



Universidade Federal
do Rio de Janeiro

Escola Politécnica

IDENTIFICAÇÃO FOTOMÉTRICA DE SUPERNOVAS ATRAVÉS DE ALGORITMOS DE MACHINE LEARNING

Felipe Matheus Fernandes de Oliveira

Projeto de Graduação apresentado ao Curso de Engenharia de Controle e Automação da Escola Politécnica, Universidade Federal do Rio de Janeiro, como parte dos requisitos necessários à obtenção do título de Engenheiro.

Orientadores: Amit Bhaya

Ribamar Rondon de Rezende
dos Reis

Rio de Janeiro
Julho de 2019

IDENTIFICAÇÃO FOTOMÉTRICA DE SUPERNovas ATRAVÉS DE
ALGORITMOS DE MACHINE LEARNING

Felipe Matheus Fernandes de Oliveira

PROJETO DE GRADUAÇÃO SUBMETIDO AO CORPO DOCENTE DO
CURSO DE ENGENHARIA DE CONTROLE E AUTOMAÇÃO DA ESCOLA
POLITÉCNICA DA UNIVERSIDADE FEDERAL DO RIO DE JANEIRO COMO
PARTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU
DE ENGENHEIRO DE AUTOMAÇÃO.

Examinado por:

Prof. Amit Bhaya, D.Sc.

Prof. Ribamar Rondon de Rezende dos Reis, Ph.D.

Prof. Heraldo Luís Silveira de Almeida, D.Sc.

Prof. Nome do Segundo Examinador Sobrenome, D.Sc.

RIO DE JANEIRO, RJ – BRASIL
JULHO DE 2019

Fernandes de Oliveira, Felipe Matheus

IDENTIFICAÇÃO FOTOMÉTRICA DE
SUPERNOVAS ATRAVÉS DE ALGORITMOS DE
MACHINE LEARNING/Felipe Matheus Fernandes de
Oliveira. – Rio de Janeiro: UFRJ/ Escola Politécnica,
2019.

XI, 18 p.: il.; 29,7cm.

Orientadores: Amit Bhaya

Ribamar Rondon de Rezende dos Reis

Projeto de Graduação – UFRJ/ Escola Politécnica/
Curso de Engenharia de Controle e Automação, 2019.

Referências Bibliográficas: p. 17 – 17.

1. Machine Learning. 2. Gaussian Process Fitting.
3. Supernova Photometric Identification. I. Bhaya,
Amit *et al.* II. Universidade Federal do Rio de Janeiro,
Escola Politécnica, Curso de Engenharia de Controle e
Automação. III. Título.

*A alguém cujo valor é digno
desta dedicatória. A você Goku*

Agradecimentos

Gostaria de agradecer a todos os Sayajins por terem nos salvado das garras maléficas de Freeza, Cell e Majin Boo.

Resumo do Projeto de Graduação apresentado à Escola Politécnica/ UFRJ como parte dos requisitos necessários para a obtenção do grau de Engenheiro de Automação.

IDENTIFICAÇÃO FOTOMÉTRICA DE SUPERNOVAS ATRAVÉS DE ALGORITMOS DE MACHINE LEARNING

Felipe Matheus Fernandes de Oliveira

Julho/2019

Orientadores: Amit Bhaya

Ribamar Rondon de Rezende dos Reis

Curso: Engenharia de Controle e Automação

Com a finalidade de estudar a expansão do universo, a cosmologia busca classificar diferentes tipos de objetos astronômicos. Entretanto, com o crescente aumento do número de objetos detectados, o método normalmente usado para a classificação (espectroscopia) torna-se muito custoso. Como consequência, utiliza-se um método com baixo custo (fotometria) embasado em algoritmos de aprendizado de máquina para a classificação desse vasto número de dados. Nesse contexto, o presente trabalho estuda otimizações para a melhoria desses algoritmos de aprendizado de máquina.

Abstract of Undergraduate Project presented to POLI/UFRJ as a partial fulfillment of the requirements for the degree of Engineer.

