**Universidad Francisco Marroquín**

**Facultad de Ciencias Económicas**

**Computer Science**

**Catedrático: Luis Fernando Leal   
Lunes y Miercoles 19:00**

# **Programa de curso**

# Machine Learning

**Objetivos del curso**

* Conocer los campos y áreas de aplicación de Machine Learning
* Conocer los fundamentos científicos detrás de los algoritmos de Machine Learning
* Identificar potenciales aplicaciones de Machine Learning en la práctica
* Aprender los diferentes sub-tipos de algoritmos de Machine Learning e identificar el mejor tipo para una aplicación especifíica.
* Aprender como transformar un conjunto de datos, en insumo para un algoritmo de Machine Learning y obtener un modelo a partir de esto.
* Conocer las diferentes métricas para evaluar la calidad de las tareas y aplicaciones de Machine Learning.
* Aplicar estas técnicas y conceptos en Python

**Oportunidades del curso**

El alumno aprenderá a derivar inteligencia a partir de un conjunto de datos, y aplicar esta para crear herramientas predictivas ,y encontrar patrones . Aprenderá como aplicar y que método aplicar para aplicaciones y tareas específicas , como evaluar la calidad de sus algoritmos y modelos, y como transformar los datos para hacer esto posible.

**Reglamento del curso**

El curso sera impartido de manera virtual , a través de sesiones interactivas entre profesor y alumnos utilizando herramientas de Google como “Google Hangouts on Air”, el alumno puede en todo momento interactuar y/o preguntar ya sea por micrófono o por el chat de la sesión y puede solicitar tomar el control de la sesión para mostrar su pantalla en caso de querer mostrar o preguntar algo respecto a su trabajo. La sesión sera almacenada automáticamente en Youtube para futuros repasos y/o referencia.

Habrá sesiones presenciales en casos específicos tales como exámenes , exposiciones ,o proyectos/exámenes prácticos

