side quants with backgrounds in computer science, statistics, maths, financial engineering, econometrics and natural sciences *should* therefore be able to reinvent themselves. Expertise in quantitative trading strategies will be the crucial skill. "It is much easier for a quant researcher to change the format/size of a dataset, and employ better statistical and Machine Learning tools, than for an IT expert, silicon valley entrepreneur, or academic to learn how to design a viable trading strategy," say Kolanovic and Krishnamacharc.

By comparison, J.P. Morgan notes that you *won't* need to know about machine learning in any great detail. - Most of the Machine Learning methods are already coded (e.g. in R): you just need to apply the existing models. As a start, they suggest you can look at small datasets using GUI-based software like Weka. Python also has extensive libraries like Keras (keras.io). And there are open source Machine Learning libraries like Tensorflow and Theano.

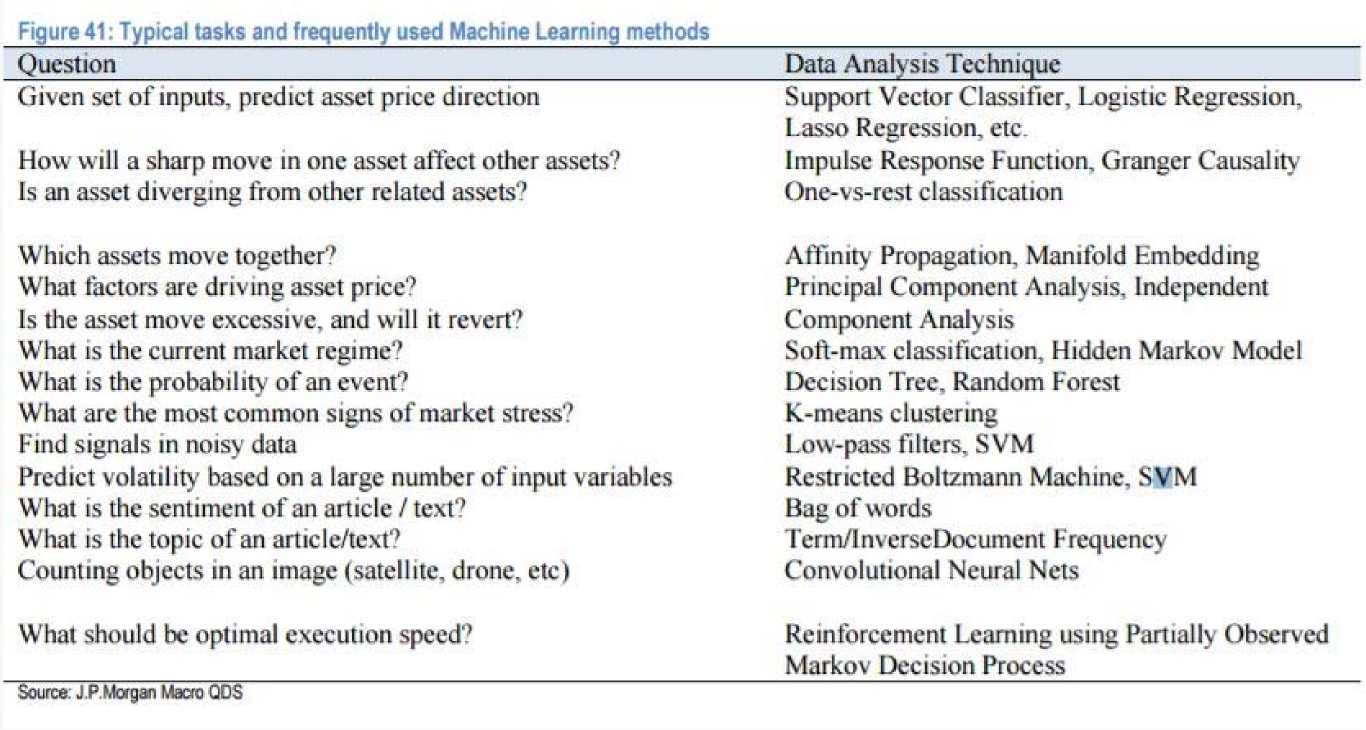


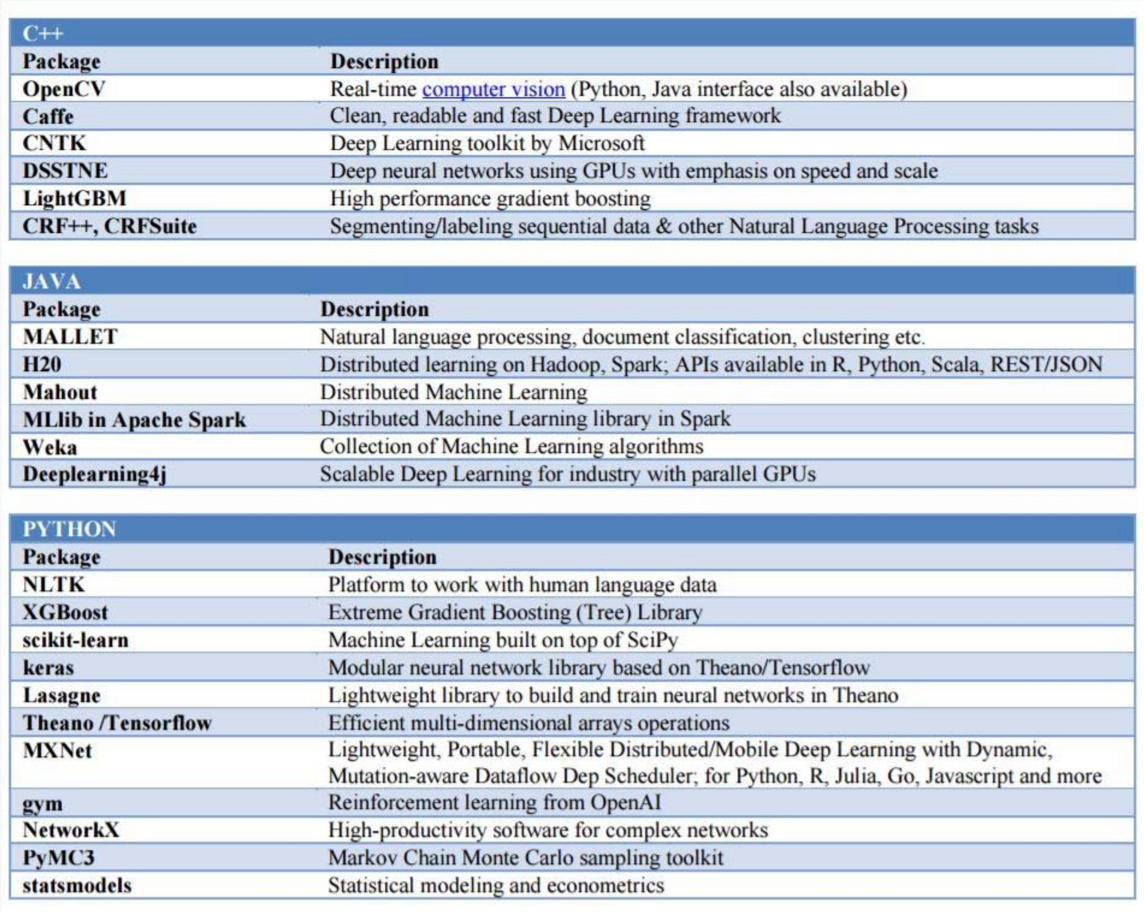
Figure 41: Typical tasks and frequently used Machine Learning methods

Question	Data Analysis Technique
Given set of inputs, predict asset price direction	Support Vector Classifier, Logistic Regression, Lasso Regression, etc.
How will a sharp move in one asset affect other assets?	Impulse Response Function, Granger Causality
Is an asset diverging from other related assets?	One-vs-rest classification
Which assets move together?	Affinity Propagation, Manifold Embedding
What factors are driving asset price?	Principal Component Analysis, Independent
Is the asset move excessive, and will it revert?	Component Analysis
What is the current market regime?	Soft-max classification, Hidden Markov Model
What is the probability of an event?	Decision Tree, Random Forest
What are the most common signs of market stress?	K-means clustering
Find signals in noisy data	Low-pass filters, SVM
Predict volatility based on a large number of input variables	Restricted Boltzmann Machine, SVM
What is the sentiment of an article / text?	Bag of words
What is the topic of an article/text?	Term/InverseDocument Frequency
Counting objects in an image (satellite, drone, etc)	Convolutional Neural Nets
What should be optimal execution speed?	Reinforcement Learning using Partially Observed Markov Decision Process

Source: J.P.Morgan Macro QDS

11. These are the coding languages and data analysis packages you'll need to know

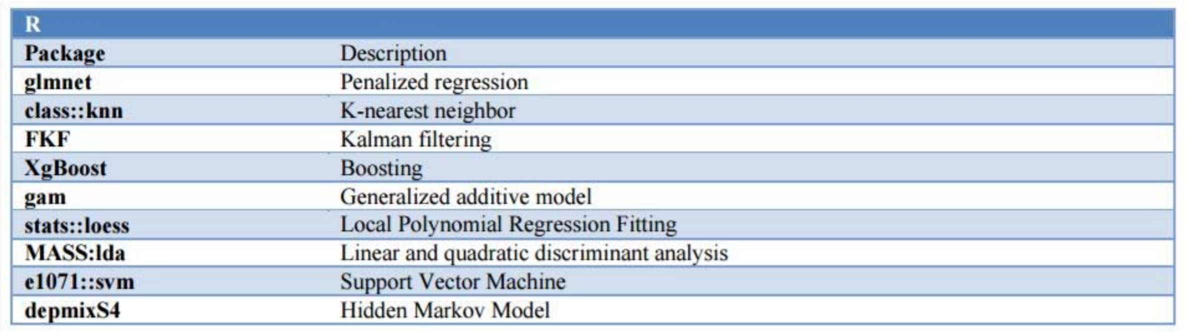
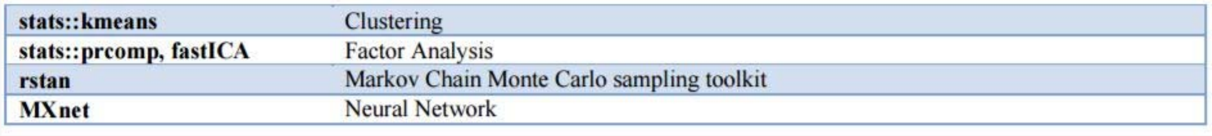
If you're only planning to learn one coding language related to machine learning, J.P. Morgan suggests you choose R, along with the related packages below. However, C++, Python and Java also have machine learning applications as shown below.



C++	
Package	**Description**
OpenCV	Real-time computer vision (Python, Java interface also available)
Caffe	Clean, readable and fast Deep Learning framework
CNTK	Deep Learning toolkit by Microsoft
DSSTNE	Deep neural networks using GPUs with emphasis on speed and scale
LightGBM	High performance gradient boosting
CRF++, CRFSuite	Segmenting/labeling sequential data & other Natural Language Processing tasks

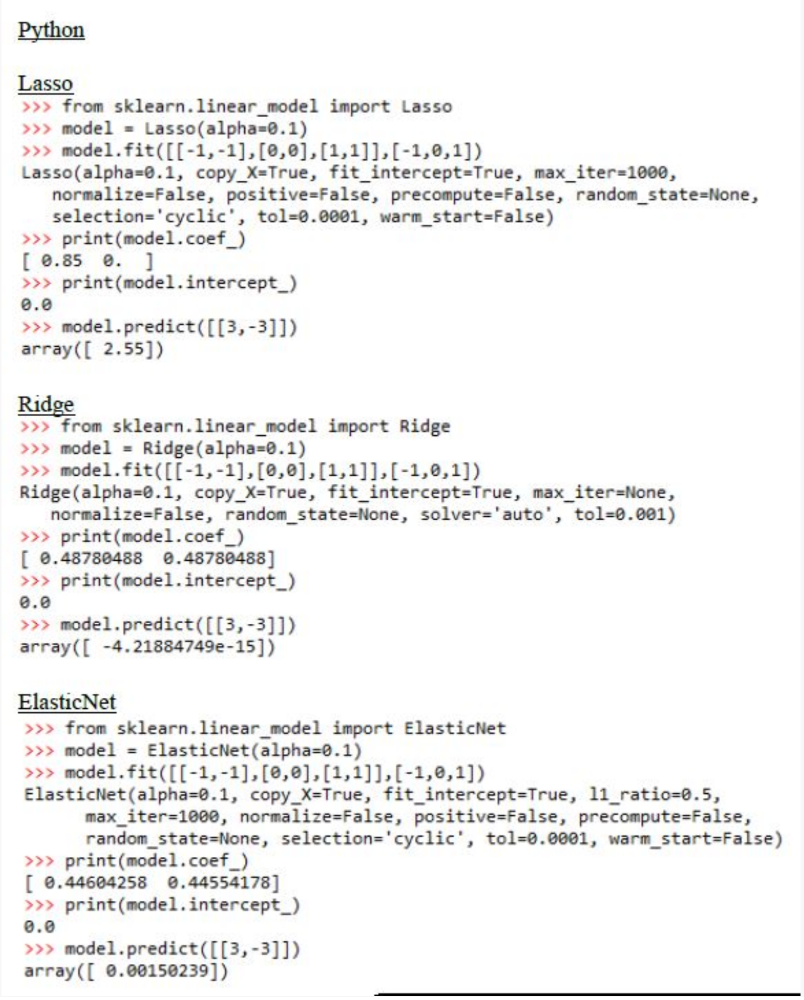
JAVA	
Package	**Description**
MALLET	Natural language processing, document classification, clustering etc.
H20	Distributed learning on Hadoop, Spark; APIs available in R, Python, Scala, REST/JSON
Mahout	Distributed Machine Learning
MLlib in Apache Spark	Distributed Machine Learning library in Spark
Weka	Collection of Machine Learning algorithms
Deeplearning4j	Scalable Deep Learning for industry with parallel GPUs

PYTHON	
Package	**Description**
NLTK	Platform to work with human language data
XGBoost	Extreme Gradient Boosting (Tree) Library
scikit-learn	Machine Learning built on top of SciPy
keras	Modular neural network library based on Theano/Tensorflow
Lasagne	Lightweight library to build and train neural networks in Theano
Theano /Tensorflow	Efficient multi-dimensional arrays operations
MXNet	Lightweight, Portable, Flexible Distributed/Mobile Deep Learning with Dynamic, Mutation-aware Dataflow Dep Scheduler; for Python, R, Julia, Go, Javascript and more
gym	Reinforcement learning from OpenAI
NetworkX	High-productivity software for complex networks
PyMC3	Markov Chain Monte Carlo sampling toolkit
statsmodels	Statistical modeling and econometrics

R	
Package	Description
glmnet	Penalized regression
class::knn	K-nearest neighbor
FKF	Kalman filtering
XgBoost	Boosting
gam	Generalized additive model
stats::loess	Local Polynomial Regression Fitting
MASS:lda	Linear and quadratic discriminant analysis
e1071::svm	Support Vector Machine
depmixS4	Hidden Markov Model
stats::kmeans	Clustering
stats::prcomp, fastICA	Factor Analysis
rstan	Markov Chain Monte Carlo sampling toolkit
MXnet	Neural Network