SUPERNOVA PHOTOMETRIC IDENTIFICATION USING MACHINE LEARNING ALGORITHMS

Felipe Matheus Fernandes de Oliveira

July/2019

Advisors: Amit Bhaya

Ribamar Rondon de Rezende dos Reis

Course: Automation and Control Engineering

To explore the expansion history of the universe, cosmology classifies different types of astronomic objects using spectroscopy. With their sample sizes increasing, spectroscopy methods cannot handle this amount of data. As a solution to this issue, photometric identification is crucial to fully exploit these large samples due to its simplicity. Once photometric identification uses machine learning algorithms, the following work tries to optimize those algorithms.

Sumário

Lista de Figuras	x
Lista de Tabelas	xi
1 Introdução	1
1.1 Tema e Contextualização	1
1.2 Problemática	2
1.3 Delimitação	2
1.4 Objetivos	3
1.4.1 Processo Gaussiano <i>Gaussian Process</i>	3
1.4.2 Aprendizagem Profunda	4
1.4.3 Tratamento de <i>Outliers</i>	4
1.5 Metodologia	6
1.6 Descrição	6
2 Fundamentos Teóricos	8
2.1 Processo Gaussiano <i>Gaussian Process</i>	8
2.2 Rede Neural Convolucional	8
2.3 Demais Conceitos	8
2.3.1 Análise de Componentes Principais	8
2.3.2 Transformadas de <i>Wavelet</i>	8
2.3.3 Validação Cruzada	8
3 Modelo Atual	9
3.1 Pré-processamento dos Dados Brutos	9
3.2 <i>Pipeline</i> Atual	10
4 Tratamento de <i>Outliers</i>	12
4.1 Estratégias Utilizadas	12
4.2 Resultados e comparações	12

5	Rede Neural Convolutacional	13
5.1	Geração de Imagens e Parâmetros	13
5.2	Escolhas de Redes Neurais	13
5.3	Resultados e comparações	13
6	Interpolação através de Processo Gaussiano	14
6.1	Escolha da Biblioteca	14
6.2	Aleatoriedades e <i>Random Seeds</i>	14
6.3	<i>Kernel Functions</i>	14
6.4	Demais Observações	14
6.5	Resultados das Interpolações	14
6.6	Identificação e Justificativa do Erro	14
7	Resultados e Discussões	15
8	Conclusões	16
8.1	Conclusões Finais	16
8.2	Trabalhos Futuros	16
	Referências Bibliográficas	17
A	Algumas Demonstrações	18

Lista de Figuras

1.1	Exemplo da interpolação não condizente com a realidade.	3
1.2	Exemplo de <i>outlier</i> . Gráfico Fluxo \times Tempo (dias).	5
1.3	Exemplo de ponto com incerteza 3 vezes maior que o pico da curva. Gráfico Fluxo \times Tempo (dias).	5
3.1	Arquivo <i>.txt</i> a ser lido pelos arquivos de pré-processamento.	10
3.2	Exemplo de 4 curvas de luz de um mesmo objeto, uma para cada filtro. Gráfico Fluxo \times Tempo (dias).	11

Lista de Tabelas

Capítulo 1

Introdução

1.1 Tema e Contextualização

Dentro da cosmologia, existe a necessidade de determinar distâncias luminosas ?? para modelar seus estudos, e como ferramenta dessa tarefa, utilizam-se as curvas de luz provenientes de supernovas do tipo IA.

No passado, o conjunto de dados de supernovas era pequeno o suficiente para poder analisar a maior parte dos objetos usando o método da espectroscopia, um método lento e custoso que, no entanto, confirma precisamente o tipo de cada uma delas.

Contudo, com o avanço das pesquisas e da tecnologia de telescópios, a astronomia está entrando em uma era de conjuntos massivos de dados, tornando-se necessário a adoção de técnicas automatizadas mais simples e práticas para classificar a enorme quantidade de objetos astronômicos captados, pois, através da espectroscopia não seria possível.

Nesse contexto, foram desenvolvidas diferentes abordagens para levantar essa grande quantidade de objetos captados através de um método chamado fotometria. Dentre essas abordagens, várias utilizam aprendizado de máquina.

Por fim, a ideia do projeto foi derivada de um artigo publicado pela cientista LOCHNER [1]. A autora busca criar uma maneira automática de classificação fotométrica usando as curvas de luz obtidas através da fotometria, essas que já foram devidamente classificadas no passado utilizando espectroscopia. Caracterizando assim um problema de classificação, visto que temos os tipos de supernovas dados pela espectroscopia, e suas representações fotométricas para que possamos criar e treinar modelos.

1.2 Problemática

Tendo em vista que o artigo [1] testou e validou diversos *pipelines*, neste trabalho adotamos aquele que ela obteve o melhor resultado de classificação, e a partir desse ponto, aplicamos modificações e avaliamos quaisquer possíveis melhoras.

O *pipeline* escolhido é constituído majoritariamente de 4 partes:

- Processo Gaussiano (*Gaussian Process* ou **GP**)
- Transformada de Wavelet
- Análise de componentes principais (*Principal Component Analysis* ou **PCA**)
- Floresta Aleatória (*Random Forest* ou **RF**)

A problemática encontra-se no método de interpolação chamado **Processo Gaussiano**. Por ser um método de interpolação, espera-se que o mesmo defina uma função que passe pelos pontos obtidos respeitando suas incertezas. Entretanto, em alguns casos o gráfico interpolado é uma constante 1.1. O que não possui sentido físico, visto que se trata do fluxo de luz após uma explosão de um objeto astronômico.

Através do Instituto de Física da UFRJ surgiu a ideia principal deste projeto que é a de tentar corrigir essas interpolações que apresentam comportamento constante durante toda a observação.

Em paralelo, outras ideias foram aplicadas na tentativa de obter uma melhora no algoritmo original da *Lochner*.

1.3 Delimitação

Todos os dados utilizados tanto neste trabalho quanto no artigo [1], foram extraídos da base de dados oferecida pelo desafio KESSLER [2]. A base de dados é de domínio público e pode ser obtida no [LINK DO DOWNLOAD](#).

Devido a granularidade dos dados brutos, é necessário um pré-tratamento visando buscar apenas os dados que iremos utilizar. Esse pré-tratamento foi aproveitado do *pipeline* do Instituto de Física da UFRJ (**IF-UFRJ**), cujo processo visa ler os arquivos de texto transformando cada informação em uma chave de dicionário.

Ao fim do uso da seleção de dados citada acima, obtivemos valores em forma de dicionários em Python para cada objeto astronômico. Esses dicionários serão os **Dados Brutos** no escopo deste trabalho.

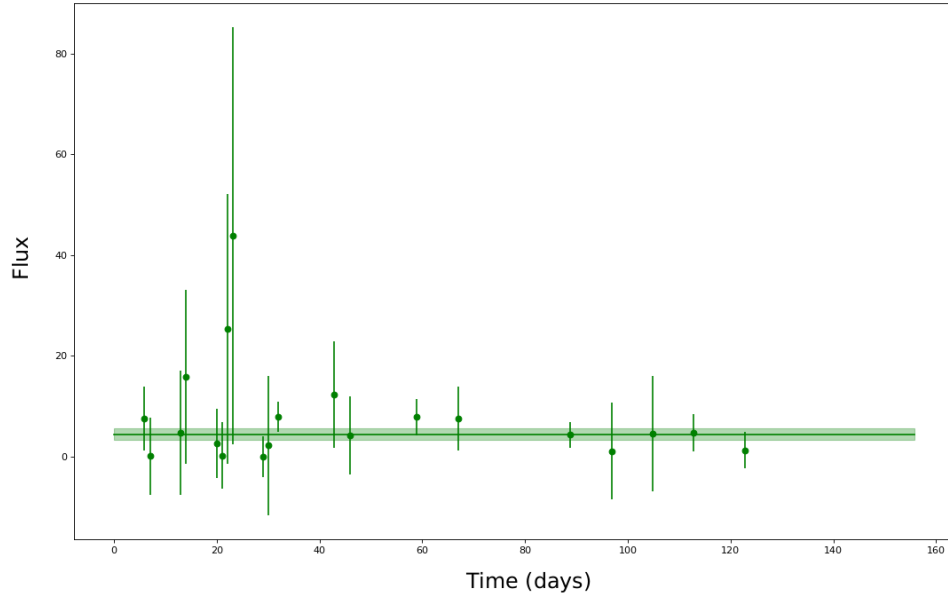


Figura 1.1: Exemplo da interpolação não condizente com a realidade.

1.4 Objetivos

O objetivo do trabalho é otimizar determinados pontos do *pipeline* utilizado atualmente, após um estudo inicial ter sido feito para identificar pontos frágeis ou passíveis de melhoras.

Esses pontos são apresentados e justificados a seguir, criando as três vertentes do trabalho

- Processo Gaussiano
- Aprendizagem Profunda
- Tratamento de *Outliers*

1.4.1 Processo Gaussiano *Gaussian Process*

O Processo Gaussiano, é a primeira parte do tratamento de dados após a transformação dos dados brutos em dicionários de Python. Iremos abordar esse procedimento no Capítulo 6.

Cada objeto astronômico irá possuir uma quantidade de pontos que representam a intensidade do fluxo de luz captado em determinado dia, bem como a incerteza desse valor. Assim, buscamos fazer uma interpolação para obter um gráfico que passe por determinados pontos.

O objetivo ao abordar esse ponto é consertar interpolações que não condizem com a interpretação física, como pode ser vista na figura 1.1.

1.4.2 Aprendizagem Profunda

A Aprendizagem Profunda (*Deep Learning* ou **DL**) vêm sendo um advento utilizado recentemente para aprimorar alguns algoritmos de Aprendizagem de Máquina (*Machine Learning* ou **ML**).

A ideia de aplicar DL nesse problema veio a partir da análise do *pipeline* inicial (visto na seção 1.2). Tendo em vista que o mesmo possui diversas partes entre os dados brutos e algoritmo de RF, buscamos aplicar DL esperando obter um *pipeline* mais simples, possuindo apenas uma etapa entre os dados brutos e o algoritmo de classificação.

Em diversos outros casos na literatura, como identificação de sons ou imagens, houveram melhorias significativas ao reduzir o número de etapas envolvendo processamento de sinais (wavelets) ou redução de componentes (PCA) em troca do uso de DL.

A segunda motivação para a aplicação de DL veio do artigo da LOCHNER [1], onde ela menciona possíveis aplicações no escopo do problema.

Assim, foi decidido aplicar algoritmos de DL após a primeira parte do tratamento de dados, visando manter uma simplicidade ao longo do *pipeline*. Estabelecendo assim, apenas a etapa do GP entre os dados brutos e o algoritmo de classificação. O detalhamento desse procedimento encontra-se no Capítulo 5.

1.4.3 Tratamento de *Outliers*

O último objetivo para buscar melhorias no algoritmo é o tratamento de pontos cuja incerteza é muito alta ou cujos valores são fora do esperado, pontos com essas características são conhecidos na literatura como *outliers*.

Analisando alguns dados brutos, observamos pontos cuja incerteza mostra-se mais de 100% maior do que a própria curva (ponto mais a esquerda da figura 1.3) e também pontos que encontram-se completamente fora da curva interpolada 1.2.

As etapas do tratamento encontram-se descritas no Capítulo 4.

