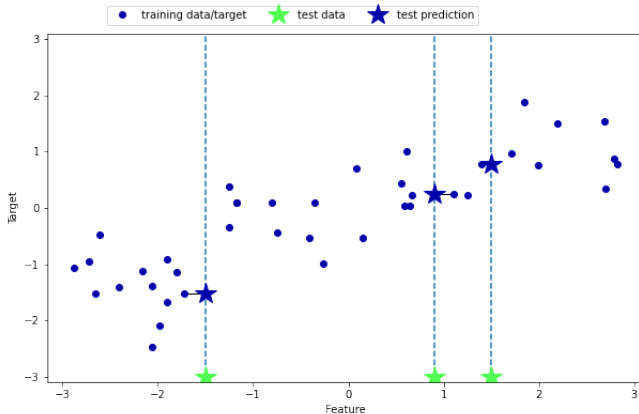


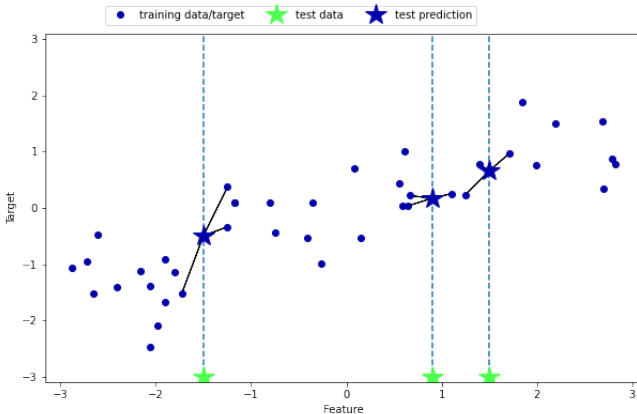
KNeighborsRegression crea una función promediando las imágenes de los k datos más cercanos

#2

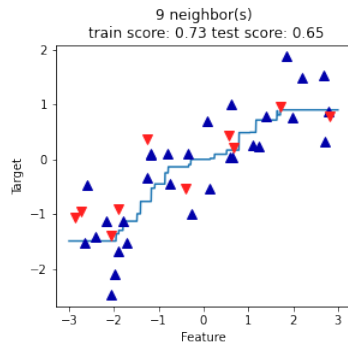
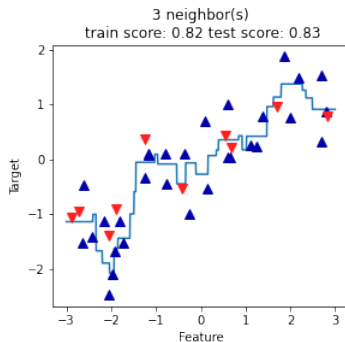
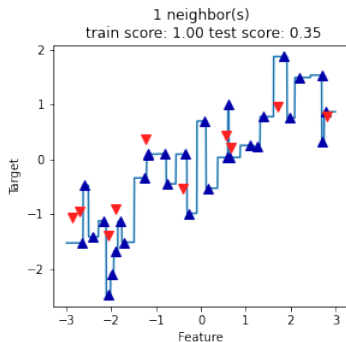


KNeighborsRegression crea una función promediando las imágenes de los k datos más cercanos

#3



En **KNeighborsRegression** la influencia del conjunto de entrenamiento disminuye mientras crece el número de vecinos

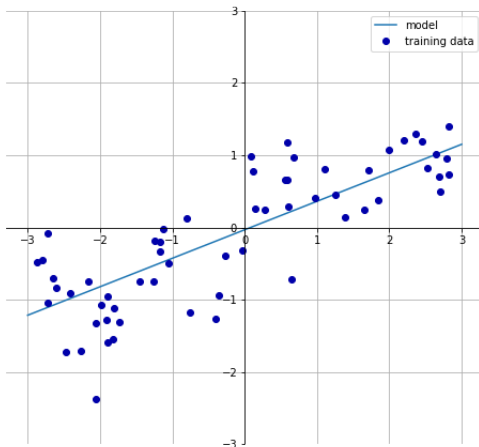


Este modelo es un buen modelo base antes de considerar técnicas más avanzadas

Sus fortalezas y debilidades son:

- Es sencillo de implementar y de entender.
- El principal parámetro es el número de vecinos. Usualmente funciona bien con 3 o 5 vecinos.
- Su desempeño empeora cuando hay muchas características.
- Es particularmente malo cuando muchas de las características son casi siempre 0.

LinearRegression busca una recta (plano, hiperplano) que minimice errores cuadrados en el conjunto de entrenamiento



Los modelos **Ridge** y **Lasso** son parecidos a **LinearRegression**, pero con *regularización*

- *Grosso modo* más regularización quiere decir que el modelo se ajusta menos al conjunto de entrenamiento.
- En **Ridge** se busca que los coeficientes sean cercanos a cero.
- En **Lasso** varios de los coeficientes serán cero. Algunas características no se tomarán en cuenta.

Para probar estos modelos de `sklearn.linear_model`, en `sklearn.data_sets` se encuentra el data set Boston Housing. Se busca predecir el valor medio de las casas de varios barrios de Boston con datos como criminalidad, cercanía al río..., etc.

