**Отчет по дисциплине:**  Теория машинного обучения

**Язык программирования:** Python

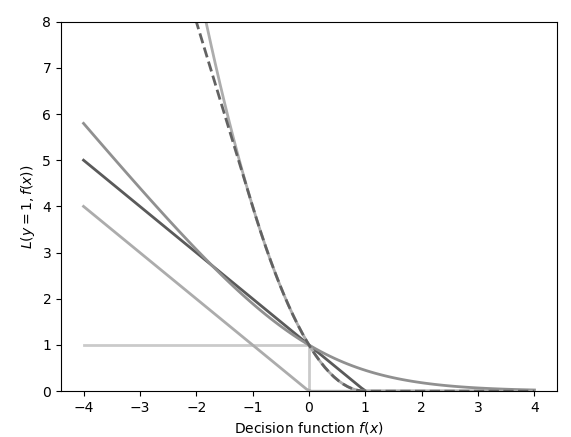
**Тема №3:** Линейному методу классификации

**Ход работы программы:**

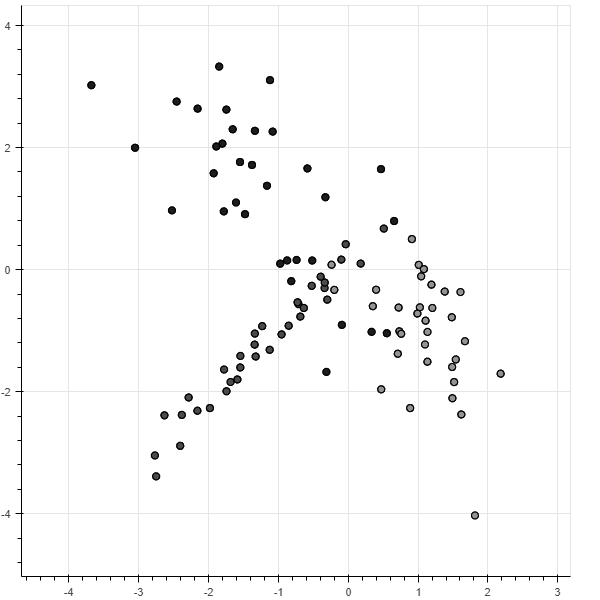
Используя готовые методы из библиотеки sklearn, применить линейные алгоритмы классификации исходных данных. Поиск гиперплоскости оптимально разделяющей элементы класса  и элементы относящиеся ко всем остальным классам. Будет использоваться метод градиентного спуска.

В пакете **sklearn** данный классификатор описан в классе **SGDClassifier**

Выбрана функция потери с такими графиками:



А так же имеется выборка:



Определяем выборку в коде:

**clf = SGDClassifier(loss="hinge", penalty="l2", max\_iter=300)**

**clf.fit(X,Y)**

classes\_field = []

for y in range(x\_steps):

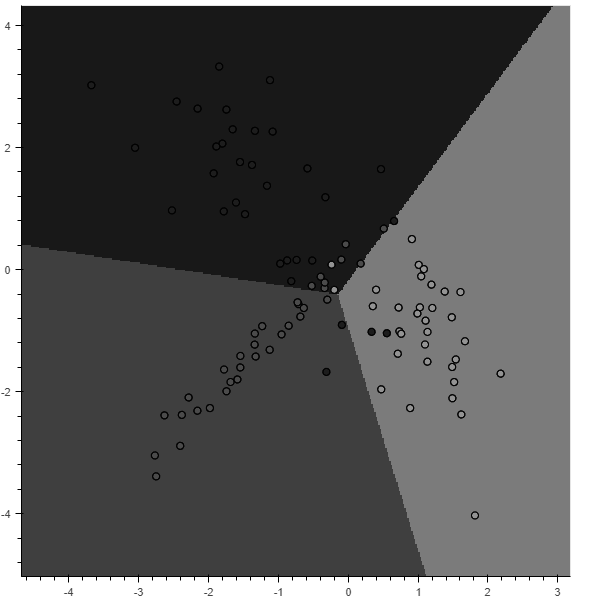
classes\_line = []

for x in range(y\_steps):

classes\_line.append(clf.predict([[x\_min+(x\*x\_step\_len),y\_min+(y\*y\_step\_len)]]))

classes\_field.append(classes\_line)

Посмотрим на работу функции потери Hinge (петли):



**clf = SGDClassifier(loss="epsilon\_insensitive", penalty="l2", max\_iter=300)**

**clf.fit(X,Y)**

classes\_field = []

for y in range(x\_steps):

classes\_line = []

for x in range(y\_steps):

classes\_line.append(clf.predict([[x\_min+(x\*x\_step\_len),y\_min+(y\*y\_step\_len)]]))

classes\_field.append(classes\_line)

Посмотрим на работу алгоритма функций потери Epsilon:

