Machine Learning Interpretable: shap, pdp, permutation

Link curso: https://www.coursera.org/projects/machine-learning-interpretable-shap-p

Temas principales:

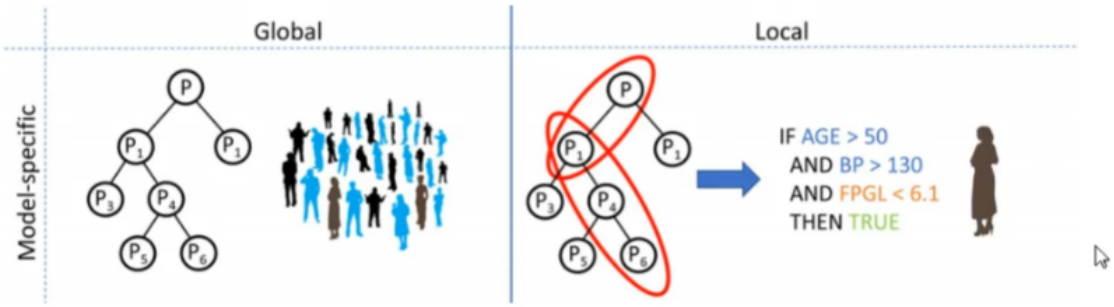
- Tema 1: Introducción a la interpretabilidad de modelos en Machine Learning
- Tarea 2: Desarrollo del modelo de Machine Learning
- Tarea 3: Importancia de las variables: **Permutation Importance**
- Tarea 4: Efecto de las variables: **Partial Dependence Plots**
- Tarea 5: Entendiendo las predicciones individuales: **SHAP**
- Tarea 6: SHAP con LightGBM

TEMA 1. Introducción

Significado

La interpretabilidad es el grado en que un ser humano puede predecir consistentemente el resultado del modelo

Dos términos claves: interpretabilidad global vs interpretabilidad local



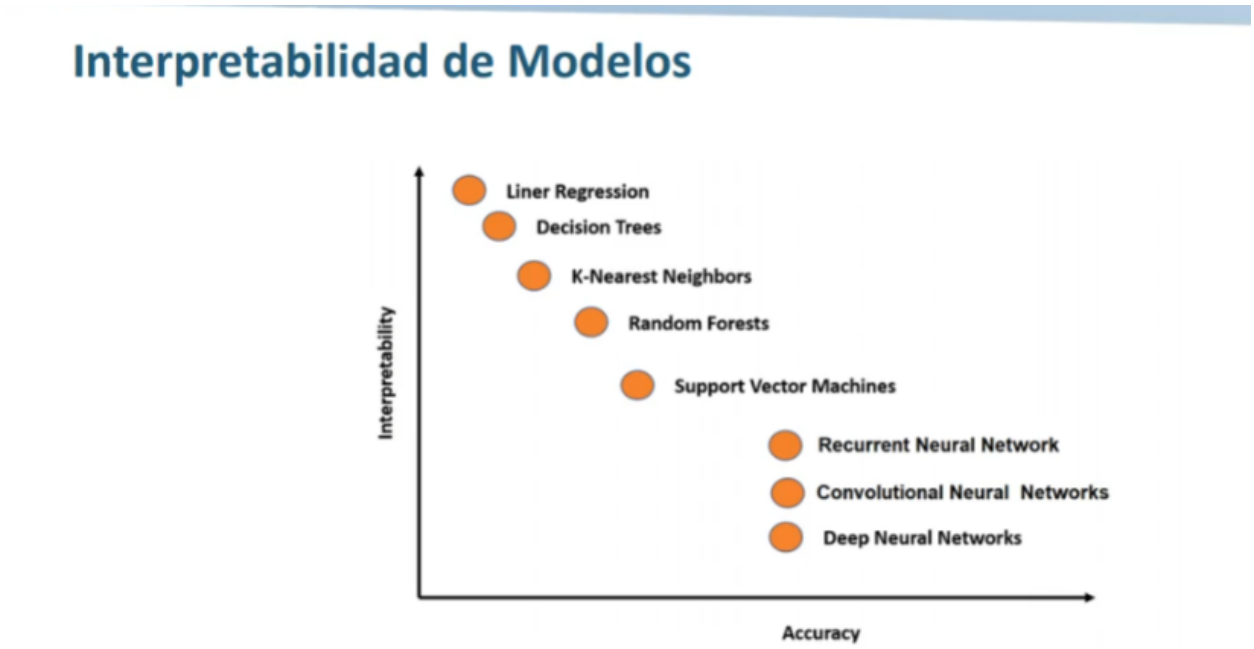
Fuente: https://onlinelibrary.wiley.com/doi/abs/10.1002/widm.1379

global: entender de forma holistica el funcionamiento del modelo. ayuda a comprender la distribución del resultado objetivo dado los objetivos. ej: Enteder si el cliente es bueno o malo

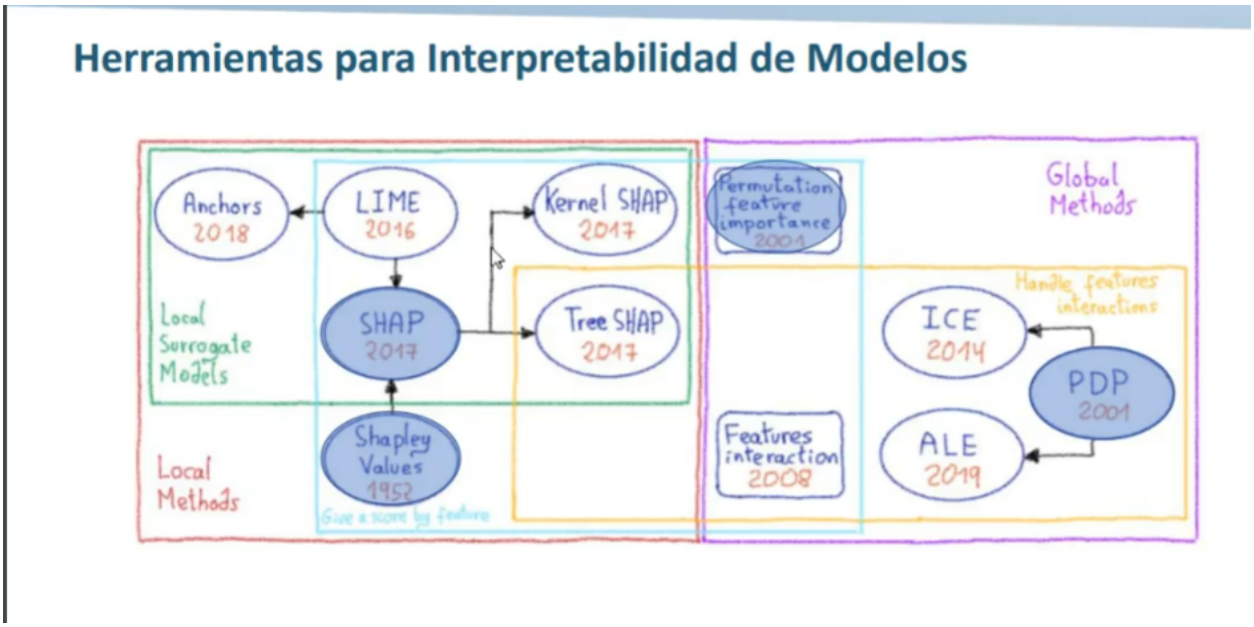
local: entender una sola predicción del modelo. por qué una clasificación de un individuo se considera mala por ejemplo por la edad sobre 50, una masa corporal alta, etc

Nivel de interpretabilidad

Entre más complejo el modelo y más acc tiene (o debería tener), es más dificil interpretarlo. Regresión lineal, árboles de decisión modelos simples con un acc menor que modelos más complejos pero tienen una interpretabilidad mayor. Por el contrario NN son modelos cajas negra, muy alto Acc pero no intepretabilidad



Herramientas para interpretabilidad



Para el curso se van a ver las herramientas que están en azul marcadas.

Packages de Python utilizados

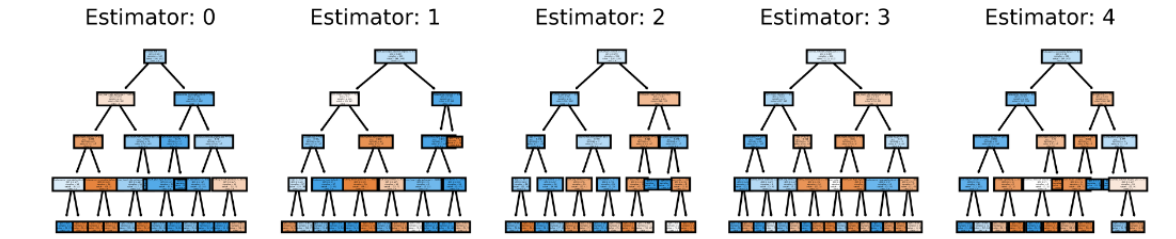
- eli5 (permutation importance)
- pdp (partial dependence plot)
- Shap
- Pdpbox (partial dependence plot - bivariate)

TEMA 2: RF y explicabilidad de este

Dificultad de interpretar modelos complejos

```
[20]: fig, axes = plt.subplots(nrows = 1,ncols = 5,figsize = (10,2), dpi=900)
for index in range(0, 5):
    tree.plot_tree(model.estimators_[index],
                   feature_names = fn,
                   class_names=cn,
                   filled = True,
                   ax = axes[index]);

    axes[index].set_title('Estimator: ' + str(index), fontsize = 11)
fig.savefig('rf_5trees.png')
```



Visualizacion del Random Forest Completo

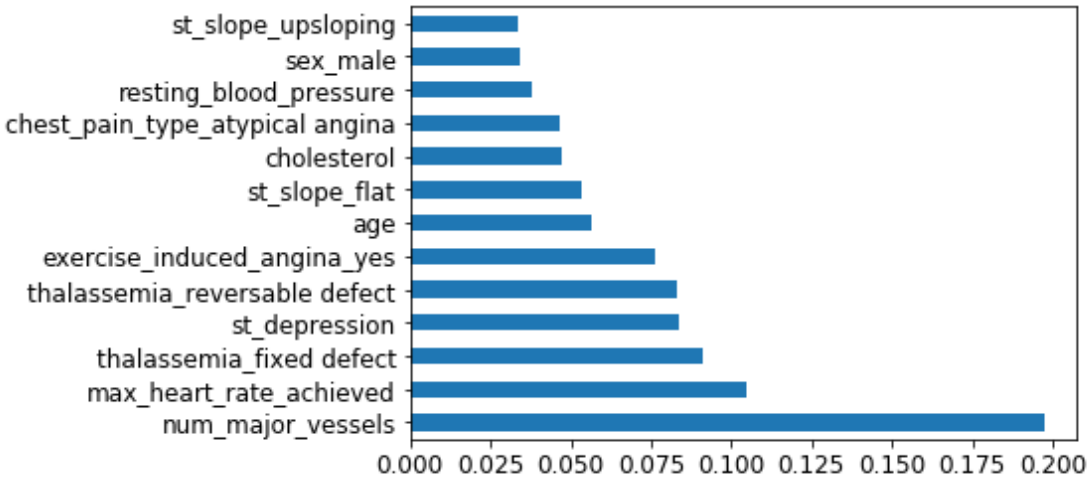
Ver features importances del RF

[38]:
```
# get importance
importance = model.feature_importances_
# summarize feature importance
for i,v in enumerate(importance):
        print('Feature: %0d, Score: %.5f' % (i,v))
```

```
Feature: 0, Score: 0.05656
Feature: 1, Score: 0.03782
Feature: 2, Score: 0.04714
Feature: 3, Score: 0.10444
Feature: 4, Score: 0.08392
Feature: 5, Score: 0.19719
Feature: 6, Score: 0.03402
Feature: 7, Score: 0.04672
Feature: 8, Score: 0.01623
Feature: 9, Score: 0.01887
Feature: 10, Score: 0.00295
Feature: 11, Score: 0.00061
Feature: 12, Score: 0.01013
Feature: 13, Score: 0.07647
Feature: 14, Score: 0.05361
Feature: 15, Score: 0.03342
Feature: 16, Score: 0.09121
Feature: 17, Score: 0.00590
Feature: 18, Score: 0.08279
```

[37]:
```
feat_importances = pd.Series(model.feature_importances_, index=X_train.columns)
feat_importances.nlargest(13).plot(kind='barh')
```

[37]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x29f985b5148>
```



Limitaciones importancia de las variables

Importancia de las features puede brindar información sobre las variables que un modelo considera bastante valiosa, pero no nos dice cómo esas características afectan a las predicciones del modelo.

Por lo tanto: Además de conocer las características importantes, también interesa saber cómo diferentes valores afectan en el resultado predictivo, así que se debe tratar con interpratabilidad de los modelos

Herramienta 1: Interpretabilidad de modelos a través de Permutation Importance

→ Objetivo: Entender cómo los diferentes factores (features) afectan en que un pasajero sobreviva o no al hundimiento del titanic

Introducción

Una de las preguntas más básicas es saber qué características tienen más impacto en las predicciones.

Cálculo de permutation importance es rápida, ampliamente utilizado y entendido, consistente con todas las propiedad que nos gustaría que tuviera una medida de importancia

Explicación

Existen features que pueden aportar o menos información.

El cálculo se hace cuando el modelo ya está entrenado

Consiste en: si mezclo aleatoriamente una sola columna del conjunto de datos, dejando todo el resto de las columnas en su lugar, cómo afectaría la predicción de esos datos mezclados

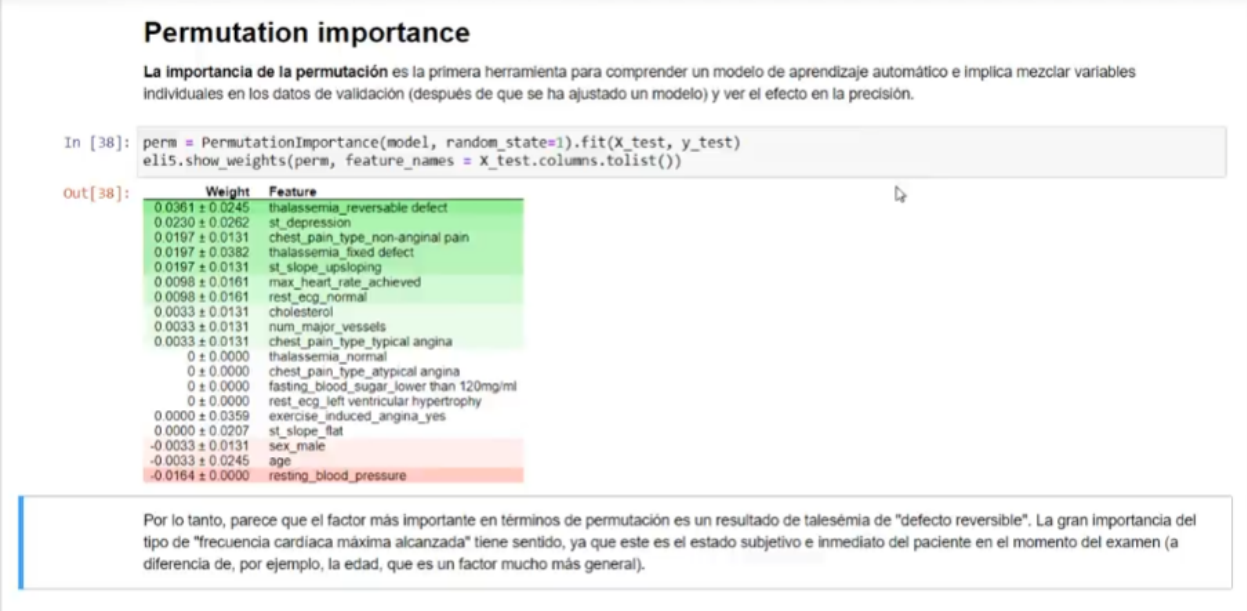
Fuente: https://www.kaggle.com/dansbecker/permutation-importance

De ordenar aleatoriamente una columna, se deberían generar predicciones menos precisas ya que los datos ya no corresponden a lo observado en el mundo real. La predicción de modelo se ve especialmente afecta si mezclamos una columna en la que el modelo se basó en gran medida para generar las predicciones, es decir, si variamos una columna importante el modelo se va a ver más afectado que si variamos una columna menos importante

Pasos para cálculo

- entrenar modelo
- Variar una columna y calcular cómo se afectó las loss la variación
- Volver al paso anterior
- Repetir los pasos anteriores hasta tener la importancia de cada columna

Ejemplo de los resultados



Permutation importance

La importancia de la permutación es la primera herramienta para comprender un modelo de aprendizaje automático e implica mezclar variables individuales en los datos de validación (después de que se ha ajustado un modelo) y ver el efecto en la precisión.

```
In [38]: perm = PermutationImportance(model, random_state=1).fit(X_test, y_test)
         eli5.show_weights(perm, feature_names = X_test.columns.tolist())
```

Out[38]:

Weight	Feature
0.0361 ± 0.0245	thalassemia_reversable defect
0.0230 ± 0.0262	st_depression
0.0197 ± 0.0131	chest_pain_type_non-anginal pain
0.0197 ± 0.0382	thalassemia_fixed defect
0.0197 ± 0.0131	st_slope_upsloping
0.0098 ± 0.0161	max_heart_rate_achieved
0.0098 ± 0.0161	rest_ecg_normal
0.0033 ± 0.0131	cholesterol
0.0033 ± 0.0131	num_major_vessels
0.0033 ± 0.0131	chest_pain_type_typical angina
0 ± 0.0000	thalassemia_normal
0 ± 0.0000	chest_pain_type_atypical angina
0 ± 0.0000	fasting_blood_sugar_lower than 120mg/ml
0 ± 0.0000	rest_ecg_left ventricular hypertrophy
0.0000 ± 0.0359	exercise_induced_angina_yes
0.0000 ± 0.0207	st_slope_flat
-0.0033 ± 0.0131	sex_male
-0.0033 ± 0.0245	age
-0.0164 ± 0.0000	resting_blood_pressure

Por lo tanto, parece que el factor más importante en términos de permutación es un resultado de talesémia de "defecto reversible". La gran importancia del tipo de "frecuencia cardíaca máxima alcanzada" tiene sentido, ya que este es el estado subjetivo e inmediato del paciente en el momento del examen (a diferencia de, por ejemplo, la edad, que es un factor mucho más general).

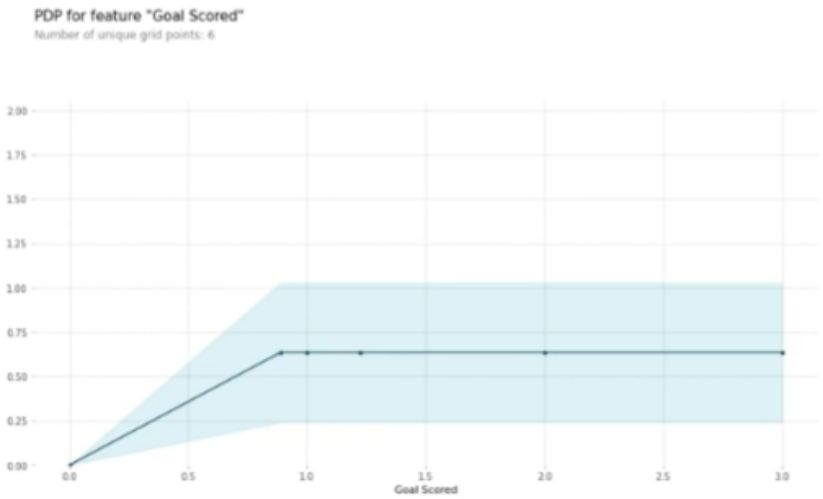
Existe una aleatoriedad en las permutaciones y estas se realizan varias veces, así se reportan los resultados con un intervalo de confianza dado los diferentes resultados de las permutaciones realizadas

Cuando los valores negativos de las importancias significa que la aleatoriedad de las permutaciones generaron resultados mejores, pero debido a la aleatoriedad, el modelo no es mejor

Herramienta 2: Interpretabilidad de modelos a través de Partial Dependence Plots (PDP)

Introducción

Partial Dependence Plots



La importancia de las características muestran qué variables afectan más a las predicciones, mientras que los gráficos de dependencia parcial (pdp) muestran cómo una característica afecta a las predicciones.

Esto es útil para responder preguntas como:

- controlando todas las demás features, qué impacto tiene la feature_x en el valor del target

Las gráficas de dependencia parcial se pueden interpretar de forma análoga a los coeficientes de un modelo de regresión lineal o regresión logística.

Sin embargo las pdp de modelos más complejos pueden capturar patrones más complejos que los modelos de coeficientes simples

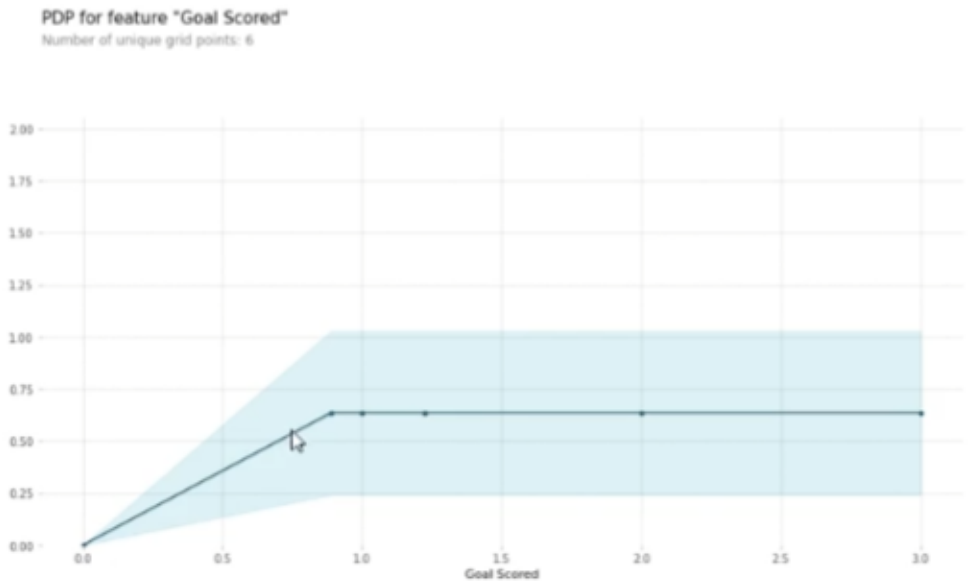
Cómo funciona

Las pdp se calculan después de entrenar un modelo

Para ver cómo los pdp separan el efecto de cada característica se comienza considerando una sola fila de datos

Se mantienen todas las características iguales pero se modifica el valor de una feature, por ejemplo, qué pasaría si tuviera posición del balón un 40%, 50%, 60%, asignando diferentes valores a una feature y ver cómo cambia el valor del target

Interpretabilidad

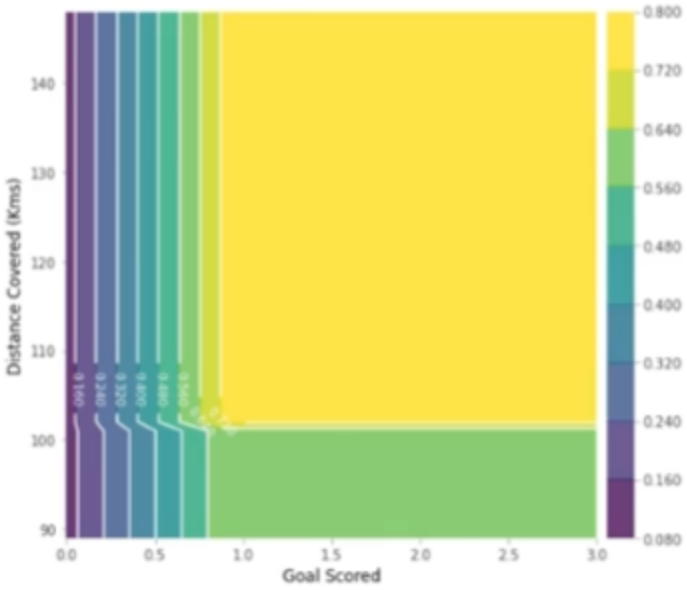


El efecto de en la probabilidad de ganar un partido (target) dado que cambian los valores de los goles convertidos (feature). Aquí en el gráfico se observa que desde que se hace 1 gol o más, la probabilidad de ganar el partido no se incrementa más

Línea azul: intervalo de confianza

Partial depedence plots (pdp) en 2 dimensiones

Se muestran predicciones para cualquier nivel de combinación entre las 2 features (conocer las interacciones entre 2 features)



Ejemplo imagen (Features: distancia recorrida y número de goles): Se observa el área amarilla donde la probabilidad de ganar es máxima cuando se convierte sobre un gol y la distancia recorrida es sobre 100 km

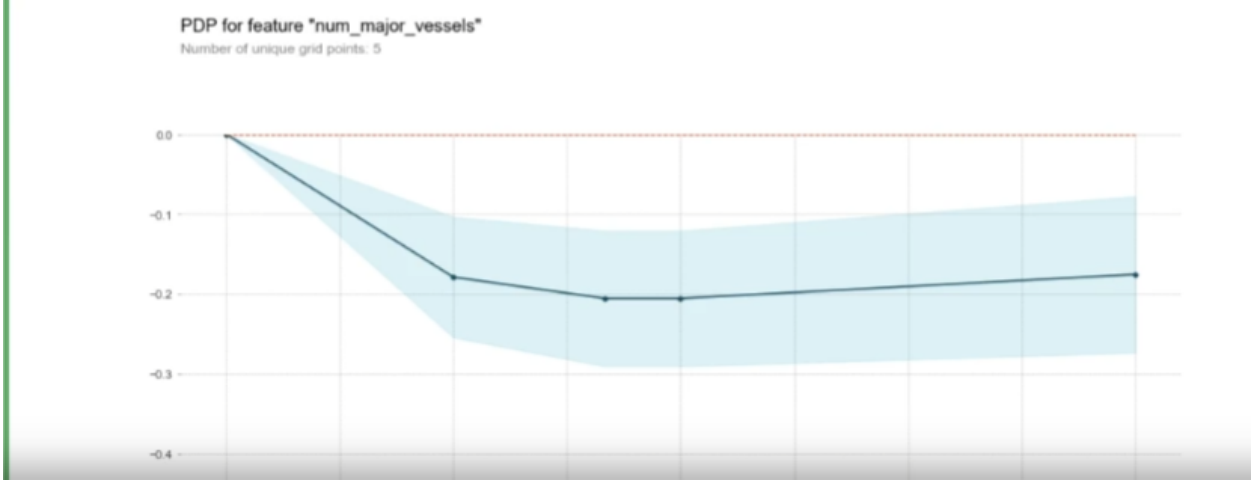
Ejemplo 1 - dependencia clara

Se observa que al incrementar la feature la probabilidad del Target disminuye hasta llegar a un cierto punto donde deja de disminuir la prob del Target e incluso comienza a disminuir el valor

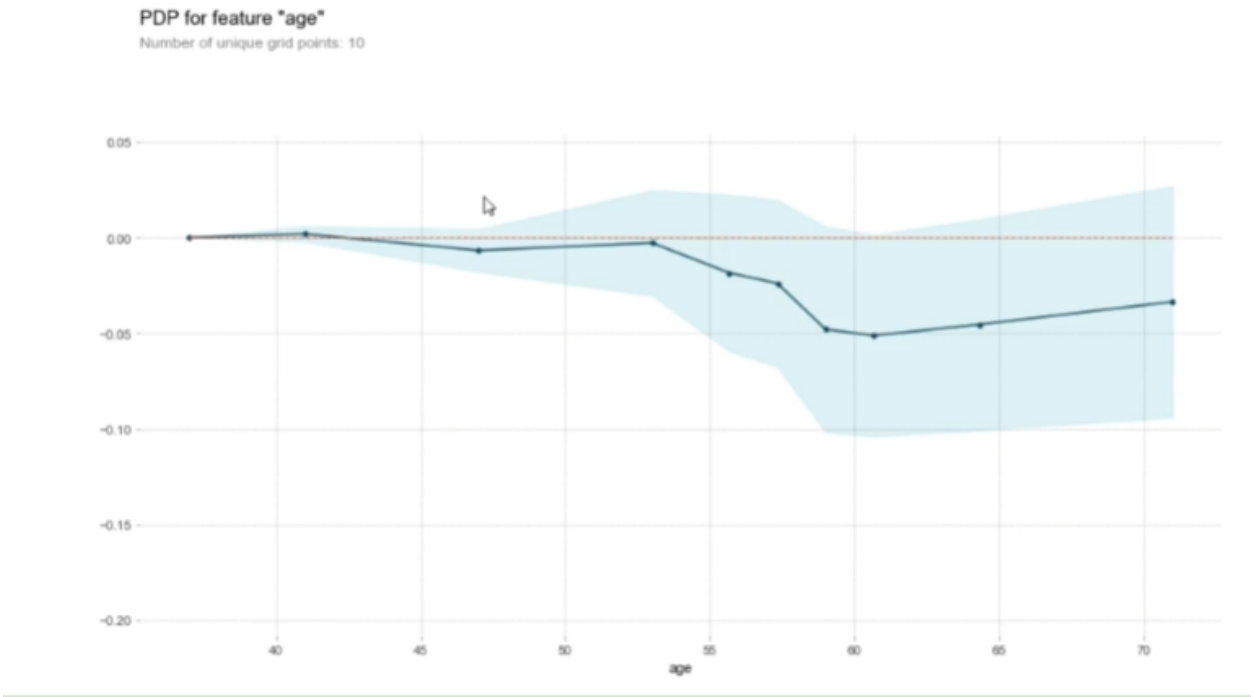
```
In [22]: base_features = dt.columns.values.tolist()
         base_features.remove('target')