12. And these are some examples of popular machine learning codes using Python



Python

Lasso

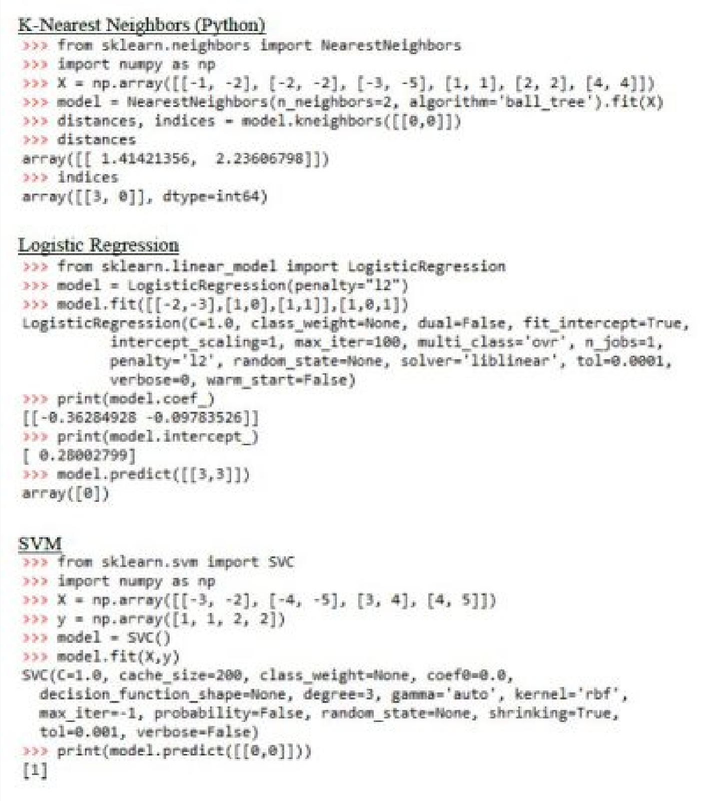
```
>>> from sklearn.linear_model import Lasso
>>> model = Lasso(alpha=0.1)
>>> model.fit([[-1,-1],[0,0],[1,1]],[-1,0,1])
Lasso(alpha=0.1, copy_X=True, fit_intercept=True, max_iter=1000,
   normalize=False, positive=False, precompute=False, random_state=None,
   selection='cyclic', tol=0.0001, warm_start=False)
>>> print(model.coef_)
[ 0.85  0.  ]
>>> print(model.intercept_)
0.0
>>> model.predict([[3,-3]])
array([ 2.55])
```

Ridge

```
>>> from sklearn.linear_model import Ridge
>>> model = Ridge(alpha=0.1)
>>> model.fit([[-1,-1],[0,0],[1,1]],[-1,0,1])
Ridge(alpha=0.1, copy_X=True, fit_intercept=True, max_iter=None,
   normalize=False, random_state=None, solver='auto', tol=0.001)
>>> print(model.coef_)
[ 0.48780488  0.48780488]
>>> print(model.intercept_)
0.0
>>> model.predict([[3,-3]])
array([ -4.21884749e-15])
```

ElasticNet

```
>>> from sklearn.linear_model import ElasticNet
>>> model = ElasticNet(alpha=0.1)
>>> model.fit([[-1,-1],[0,0],[1,1]],[-1,0,1])
ElasticNet(alpha=0.1, copy_X=True, fit_intercept=True, l1_ratio=0.5,
      max_iter=1000, normalize=False, positive=False, precompute=False,
      random_state=None, selection='cyclic', tol=0.0001, warm_start=False)
>>> print(model.coef_)
[ 0.44604258  0.44554178]
>>> print(model.intercept_)
0.0
>>> model.predict([[3,-3]])
array([ 0.00150239])
```



K-Nearest Neighbors (Python)