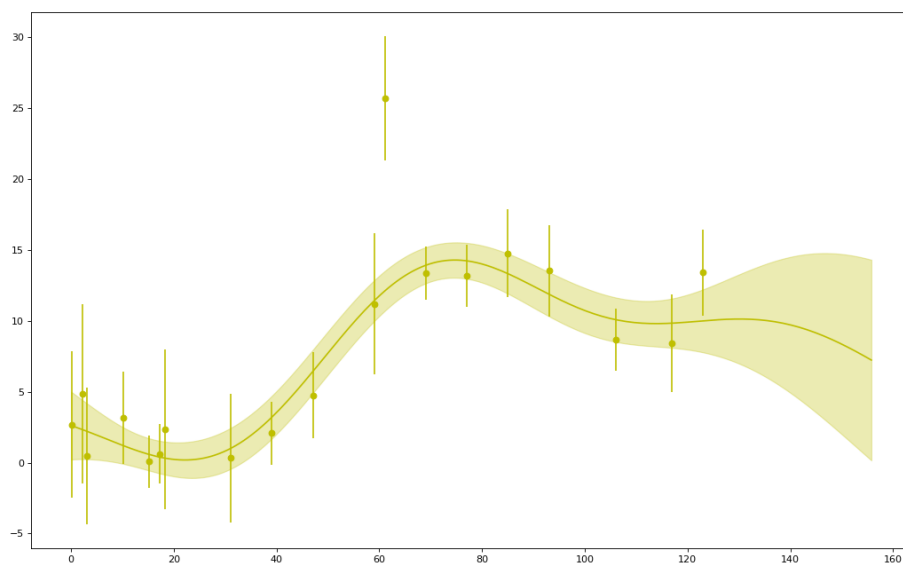


Figura 1.2: Exemplo de *outlier*. Gráfico Fluxo \times Tempo (dias).

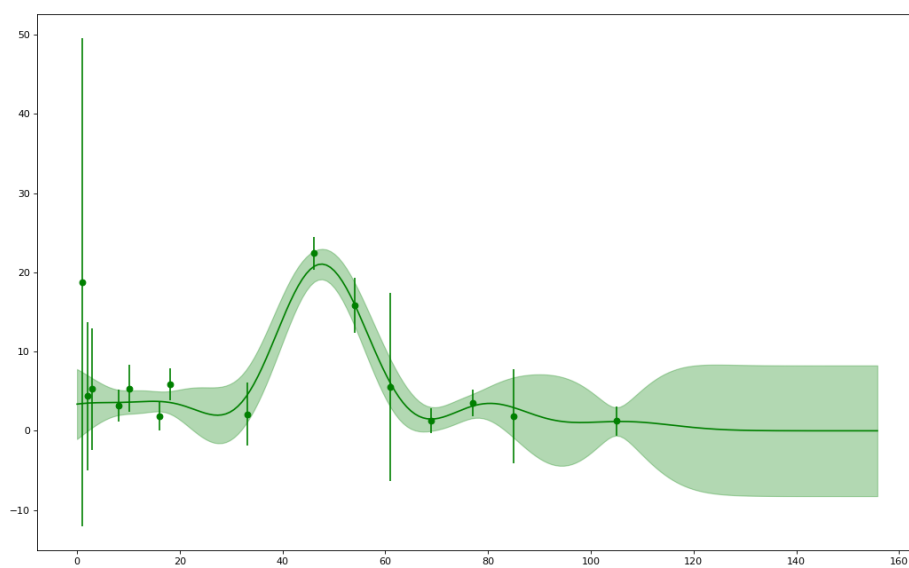


Figura 1.3: Exemplo de ponto com incerteza 3 vezes maior que o pico da curva. Gráfico Fluxo \times Tempo (dias).

1.5 Metodologia

Devido ao código de pré-processamento do IF-UFRJ estar escrito em Python, todo o trabalho também foi escrito na mesma linguagem. Outro ponto para a escolha da linguagem foi a facilidade que a mesma possui para realizar experimentos através da plataforma *Jupyter*, além da grande quantidade de bibliotecas de ciências de dados e suporte para desenvolver algoritmos de ML.

As principais bibliotecas utilizadas no contexto de tratamento de dados foram *Pandas* e *Numpy*. Já para a aplicação dos métodos as bibliotecas mais importantes foram *scikit-learn*, *Keras* e *PyMC3*.

A entrada dos modelos sempre serão os dicionários gerados a partir dos dados brutos após o pré-processamento do IF-UFRJ. Assim, pôde-se filtrar com maior facilidade as propriedades desejadas para cada uma das três vertentes do trabalho.

Cada uma dessas vertentes possuiu um trabalho específico e isolado para poder melhor averiguar suas mudanças posteriormente. Em outras palavras, a cada mudança realizada houve a comparação direta com os resultados do *pipeline* original, para poder avaliar especificamente a eficiência e melhora dessa mudança.