         feat_name = 'num_major_vessels'
         pdp_dist = pdp.pdp_isolate(model=model, dataset=X_test, model_features=base_features, feature=feat_name)

         pdp.pdp_plot(pdp_dist, feat_name)
         plt.show()
```



Ejemplo 2 - dependencia más compleja

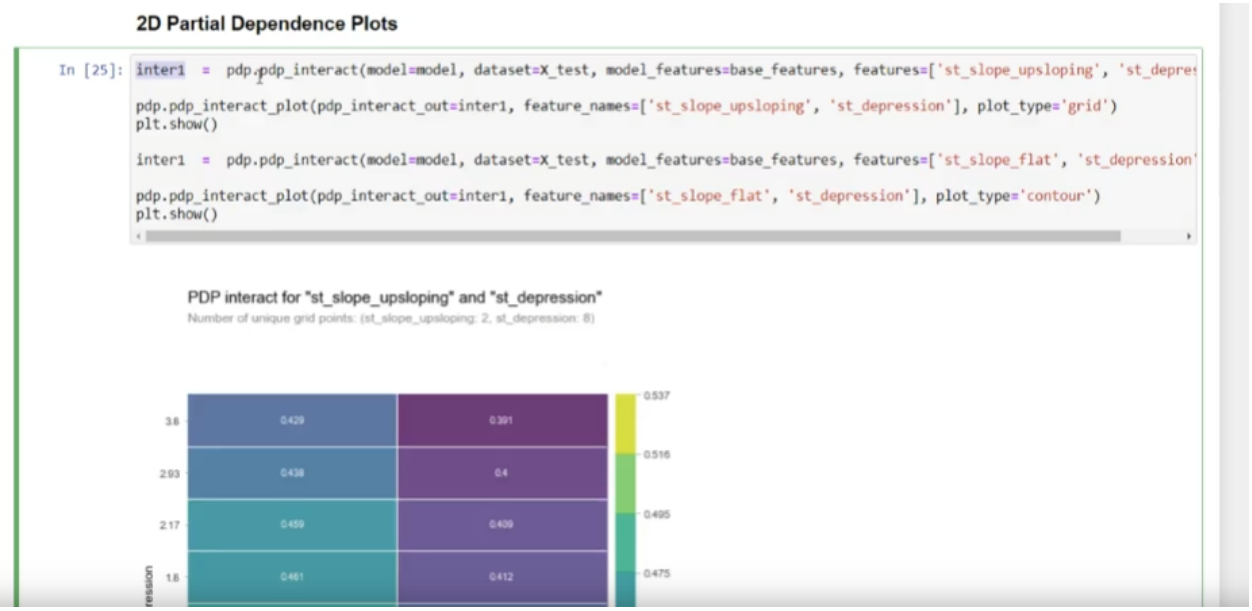


Aquí se observa que a medida que incrementa la edad (feature) el riesgo de enfermedad cardiaca disminuye (target). ESTO ES EXTRAÑO, ya que la lógica debería ser que a medida que incrementa la edad el riesgo aumenta.

Pero al ver el intervalo de confianza podría tratar de rechazar

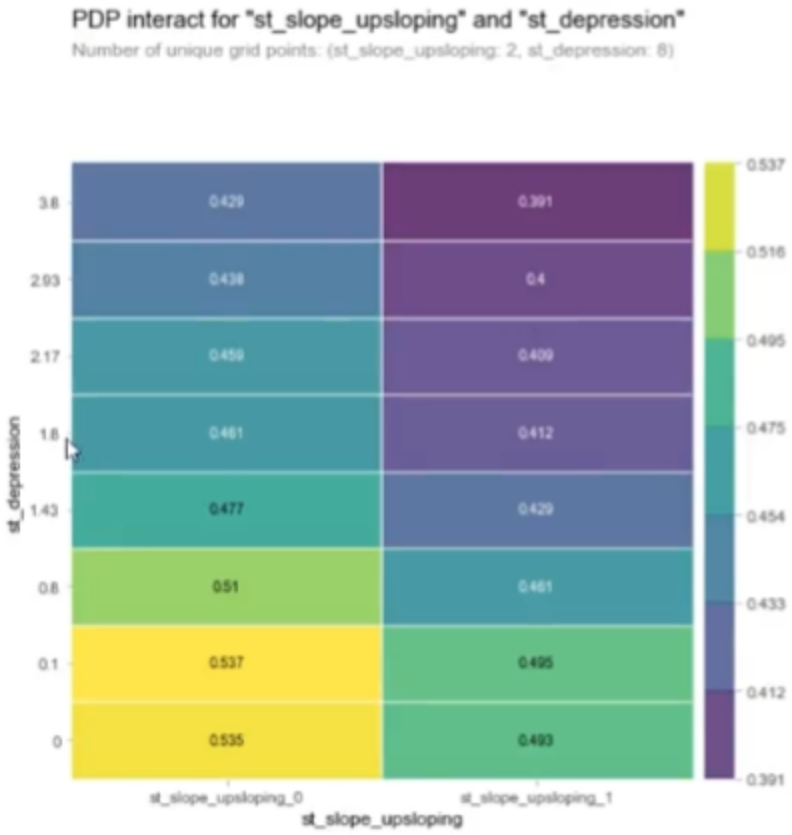
Ejemplo 3 - pdp 2 features

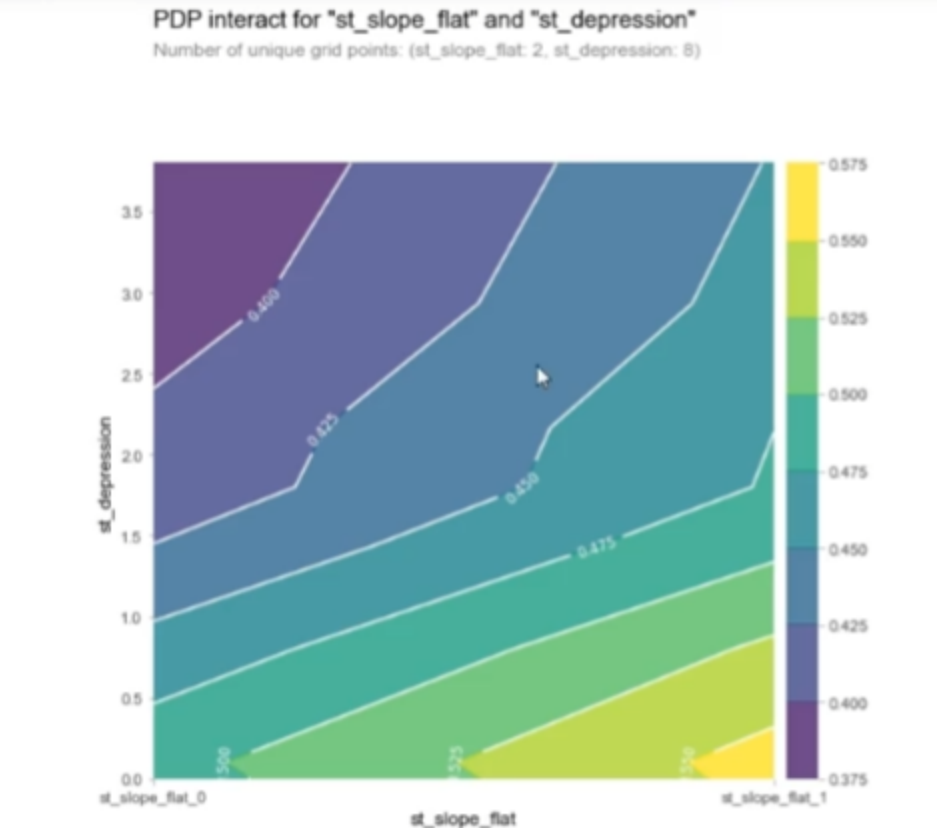
Ejemplo 3 - pdp - 2 features



Se pueden mostrar los resultados como gráficos de barra o de contorno

Gráfico de barras


PDP interact for "st_slope_upsloping" and "st_depression"
Number of unique grid points: (st_slope_upsloping: 2, st_depression: 8)
st_depression
3.8
2.93
2.17
1.8
1.43
0.8
0.1
0
0.429
0.391
0.438
0.4
0.459
0.409
0.461
0.412
0.477
0.429
0.51
0.461
0.537
0.495
0.535
0.493
st_slope_upsloping_0
st_slope_upsloping_1
st_slope_upsloping
0.537
0.516
0.495
0.475
0.454
0.433
0.412
0.391



Por ejemplo, cuando las features_x y features_y toman los valores más bajos, la probabilidad del Target toma los valores más bajos

Herramienta 3: Interpretabilidad de modelos a través de SHAP

Introducción

Permite analizar cómo se comporta el modelo para una predicción individual. Por ejemplo, que un médico quiera saber qué factores están impulsando más el riesgo de que cada paciente pueda padecer una enfermedad

