

---

# **EntityEmbeddingsCategorical Documentation**

*Release 1.0.0*

**Rodrigo Bresan**

**Dec 09, 2019**



---

# Contents

---

<b>1</b>	<b>About the Project</b>	<b>3</b>
1.1	Entity Embeddings . . . . .	3
<b>2</b>	<b>Getting Started</b>	<b>5</b>
2.1	Installation . . . . .	5
<b>3</b>	<b>Usage</b>	<b>7</b>
3.1	Default . . . . .	7
3.2	Custom . . . . .	7
3.3	Visualization . . . . .	7
3.4	Troubleshooting . . . . .	8
<b>4</b>	<b>Project Modules</b>	<b>9</b>
4.1	entity_embeddings.config . . . . .	9
4.2	entity_embeddings.embedder . . . . .	9
<b>5</b>	<b>Indices and tables</b>	<b>11</b>



Contents:



### 1.1 Entity Embeddings

This project is aimed to serve as an utility tool for the preprocessing, training and extraction of entity embeddings through Neural Networks using the Keras framework. It's still under construction, so please use it carefully.

GitHub Link: [https://github.com/bresan/entity\\_embeddings\\_categorical/](https://github.com/bresan/entity_embeddings_categorical/)

Travis CI Build: [https://travis-ci.org/bresan/entity\\_embeddings\\_categorical](https://travis-ci.org/bresan/entity_embeddings_categorical)

Coverall: [https://coveralls.io/github/bresan/entity\\_embeddings\\_categorical](https://coveralls.io/github/bresan/entity_embeddings_categorical)





### 2.1 Installation

The installation is pretty simple if you have a virtualenv already installed on your machine. If you don't please rely to VirtualEnv official documentation.



The usage of this utility library is provided in two modes: default and custom. In the default configuration, you can perform the following operations: Regression, Binary Classification and Multiclass Classification.

If your data type differs from any of these, you can feel free to use the custom mode, where you can define most of the configurations related to the target processing and output from the neural network.

### 3.1 Default

The usage of the default mode is pretty straightforward, you just need to provide a few parameters to the Config object:

So for creating a simple embedding network that reads from file `sales_last_semester.csv`, where the target name is `total_sales`, with the desired output being a binary classification and with a training ratio of 0.9, our Python script would look like this:

### 3.2 Custom

If you intend to customize the output of the Neural Network or even the way that the target variables are processed, you need to specify these when creating the configuration object. This can be done by creating a class that extend from `TargetProcessor` and `ModelAssembler`.

### 3.3 Visualization

Once you are done with the training of your model, you can use the module `visualization_utils` in order to create some visualizations from the generated weights.

## 3.4 Troubleshooting

In case of any issue with the project, or for further questions, do not hesitate to open an issue on [GitHub](#).

---

## Project Modules

---

This page contains the list of project's modules

---

```
entity_embeddings.config  
entity_embeddings.embedder
```

---

### **4.1 entity\_embeddings.config**

### **4.2 entity\_embeddings.embedder**



## CHAPTER 5

---

### Indices and tables

---

- `genindex`
- `modindex`
- `search`