Por se tratar de um algoritmo de classificação, a saída do nosso modelo será booleana e classificará se um objeto astronômico é ou não do tipo IA. Entretanto, nos dados brutos, as classes são divididas em 8 tipos de objetos: *IA*, *IB*, *IBC*, *IC*, *II*, *IIL*, *IIN* e *IIP*.

É importante ressaltar também que para validar os modelos do projeto, foi usado o método *K-fold*. Todavia, este trabalho se diferenciou da divisão clássica presente na literatura, que é uma divisão de 80% dos dados para treinamento e 20% para o teste dos modelos. Neste trabalho foram utilizados apenas 1100 objetos para treino dentre 21316 objetos totais. A justificativa é a limitação física das amostras.

Por fim, as mudanças foram verificadas através de **Matrizes de Confusão** (*Confusion Matrices* ou **CM**) ou através de *score* da biblioteca *scikit-learn*.

Como última observação, a fim de estudos posteriores fora do escopo do projeto, todo o código desenvolvido encontra-se disponível em um repositório público do *GitHub*. E nele foram feitas classificações envolvendo os 8 tipos de objetos, além de treinamentos adotando a proporção de 80% para treino e 20% para teste.

1.6 Descrição

No Capítulo 2, iremos explicar os principais conceitos teóricos utilizados na aplicação do **Processo Gaussiano** e no *Deep Learning*. Também serão levantados alguns conceitos não tão centrais do trabalho, mas de importância igualmente significativa.

O Capítulo 3 será destinado para detalhar especificamente cada etapa do *pipeline* atual, assim como uma breve menção a maneira na qual os dados brutos são tratados.

Os capítulos 4, 5 e 6 serão destinados à explicação dos objetivos citados na seção 1.4, assim como os resultados das mudanças que os mesmos causaram, buscando averiguar a eficiência de aplicação ou explicar possíveis erros.

No Capítulo 7 serão apresentados os resultados como um todo, seguidos de análises dos métodos através uma perspectiva global do projeto, diferentemente de como será nos capítulos anteriores, nos quais teremos uma visão específica para cada objetivo.

Por fim, no Capítulo 8 tiraremos as conclusões finais do projeto e serão apresentadas sugestões de trabalhos futuros junto as limitações e possíveis soluções encontradas.

Capítulo 2

Fundamentos Teóricos

2.1 Processo Gaussiano *Gaussian Process*

2.2 Rede Neural Convolucional

2.3 Demais Conceitos

Conceitos menos usados ao longo do *pipeline*, estão brevemente explicados a seguir.

2.3.1 Análise de Componentes Principais

2.3.2 Transformadas de *Wavelet*

2.3.3 Validação Cruzada

Capítulo 3

Modelo Atual

Neste capítulo iremos detalhar o modelo atual utilizado para estudos cosmológicos do IF-UFRJ.

3.1 Pré-processamento dos Dados Brutos

Como pode ser visto na figura 3.1, os dados brutos chegam da seguinte forma para o pré-processamento.

Através dos arquivos de pré-tratamento iremos obter todas essas informações em forma de dicionário, entretanto apenas as seguintes serão utilizadas durante o projeto.

- **SN TYPE** : Informa o tipo de supernova
- **TERSE LIGHT CURVE OUTPUT** : Informa os pontos de observações obtidos. Será nosso *Data Frame*.
 - **MJD** : O momento em que a observação foi obtida. A medida está descrita em dias do calendário Juliano.
 - **FLT** : O filtro em que a observação foi obtida, pois cada observação é obtida a partir de um determinado espectro de luz.
 - **FLUXCAL** : O valor do fluxo de luz obtido naquela observação.
 - **FLUXCALERR** : O valor do erro do fluxo de luz obtido naquela observação.