Shap: ***SHapley Additive exPlanations***

Interpretan el impacto de tener un cierto valor de una feature determinada vs que esa feature tome un valor de referencia

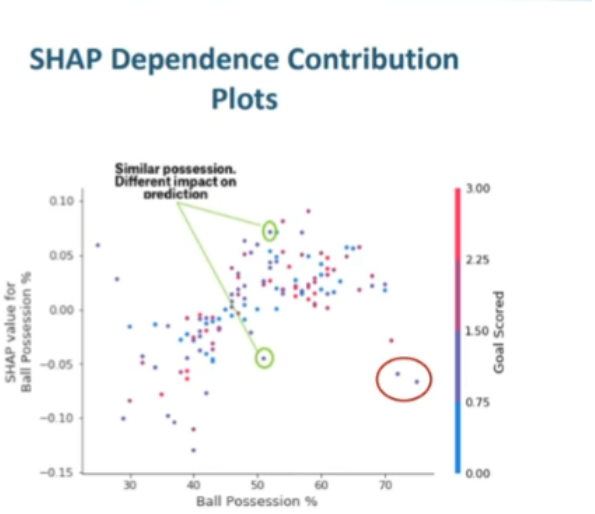
Se suman los shapley values para explicar porqué una prediucción en concreta fue diferente de la línea base

*linea base: es una predicción base

Tipos de gráficos y su explicación

En rojo todos las features que generar un incremento en las predicciones. Entre más grande mayor es el efecto

En azul todas las features que generan una disminución en las predicciones



Summary plot

la importancia de la permutación permitió saber qué caterísticas son importantes pero puede pasar que tenga un efecto medio y que tenga un efecto para algunas predicciones pero en general ningun efecto.

En cambio, los gráficos resúmenes de shap muestran el detalle individual:

- eje vertical: ordenadas las features de mayor importancia a menor importancia (según el video)
- Eje hortizontal: shap value. La contribución en el valor de la predicción
- Pintadas: el valor que toma la feature

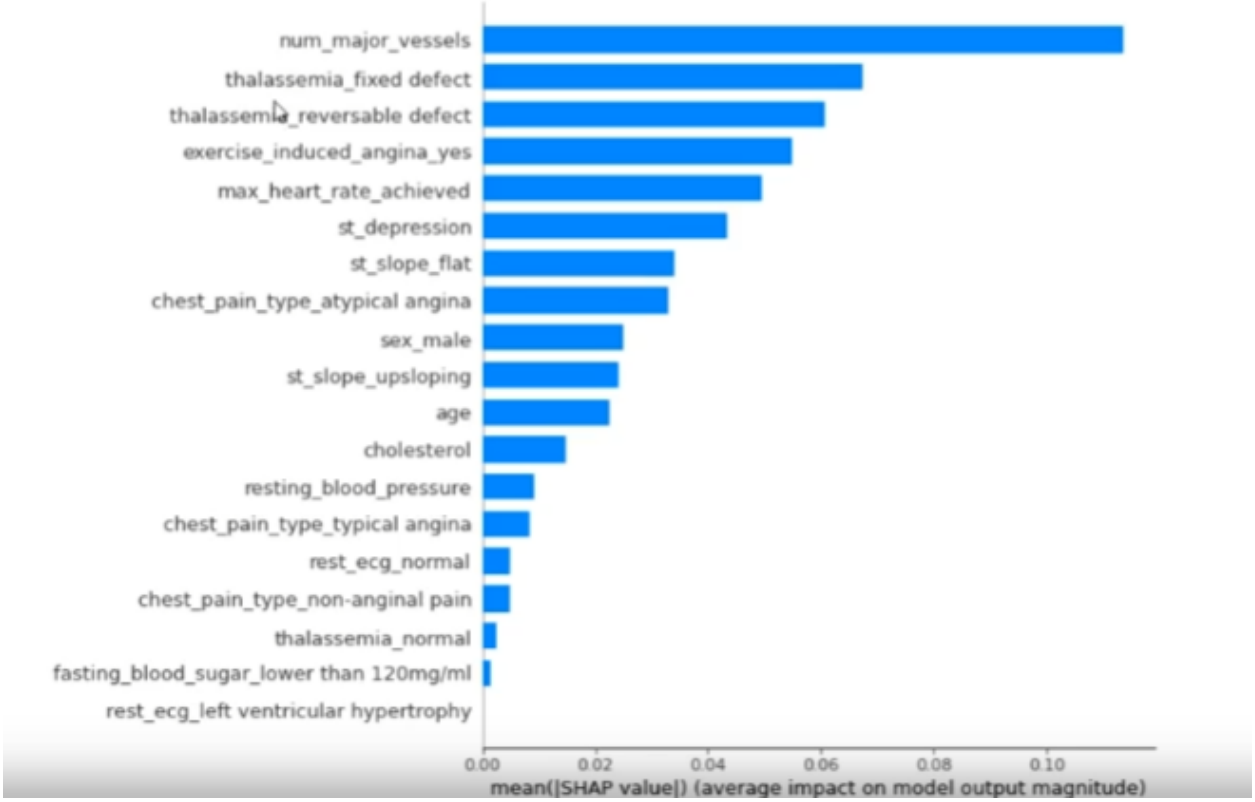
Por ejemplo se puede observar en la primera fila “gola scored” del summary plot que valores bajos de la feature (pintada de color azul) generan un impacto negativo en la predicción del Target (shap value bajo)

Mean shap value de cada feature

También se puede calcular el promedio de shapley value para cada una de las filas y obtener un gráfico de importancia similar al de feature permutation

