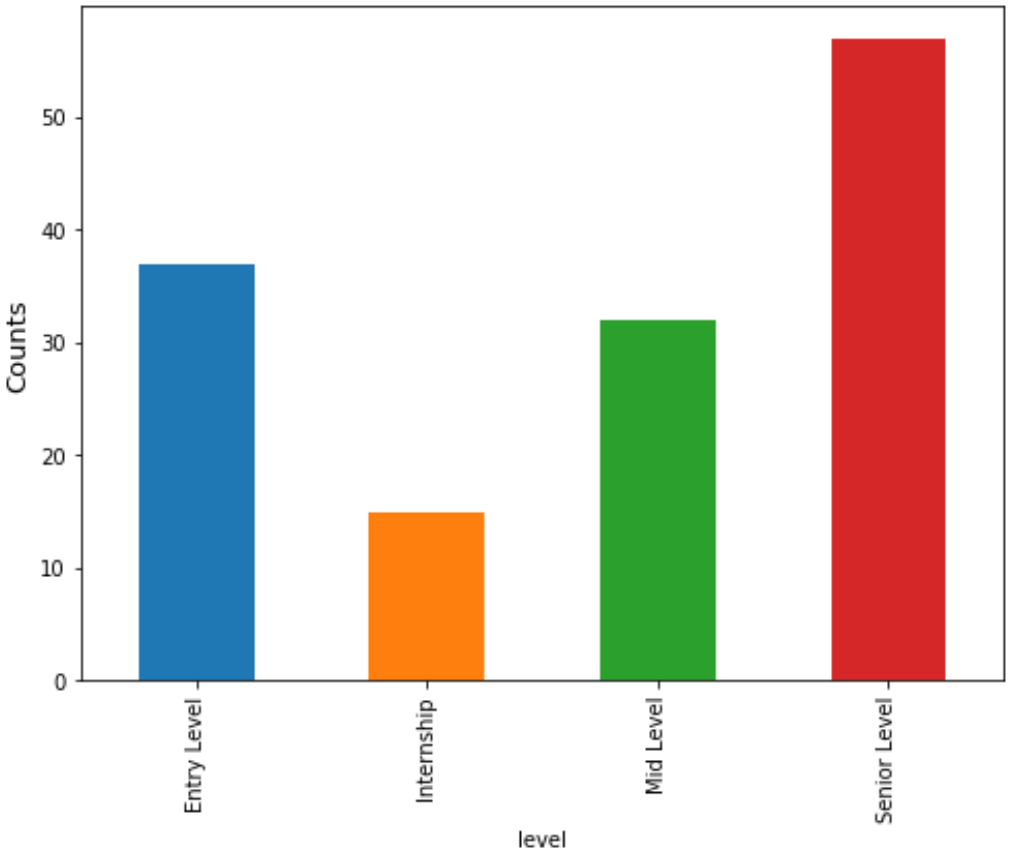
Approach, Solution, and Results

The JSON dataset consists of three features such as Level, description, and title. The task was to find the missing levels. After seeing the text corpus, I addressed this task as a multiclass classification problem.

The text corpus includes 4 classes where all the classes are unbalanced. Due to insufficient data, the upsampling or downsampling methods is not involved.

Fig: Labels count in the training data.



Steps involved to solve this problem:

step1: Data preprocessing

- Stopwords removal
- HTML decoding
- Remove extra spaces
- Remove numbers

- Remove (word>1)

Step2: Feature engineering

- Countervector for features Description and Title
- Tfidf for Description and Title
- Concatenate both features for training data
- Concatenate both features for prediction data

step3: Classes

- Convert categorical data to numerical
- Use LabelEncoder

Step4: Train/test split

- 90% data for training
- 10% data for testing

Step5: Model building

- Initialize SVM algorithm
- Apply SVM on concatenated features

Step6: Result on test data

Step7: Prediction on unseen data

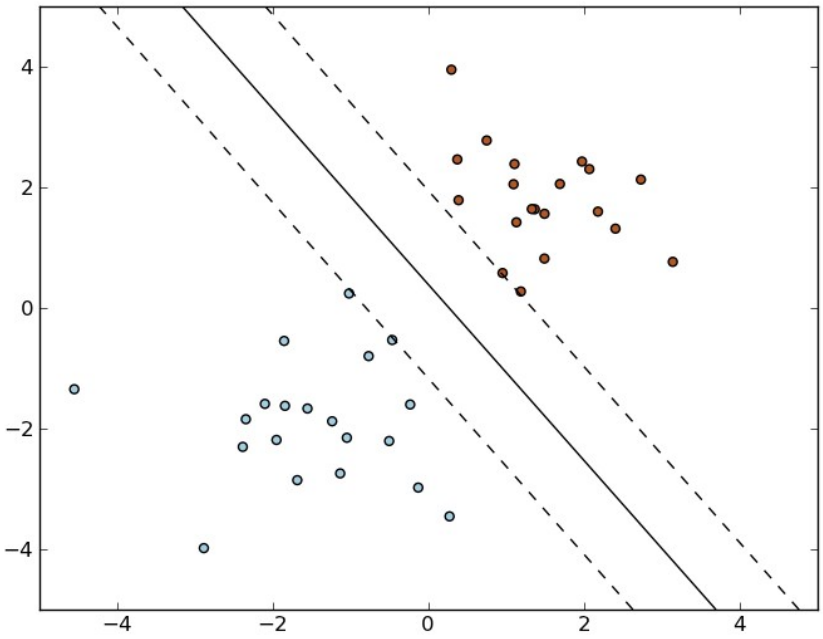
1. **Explain the choice of language / technology stack.**

PYTHON:

Python is easy for analysts to learn and use, but powerful enough to tackle most difficult problems in machine learning and deep learning. It integrates well with existing IT infrastructure and is independent of the platform. In terms of performance, It is fast and robust. For this task, I have used list and dictionary which are much faster in python compare to any other programming languages.

2. **Explain the choice of approach and algorithm.**

I addressed this problem as a multiclass classification problem. A Support Vector Machine (SVM) is a discriminative classifier formally defined by a separating hyperplane. In other words, given labeled training data (supervised learning), the algorithm outputs an optimal hyperplane which categorizes new examples. In two dimensional space, this hyperplane is a line dividing a plane into two parts wherein each class lay in either side.



Reference : http://scikitlearn.sourceforge.net/0.7/modules/sgd.html

3. Estimate quality of the result.

Our dataset has 141 samples for training. The dataset is divided into 90% for training and 10% for the test. After splitting the data, I got 126 samples for training and 15 samples for testing.

The results on test data are given below:

Table: Classification reports

Class name	Precision	Recall	F1 Score	Support
Entry Level	1.00	0.25	0.40	4
Internship	1.00	1.00	1.00	2
Mid Level	0.25	0.33	0.29	3
Senior Level	0.62	0.83	0.71	6

Fig: Confusion Matrix for test data.