A diferencia de Linear Regression, Ridge y Lasso tienen parámetros ajustables

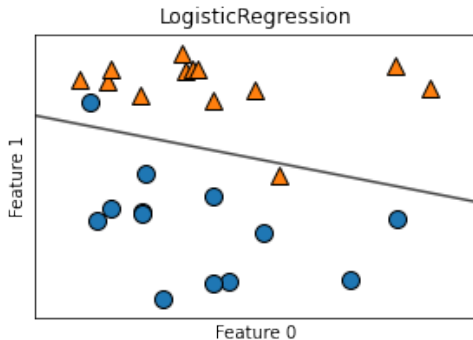
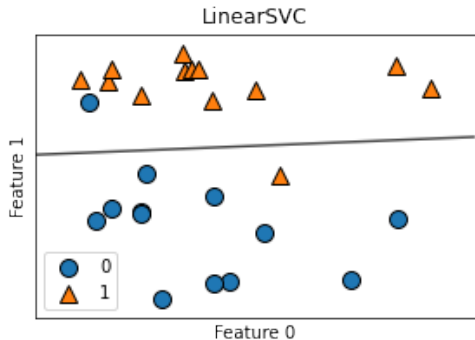
Unos puntos de consideración son:

- **Ridge** es usualmente la primera opción entre los dos métodos. Pero es más difícil de entender.
- **Lasso** es más fácil de entender, pues escoge unas características y descarta otras.
- En **scikit_learn** existe la opción de combinarlos con **ElasticNet**.

Otros métodos lineales dividen el espacio muestral en dos regiones

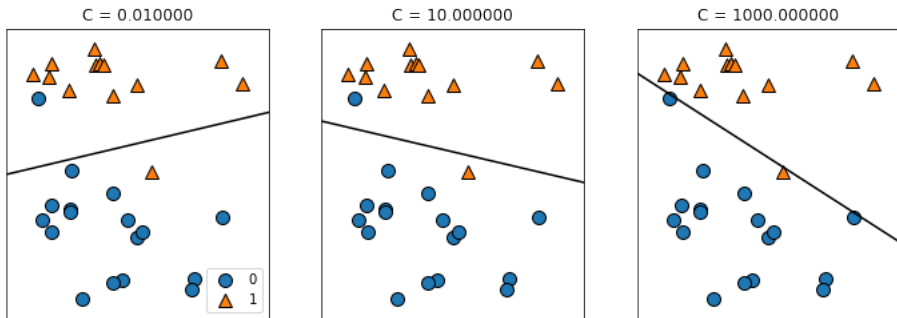
En linear_model tenemos **LogisticRegression**.

En svm (*support vector machines*) tenemos **LinearSVC**.



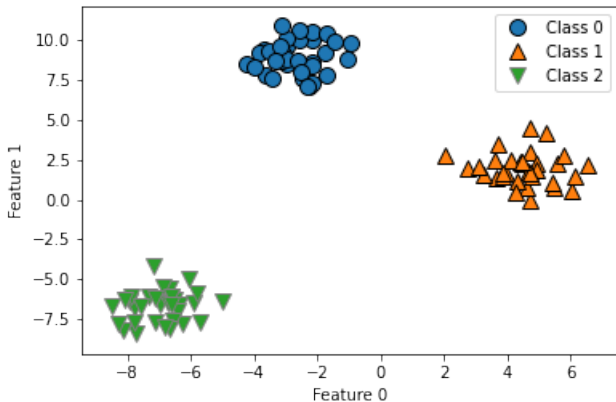
Ambos métodos dependen de un parámetro C

A mayor parámetro C , más se ajusta el modelo al conjunto de entrenamiento.



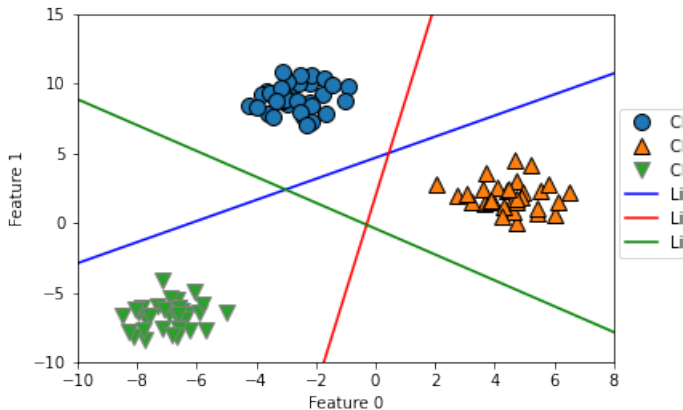
Estos métodos se pueden generalizar con el método *uno vs. el resto*

- Se crea un modelo para separar cada clase del resto.
- Para un dato de prueba, se asigna la mejor de las predicciones individuales.



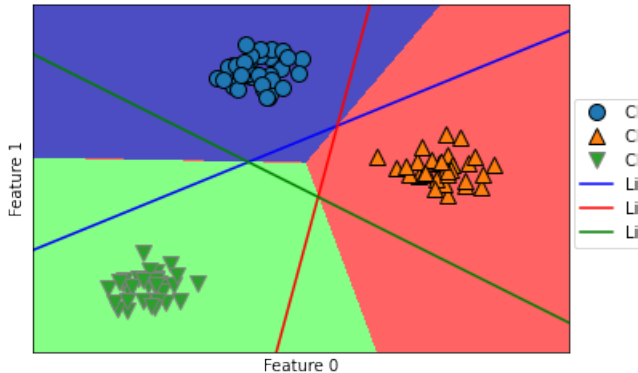
Estos métodos se pueden generalizar con el método *uno vs. el resto*

- Se crea un modelo para separar cada clase del resto.
- Para un dato de prueba, se asigna la mejor de las predicciones individuales.



Estos métodos se pueden generalizar con el método *uno vs. el resto*

- Se crea un modelo para separar cada clase del resto.
- Para un dato de prueba, se asigna la mejor de las predicciones individuales.



Los modelos lineales son rápidos en entrenarse y en predecir

- Es fácil entender cómo se hacen las predicciones en estos modelos.
- Se desempeñan muy bien con datasets muy grandes.
- A menudo no es fácil entender por qué los coeficientes son los resultantes, sobre todo cuando hay gran correlación entre las características.