Em relação ao **MJD**, vale ressaltar que é um método usado na astronomia para contar os dias sequencialmente, começando em uma data arbitrária no passado. Neste projeto, essas datas serão normalizadas, tendo como 0 o menor valor do MJD para que nosso eixo das abscissas tenha sempre início em 0.

```

FAKE:      2      (=> simulated LC with snlc_sim.exe)
MWEBV:    0.0111      MW E(B-V)
REDSHIFT_HELIO: 0.35020 +- 0.04680 (Hello, z_best)
REDSHIFT_FINAL: 0.35020 +- 0.04680 (CMB)
REDSHIFT_SPEC: -9.00000 +- 9.00000
REDSHIFT_STATUS: OK

HOST_GALAXY_GALID: 18943
HOST_GALAXY_PHOTO-Z: 0.3502 +- 0.0468

SIM_MODEL: NONIA 10 (name index)
SIM_NON1a: 31 (non1a index)
SIM_COMMENT: SN Type = II , MODEL = SDSS-017862
SIM_LIBID: 4
SIM_REDSHIFT: 0.3972
SIM_HOSTLIB_TRUEZ: 0.4000 (actual Z of hostlib)
SIM_HOSTLIB_GALID: 18943
SIM_DLMU: 41.666473 mag [ -5*log10(10pc/dL) ]
SIM_RA: 0.500000 deg
SIM_DECL: -43.000000 deg
SIM_MWEBV: 0.0113 (MilkyWay E(B-V))
SIM_PEAKMAG: 26.78 28.54 28.42 26.27 (griz obs)
SIM_EXPOSURE: 1.0 1.0 1.0 1.0 (griz obs)
SIM_PEAKMJD: 56214.511719 days
SIM_SALT2x0: 2.155e-17
SIM_MAGDIM: 0.000
SIM_SEARCHEFF_MASK: 3 (bits 1,2=> found by software,humans)
SIM_SEARCHEFF: 1.0000 (spectro-search efficiency (ignores pipelines))
SIM_TRESTMIN: -31.11 days
SIM_TRESTMAX: 64.81 days
SIM_RISETIME_SHIFT: 0.0 days
SIM_FALLTIME_SHIFT: 0.0 days

SEARCH_PEAKMJD: 56214.434

# =====
# TERSE LIGHT CURVE OUTPUT:
#
NOBS: 93
NVAR: 9
VARLIST: MJD FLT FIELD FLUXCAL FLUXCALERR SNR MAG MAGERR SIM_MAG
OBS: 56171.051 g NULL -3.679e+01 7.809e+01 -0.47 99.000 5.000 99.055
OBS: 56172.039 r NULL 4.410e+00 1.758e+01 0.25 99.000 5.000 98.914
OBS: 56172.051 i NULL 6.447e-01 4.995e+01 0.01 99.000 5.000 98.994
OBS: 56176.160 z NULL 1.337e+00 3.581e+00 0.37 27.184 100.816 59.034
OBS: 56179.023 g NULL 6.122e+00 6.377e+00 0.96 25.533 102.467 60.656
OBS: 56179.031 r NULL -3.697e+00 4.331e+00 -0.85 128.000 0.000 63.358
OBS: 56179.047 i NULL 1.439e+01 7.145e+00 2.01 24.605 0.745 59.327
OBS: 56179.062 z NULL 5.544e+00 8.139e+00 0.68 25.640 102.359 54.103
OBS: 56180.031 g NULL 2.865e+00 5.671e+00 0.51 26.357 101.643 59.296
OBS: 56180.047 r NULL 1.002e+00 4.492e+00 0.22 27.498 100.502 61.984
OBS: 56180.062 i NULL 1.095e+01 6.740e+00 1.62 24.902 1.038 57.955
OBS: 56180.078 z NULL -1.652e+00 1.016e+01 -0.16 128.000 0.000 52.927
OBS: 56188.004 g NULL 5.279e+00 5.677e+00 0.93 25.694 102.306 52.990
OBS: 56188.016 r NULL 5.595e-01 4.368e+00 0.13 28.130 99.870 52.880
OBS: 56188.027 i NULL 3.962e+00 6.462e+00 0.61 26.005 101.995 50.157

```

Figura 3.1: Arquivo *.txt* a ser lido pelos arquivos de pré-processamento.

