**Python Machine Learning Projects**

**Section 1: Build an App to Find Cheap Airfares**

**1.1 The Course Overview**

This video gives an overview of the entire course.

**1.2 Sourcing Airfare Pricing Data**

We need the air pricing data from a website to work with. You will learn to do that in this section.

* Get flight explorer and look at an example

**1.3 Retrieving the Fare Data with Advanced Web Scraping Techniques**

After determining the source of the data, we need to retrieve the data.

* Install Phantom JS and the necessary libraries
* Instantiate the browser object
* Send the user agent to the receiving server

**1.4 Parsing the DOM to Extract Pricing Data**

DOM is the structure of elements that form the web page. We need to get some details of the structure by parsing it.

* Feed the page source and retrieve a list of best prices
* Extract the best price or the cheapest price
* Identify outliers with clustering techniques

**1.5 Sending Real-Time Alerts Using IFTTT**

To get real-time alerts when a particular event occurs, we need to use IFTTT.

* Sign up for the Maker channel
* Create an event fare alert
* Fill in the message and customize it

**1.6 Putting It All Together**

To deploy our app, we'll move on to working in a text editor. You will put together the entire code to get the final result.

* Import packages
* Create a function that pulls down the data and runs your clustering algorithm
* Include a scheduler and run the file from the command line

**Section 2: Forecast the IPO Market Using Logistic Regression**

**2.1 The IPO Market**

Before deciding strategies for the IPO market, we need to study the IPO market and derive inferences from it.

* Read about the IPO market
* Look at the performance of the IPO market
* Study strategies

**2.2 Feature Engineering**

The consideration and inclusion of all factors affecting the market is called feature engineering. Modeling this is as important as the data used in building the model.

* Add features. Retrieve data.
* Tidy up the underwriter data. Add final features.
* Transform data into matrix form

**2.3 Binary Classification**

Instead of giving the value of the return, you can predict the IPO for a trade you will buy or not buy. The model used is logistic regression.

* Apply logic regression to the data
* Split data into training and testing datasets. Fit the model.
* Evaluate the model

**2.4 Feature Importance**

It is important to know which features will make the offering successful. You can find that out in this section.

* Examine coefficients for logistic regression.
* Fit random forest classifier.
* Evaluate summary

**Section 3: Create a Custom Newsfeed**

**3.1 Creating a Supervised Training Set with the Pocket App**

To create a model, we have to first have a training dataset. We will use the pocket app for this.

* Install the pocket chrome extension.
* Use the pocket API to retrieve stories.

**3.2 Using the embed.ly API to Download Story Bodies**

You can't move forward with just the URLs of the stories. You would need the full article. So let's check out how to do that in this video.

* Sign up for embed.ly API access.
* Feed plain text to the model.

**3.3 Natural Language Processing Basics**

Machine learning models work on numerical data. So we will need to transform our text into numerical data using NLP.

* Convert the corpus into a BOW representation. Remove stop words.
* Use the tf-idf algorithm. Convert the training set into a tf-idf matrix.

**3.4 Support Vector Machines**

You will learn about the linear support vector machine in this video. The SVM algorithm separates data points linearly into classes.

* Feed the tf-idf matrix into the SVM.

**3.5 IFTTT Integration with Feeds, Google Sheets, and E-mail**

We have provided a training dataset. But we also need a stream of articles as a testing dataset to run our model against.

* Set up news feeds and Google sheets.
* Pull down articles using a Python library.
* Make changes if necessary and rebuild the model.

**3.6 Setting Up Your Daily Personal Newsletter**

It would make life easier if you get a personalized e-mail of your stories, right? So you will learn how to do that in this video.

* Create a recipe. Receive a web request and create a trigger.
* Generate a script that will send us articles daily.

**Section 4: Forecasting the Stock Market with Machine Learning**

**4.1 What Does Research Tell Us about the Stock Market?**

Research is the most important thing before we start working on designing a strategy.

* Study the market and understand it.
* Understand different forms of the market.

**4.2 Developing a Trading Strategy**

Once you have studied the various aspects of the market, it is time to develop a trading strategy. You will learn it in this video.

* Read and plot data.
* Pull data for various statistical values.
* Extend the time span.

**4.3 Building a Model and Evaluating Its Performance**

Now that we have our baseline, we will build our first regression model for prediction of stocks.

* Set up a dataframe. Import SVM and set training and testing datasets.
* Fit the model. Compare with the actual data.
* Evaluate the performance by adjusting the different parameters till the desired result is met.

**4.4 Modeling with Dynamic Time Warping**

Another algorithm to work with is dynamic time warping. It provides us a metric which will inform us about the similarity between two time series.

* Calculate the distance between two time series.
* Compare a series against all other series and infer.
* Evaluate trades.

**Section 5: Build an Image Similarity Engine**

**5.1 Machine Learning on Images**

It is very important to understand machine learning's concepts before working with it.

* Understand basic machine learning
* Look at similar technologies by Google and Facebook.

**5.2 Working with Images**

In order to work with images, we need to transform them into a matrix form, that is, numerical form.

* Load the MNIST database
* Get the matrix. Perform operations on it.

**5.3 Finding Similar Images**

We will use algorithms to find similar images in the database.

* In cosine similarity algorithm, compute the similarity
* Take a look at the results.
* Test with chi-squared algorithm

**5.4 Building an Image Similarity Engine**

We will combine what we have studied so far to build an image similarity engine.

* Utilize GraphLab Create, and load images
* Extract deep features and compare them.
* Show the image

**Section 6: Building a Chatbot**

**6.1 The Design of Chatbots**

Design of chatbots consists of parameters like mode of communication, the content, and so on. You will look at that in this video.

* Observe Python NLTK
* Run a few sample questions and response samples

**6.2 Building a Chatbot**

Having looked at the working of a chatbot, we will now build a chatbot.

* Get training dataset. Load data
* Parse the data into a question-answer form
* Get similarity scores. Test the chatbot.