```
]: explainer = shap.TreeExplainer(model)
   shap_values = explainer.shap_values(X_test)

   shap.summary_plot(shap_values[1], X_test, plot_type="bar")
```



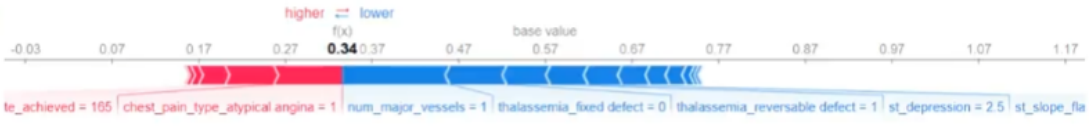
Análisis individual - shapley values

```
In [71]: def heart_disease_risk_factors(model, patient):

             explainer = shap.TreeExplainer(model)
             shap_values = explainer.shap_values(patient)
             shap.initjs()
             return shap.force_plot(explainer.expected_value[1], shap_values[1], patient)
```

```
In [72]: data_for_prediction = X_test.iloc[1,:].astype(float)
         heart_disease_risk_factors(model, data_for_prediction)
```

Out[72]:

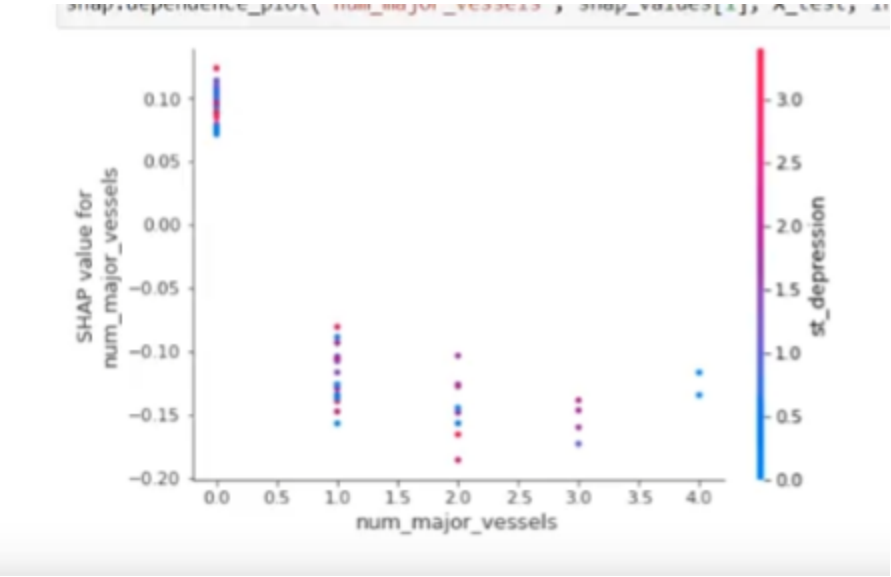


Para esta persona, su predicción es del 36% (en comparación con una línea de base del 58,4%). Muchas cosas están funcionando a su favor, incluido tener un vaso principal, un defecto de talasemia reversible y " no " tener una pendiente plana.

El valor base es 0.57 pero para este paciente su predicción es 0.34. Esto porque el valor num_major_veselss es igual a 1 que de acuerdo a los datos es un valor bajo que hace bajar el target

Shap dependence contribution plots

Además se puede graficar: “Gráfico de contribución de dependencia de SHAP” (dependence plot) que son bastantes auto explicativos en el contexto de los valores de shap



con la feature en el eje x el efecto en subir el valor del Target es bastante alto (shapley value en el eje y bastante alto). Por otro lado, la segunda feature de comparacíón que aparece pintada no se observa un efecto claro

Gráfico de efecto individual para múltiples observations

Finalmente, también se puede ver un gráfico con el efecto para múltiples observaciones

Rojo y azul el efecto de subir y bajar las predicciones respectivamente