**Descripción de 18 semanas 2 veces por semana sesiones de 1.20 - horas cada una**

|  |  |  |  |
| --- | --- | --- | --- |
| **Tema** | **Lecturas y Contenido (secuencias y temas específicos)** | **Aplicaciones Prácticas**  **(hands-on)** | **Tareas asignadas este día para la siguiente sesión** |
| 1 | * Introducción del profesor y auxiliar * Introducción de los alumnos , background, intereses,experiencia, etc | Examen de ubicación(no tiene nota en el curso, es para evaluar conocimiento en programación, matematica y estadística) |  |
| 2 | * Bienvenidos a machine learning!:ejemplos , casos de uso conocidos, motivación, experiencias , antecedentes, industria y actualidad. |  |  |
| 3 | * Introducción y descripción a las herramientas del curso. * Repaso básico: matemática(Calculo) para ML |  | * Instalación de herramientas * Crear cuenta en Github |
| 4 | * Repaso básico: matemática(álgebra lineal) . * Repaso básico: estadística y probabilidad |  |  |
| 5 | Programación con Python:   * Algoritmos * Variables * Listas y diccionarios * Operaciones aritméticas * Condicionales * Ciclos * Procedimientos y funciones * Librerias y paquetes | Programación básica |  |
| 6 | * Que es machine learning? : definiciones y explicación intuitiva de las definiciones y su relación con estadística tradicional. * Relación con data engineering/data science,big data y business intelligence Y otras disciplinas .DS process. * Jupyter notebooks para data science | Ejercicio: que es y que no es ML entre una lista de ejemplos?  Ejemplo e introducción a Jupyter notebooks | Ensayo: Investigar o proponer 2 aplicaciones de en áreas de tu interés o que llamen tu atención. |
| 7 | * Tipos básicos: aprendizaje supervisado, aprendizaje no supervisado, aprendizaje por refuerzo, otros. * Aprendizaje supervisado * NumPy | Ejercicio Numpy | *Ejercicios Python y Numpy en Jupyter* |
| 8 | * Aprendizaje supervisado: regresión * Aprendizaje supervisado: clasificación. * MatplotLib y seaborn | Ejercicio: identificar regresión y clasificación |  |
| 9 | * Aprendizaje no supervisado * Aprendizaje no supervisado: reducción de dimensionalidad * Pandas |  |  |
| 10 | * Aprendizaje no supervisado: clustering and retrieval * Aprendizaje no supervisado: deteccion de anomalias |  |  |
| 11 | * Aprendizaje por refuerzo * Otros tipos de aprendizaje: sistemas de recomendación,aprendizaje de reglas de asociación, aprendizaje semi-supervisado, modelos generativos,one-shot learning |  | Seleccionar 5 aplicaciones o ideas de interés y clasificar a que tipo de aprendizaje pertenecen.  Recolectar datos ,analizarlos con numpy/pandas y graficarlos en jupyter. |
| 12 | **Examen Parcial 1 – Tipos de aprendizaje y regresion (15 pts.)** | | |
| 13 | Tu primer algoritmo de ML como base del resto de algoritmos( regresión lineal sencilla): utilizaremos como caso base una regresión lineal sencilla para tratar temas que serán comunes en la mayoría de algoritmos de ML que veremos tales como: modelo e hipótesis,parámetros entrenables, función de costo , proceso de “aprendizaje” / entrenamiento y gradient descent,learning rate, proceso de inferencia o predicción , pre-procesamiento de datos y feature engineering(feature scaling and normalization) |  |  |
| 14 | * Regresion lineal multivariable * Regresion polinomial * Solucion analitica: ecuacion normal |  | Ejercicio de regresion utilizando scikit-learn y jupyter. |
| 15 | * Clasificacion: diferencia con regresión, modelo/hipotesis, frontera de decisión * Naive Bayes classifier * K-nearest neighbors * Desicion trees |  |  |
| 16 | Primer algoritmo de clasificación por GD: regresión logística, su función de costo, gradient descent y clasificación multiclase(overview de softmax) |  |  |
| 17 | * Segundo algoritmo de clasificación por GD: support vector machines, objetivo de optimizacion, intuicion “large margin” * Kernels |  | Ejercicio de clasificacion usando scikit-learn y jupyter. |
| 18 | Overview de deep learning y redes neuronales: breve introducción a deep learning, redes neuronales, aplicaciones especiales y similitudes con los algoritmos vistos hasta el momento |  | Investigar 5 avances recientes de deep learning.  Proponer 1 caso de uso no comun o aun no explotado. |
| 19 | Problemas comunes: overview underfitting, overfitting , como identificarlos y posibles soluciones(regularización) |  |  |
| 20 | **Examen Parcial 2– Regresión y Clasificación (15 pts.)** | | |
| 21 | * Clustering: k-means * Reducción de dimensionalidad: PCA |  | Ejercicio en scikit-learn y jupyter. |
| 22 | Deteccion de anomalias |  | Ejercicio en scikit-learn, scipy y jupyter |
| 23 | Sistemas de recomendación |  |  |
| 24 | * Aprendizaje por refuerzo: * RL: proceso de decisión markoviano y cadenas de markov |  |  |
| 25 | RL: q-table e introduccion a q-learning |  | Investigacion: OpenAI gym |
| 26 | * Modelos generativos :overview y ejemplos ( neural-style transfer, text generators, GANs, text embeddings) * Semi-supervised learning: overview y ejemplos |  | Proponer 5 casos de uso o aplicaciones de modelos generativos. |
| 27 | One-shot learning: descripcion , motivación y ejemplo aplicado a computer vision: face recognition |  |  |
| 28 | * Mejorando el performance y,consejos y diseño(tuning de hyperparametros, optimización,y otras técnicas): explicaremos que posibles opciones tiene un desarrollador para mejorar el performance de su sistema, y como priorizar sus siguientes pasos. * Evaluacion de un modelo y seleccion de modelos a traves de cross-validation, k-fold cross valiation |  | Ejercicio k-fold cross validation en scikit-learn y jupyter |
| 29 | * Metricas de performance: accuracy, error, confusion matrix, precicion, recall, f1-score * Diagnosticando bias(underfitt) y variance(overfitt) * Curvas de aprendizaje |  |  |
| 30 | * Regularizacion: l1 regularization, l2 regularization, dropout, artificial data synthesis * Analisis del error * Introduction to Feature engineering, feature engineering vs end-to-end ML y big data en ML |  | Ejercicio feature engineering. |
|  | **Examen Parcial 3– Aprendizaje no supervisado y ML performance (15 pts.)** | | |
| 31 | * Inicialización de parámetros(Xavier initialization) * Priorizando y decidiendo tareas: ceiling analysis * Ensembling,boosting, bagging |  | Ejercicio ensembling |
| 32 | * Otros algoritmos de optimización: stochastic gradient descent, mini-bach gradient descent, momentum, rmsprop ,adagrad, adam. * Large Scale ML: procesamiento distribuido, map reduce y otros frameworks o técnicas, GPUs |  |  |
| 33 | * Machine learning pipeline * Cierre del curso |  | Proponer un caso/ejemplo de machine learning pipeline e implementarlo con scikit-learn |
| 34 | **Examen y proyecto final – 40 puntos** | | |

**Bibliografía**

* Se proporcionara material de apoyo y referencias a lo largo del curso.
* Curso interactivo gratuito para Python: <https://www.codecademy.com/learn/learn-python>
* Tutorial Python: <https://www.tutorialspoint.com/python/>
* Notas del curso “Machine Learning” impartido en Stanford : <http://www.holehouse.org/mlclass/>
* Grokking Deep Learning, Andrew Trask,
* Manipulación de datos con Pandas:  
  <https://www.datacamp.com/community/tutorials/pandas-tutorial-dataframe-python>

**Evaluación**

|  |  |
| --- | --- |
| Parcial 1 | 15 pts |
| Parcial 2 | 15 pts |
| Parcial 3 | 15 pts |
| Tareas , y ejercicios | 15 pts |
| Proyecto final | 20 pts |
| Examen final | 20 pts |