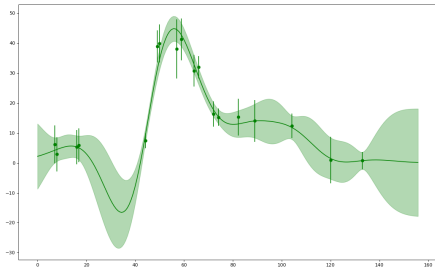
Em relação ao **FLT**, cada objeto astronômico é visto através de 4 filtros de luz diferentes, obtendo assim 4 curvas de luz para cada um deles. Para efeitos de aprendizagem de máquina, cada objeto possuirá 4 *data frames*, um para cada filtro, e posteriormente essas propriedades serão convertidas em *features* de cada objeto.

A figura 3.2 ilustra as 4 curvas de luz para o mesmo objeto.

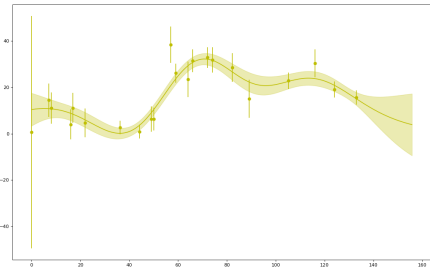
3.2 Pipeline Atual

A partir do pré-processamento explicado na seção anterior, o *pipeline* atual irá dispor destes dados para realizar as etapas do processo de classificação.

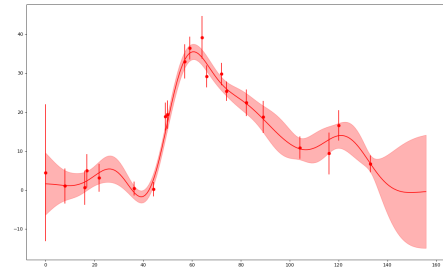
Inicialmente ele irá separar os dados do *Terse Light Curve Output* pelos filtros e irá criar 4 *dataframes* possuindo 3 *features* cada, são eles: **MJD**, **FLUXCAL** e **FLUXCALERR**. ??



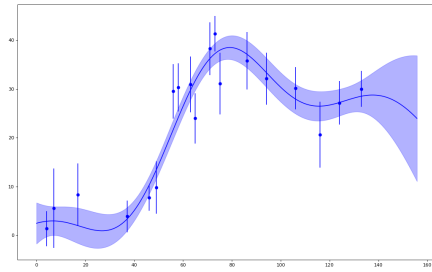
(a) Filtro *desg*



(b) Filtro *desi*



(c) Filtro *desr*



(d) Filtro *desz*

Figura 3.2: Exemplo de 4 curvas de luz de um mesmo objeto, uma para cada filtro. Gráfico Fluxo \times Tempo (dias).

Capítulo 4

Tratamento de *Outliers*

4.1 Estratégias Utilizadas

4.2 Resultados e comparações

Capítulo 5

Rede Neural Convolucional

A ideia foi blablabla usar deep learning, vimos exemplos das roupas, e colocamos la deixamos a princípio o pipeline do Marcelo

5.1 Geração de Imagens e Parâmetros

Os parâmetros na hora de gerar as imagens e o PQ

5.2 Escolhas de Redes Neurais

escolhemos 2CNN pra fazer por isso e por aquilo

5.3 Resultados e comparações

Capítulo 6

Interpolação através de Processo Gaussiano

6.1 Escolha da Biblioteca

6.2 Aleatoriedades e *Random Seeds*

6.3 *Kernel Functions*

6.4 Demais Observações

Valor negativo e Mean function, erro grande, melhor overfittado que underfittado

6.5 Resultados das Interpolações

6.6 Identificação e Justificativa do Erro

Capítulo 7

Resultados e Discussões

Capítulo 8

Conclusões

8.1 Conclusões Finais

Falar que o Algoritmo de hoje já é bem robusto

8.2 Trabalhos Futuros

Falar de consertar o Erro e pah

Referências Bibliográficas

- [1] LOCHNER, M. “Photometric Supernova Classification with Machine Learning”, *ApJS (2016)*, v. 225, n. 2, pp. 31, set. 2016.
- [2] KESSLER, R. “Supernova Photometric Classification Challenge”, *arXiv*, v. 6, n. 5210, pp. 4, abr. 2010.

Apêndice A

Algumas Demonstrações