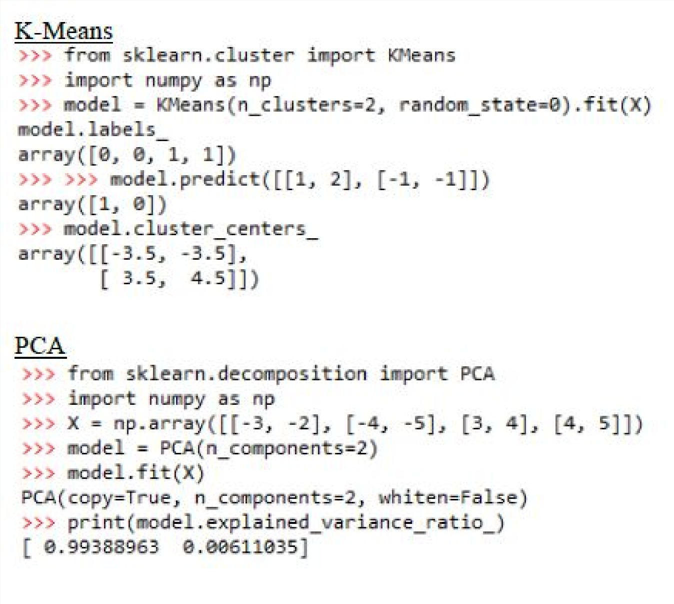
```
>>> from sklearn.neighbors import NearestNeighbors
>>> import numpy as np
>>> X = np.array([[-1, -2], [-2, -2], [-3, -5], [1, 1], [2, 2], [4, 4]])
>>> model = NearestNeighbors(n_neighbors=2, algorithm='ball_tree').fit(X)
>>> distances, indices = model.kneighbors([[0,0]])
>>> distances
array([[ 1.41421356,  2.23606798]])
>>> indices
array([[3, 0]], dtype=int64)
```

Logistic Regression

```
>>> from sklearn.linear_model import LogisticRegression
>>> model = LogisticRegression(penalty="l2")
>>> model.fit([[-2,-3],[1,0],[1,1]],[1,0,1])
LogisticRegression(C=1.0, class_weight=None, dual=False, fit_intercept=True,
          intercept_scaling=1, max_iter=100, multi_class='ovr', n_jobs=1,
          penalty='l2', random_state=None, solver='liblinear', tol=0.0001,
          verbose=0, warm_start=False)
>>> print(model.coef_)
[[-0.36284928 -0.09783526]]
>>> print(model.intercept_)
[ 0.28002799]
>>> model.predict([[3,3]])
array([0])
```

SVM

```
>>> from sklearn.svm import SVC
>>> import numpy as np
>>> X = np.array([[-3, -2], [-4, -5], [3, 4], [4, 5]])
>>> y = np.array([1, 1, 2, 2])
>>> model = SVC()
>>> model.fit(X,y)
SVC(C=1.0, cache_size=200, class_weight=None, coef0=0.0,
  decision_function_shape=None, degree=3, gamma='auto', kernel='rbf',
  max_iter=-1, probability=False, random_state=None, shrinking=True,
  tol=0.001, verbose=False)
>>> print(model.predict([[0,0]]))
[1]
```



K-Means

```
>>> from sklearn.cluster import KMeans
>>> import numpy as np
>>> model = KMeans(n_clusters=2, random_state=0).fit(X)
model.labels_
array([0, 0, 1, 1])
>>> >>> model.predict([[1, 2], [-1, -1]])
array([1, 0])
>>> model.cluster_centers_
array([[-3.5, -3.5],
       [ 3.5,  4.5]])
```

PCA

```
>>> from sklearn.decomposition import PCA
>>> import numpy as np
>>> X = np.array([[-3, -2], [-4, -5], [3, 4], [4, 5]])
>>> model = PCA(n_components=2)
>>> model.fit(X)
PCA(copy=True, n_components=2, whiten=False)
>>> print(model.explained_variance_ratio_)
[ 0.99388963  0.00611035]
```

13. Support functions are going to need to understand big data too

Lastly, J.P. Morgan notes that support functions need to know about big data too. The report says that too many recruiters and hiring managers are incapable of distinguishing between an ability to talk broadly about artificial intelligence and an ability to actually design a tradeable strategy At the same time, compliance teams will need to be able to vet machine learning models and to ensure that data is properly anonymized and doesn't contain private information. The age of machine learning in finance is upon us.